OVERVIEW OF THE ROM
v3.0
by
Andy Wright

NOTES ON THE ROM SOURCE LISTING

This disk contains the complete ROM 3.0 source code. Source code is the text that an assembler program reads and converts into machine code, ready to go into a ROM hip. The files on the disk are in a slightly different form than normal source files. They were made by re-directing the output that the assembler would have sent to a printer during assembly to a disk file instead. This gives addresses in the left-hand column, followed by the values at those addresses (both in hexadecimal) followed by the actual source file text. So, for example, near the start of the lower ROM you will see:

```
0016 C5       BCJUMP:     PUSH BC      ;16
```

The first number gives the address in the ROM, and the second number is the value there, so that PRINT HEX$(PEEK &0016) gives C5. BCJUMP is a Label that is used to refer to this bit of the ROM in other parts of the source listing. PUSH BC is the Assembly Language text that caused the Assembler to generate the C5 machine code equivalent. A semi-colon precedes any REMs by the author - here I just put a 16 to remind myself as I wrote the source (without the benefit of the left-hand two columns) where things would be in memory.

This doesn't usually matter very much, but the first part of a ROM used by a Z80 processor is a special case - see the section on RESTARTS below.

You will find the Technical Manual (available via FORMAT) useful for a full understanding of interactions with the hardware. The text of the ROM source has been left just as I wrote it. Some parts are not terribly well documented, since I was writing to explain things to myself rather than an audience. The symbols used may not correspond exactly to the Technical Manual's usage - for example, I started using URPORT and LRPORT (for Upper RAM Port and Lower RAM Port) before Bruce Gordon told me he was using HMPR and LMPR for the same things! Also, to the hardware, the screen MODEs are 0-3, not 1-4, and I think I may have used MODE inconsistently. Some sections have been "commented out" by inserting semicolons; for example, a table of EQUs that allowed me to write e.g. CP BORDERTOK instead of CP 9EH had to be cut when the assembler ran out of symbol space! At one point I changed assemblers, and the new one didn't allow e.g. LD A,'K'. I did a search and change of all single quotes to double quotes; this accounts for the occasional appearance in the text of CAN"T and A" rather than CAN'T and A'.

You may think the code is oddly arranged, or contains odd things like DB 0DDH used to jump three bytes. This is because space was very tight, and any trick that saved a byte was needed. This included moving routines into "illogical" positions if this allowed a Label to be reached by a 2-byte JR instead of a 3-byte JP instruction.

A NOTE ON HEXADECIMAL NUMBERS

I know some people don't like hex, but it does make life much easier if you can get used to it. For example, in the first part of the lower ROM is the line:

```
000A C3CE37        JP ERROR2
```

In any 3-byte Z80 instruction, the first byte is the code for the operation, or "op code". In this case it is C3, for JP. The next 2 bytes are a value, or address, in reverse order, so they should be read 37 and then CE. In other words, JP to 37CE hex, which you can easily find in the later part of the lower ROM. (The Label ERROR2 is at 37CE.) A decimal version would not be as easy to use:

```
0010 195 206 55JP ERROR2
```

Besides, it takes more room and it isn't as neat.

PAGING

There is only one actual ROM chip in the SAM, but it is handled in two sections. The lower ROM, also called ROM0 (machine code programmers often start counting at zero!) is usually paged in at 0000 to 3FFF (0 to 16383) whereas the upper ROM, or ROM1, is paged in at C000 to FFFF (49152 to 65535) when required.

The lower ROM is ALWAYS paged in unless user-loaded software initiates complex paging operations. The lower ROM pages in the upper ROM as required, for example, to handle floating-point calculations, BEEP or tape operations.

RAM page 0 is usually paged in at 4000-7FFF (16384-32767) and holds important system variables and the start of the current Basic program. The upper half of memory, 8000-FFFF, is used to hold the section of RAM that is currently being worked on. The early part of a Basic program would involve RAM page 0 being paged in at 8000 (which automatically means that page 1 follows at C000). It doesn't matter that RAM page 0 is also simultaneously paged in at 4000. As later parts of the Basic program are interpreted, the start of the current line will occur higher and higher in memory, passing from, say, BF85 to C023. While any structure in memory, like a Basic line, starts below C000, it can be 16K long before it will "fall off" the end of the memory map. This simplifies the job of the ROM, since it doesn't need to keep checking, for every byte it reads, whether the end of the memory map has been reached. It is sufficient if it checks, say, every line, if the start of the line is above C000; if it is, the paging can be altered so that the same byte of RAM is re-mapped 16K lower (which will be somewhere in the region 8000 to BFFF) and again we have ample room before we fall off the end of the memory map. This system can be extended with up to 32 pages of RAM (the maximum that the ports that page internal RAM can control) allowing for very big programs indeed.

The screen is paged in, starting at 8000, when it must be read from or written to.

The fly in the ointment, in a way, is the upper ROM. When paged in, it covers RAM at C000-FFFF and could obscure the later part of a Basic line or variable, or the later part of the screen. This makes it unsuitable for many kinds of routines. Floating-point calculation routines are fine, because they work on numbers in the "system page" - RAM page 0, paged in at 4000-7FFF. However, graphic routines should not be in the upper ROM because the entire screen could not be paged in at the same time. (Well, it could be paged in at 4000-BFFF, but that would mean paging out the system page, which holds important variables like the PEN colour, screen MODE, the machine stack, information for interrupts, etc.)

All this means that too much code cannot be run in the upper ROM. To get round this, I had to do nasty things like copying routines from this ROM to a RAM buffer in the system page for execution. The first part of the upper ROM consists of such routines, and the addresses given in the left-hand column correspond to the addresses in the RAM buffer where the code is RUN, not to the actual addresses in the upper ROM where the code is

STORED. The routines involved were chosen because their speed is not critical - the copying process obviously imposes a slight delay. Other routines are quite convoluted because of the problem with the upper ROM. The PRINT routines, for example, start in the lower ROM, use the upper ROM where this is feasible for the interpretation of characters, and then use the lower ROM again for actually putting data on the screen.

SAMDOS, MasterDOS and MasterBasic all page into 4000-7FFF for execution. They page out the normal machine stack and replace it with one of their own. Special routines are used to read or write the (paged out) system page variables, or to Call ROM routines with the system page temporarily paged in.

HOW THE BASIC INTERPRETER WORKS

I do not have the time to explain this in detail, but the following rough outline may help.

When the SAM is turned on, the start of the ROM, at 0000, is executed. This jumps to an initialisation routine in the upper ROM, which clears RAM, sets up the system variables, screen mode, rainbow start-up screen, copyright message etc. and then waits for you to press any key. It then indirectly enters the interpreter's "main loop", as it would after an error report.

The main loop is at 0E84 in the lower ROM. The first action is a CALL AUTOLIST which would list the current part of the Basic program, if there was one. After clearing various scratch-pad areas (such as the edit line) with CALL SETMIN, the lower screen is selected for input and output with CALL STRM0. Then theEDITOR is called to get input from the user. When the user presses RETURN, the Editor hands back control, and the main loop tokenises any keywords in the input using CALL TOKMAIN. (E.g. print is turned into a single-byte internal form that is shorter and easier to handle.) The modified line is then checked for correct syntax using CALL LINESCAN. This process also inserts invisible forms of numbers in the line. The invisible form is used to speed things up during program execution.

If there is an error, the invisible forms are stripped out again at 0EB0, a warning BEEP is produced, and the loop is gone round again, giving the user a chance to modify the line.

If there were no errors, MAINE2 at 0EBB is entered. If the line has a line number, a jump back to MAINEADD at 0E7A is made to add the line to the program and then re-enter the main loop. If there is no line number, a check is made to see if the line is just RETURN. (At 0EC0, RST 18H is equivalent to CALL 0018 - this gets the current character in the A register. 0DH is the code for RETURN or ENTER.) If the line is not just a RETURN, it must start with a direct command, such as RUN, PRINT, CLS, etc.

The section from 0EC5-0ED8 is to do with clearing the lower screen, deciding whether to clear the upper screen or not, and setting a counter for the the "scroll?" prompt. The instruction at 0EDE is very important; bit 7 of the FLAGS system variable is SET, signalling "running, not syntax checking". Many of the interpreter's routines to handle commands can be run in two states; either with this bit RESET, to check syntax during LINESCAN, or with it SET when the line is to be executed.

CALL COMPILE creates LABEL values and inserts information about the location of procedures and functions into the program. CALL LINERUN actually executes the line, including executing all or part of the program, if the line contains e.g. RUN or GOTO. When a report is to be printed, such as "OK" or "BREAK" or "Out of memory", the operations started by LINERUN are terminated and the program returns to MAINER at 0EED. Various "housekeeping" tasks are performed, then, provided ON ERROR is not active, the program goes to MAINER3 at 0F67, prints the "report" via a CALL ERRHAND1, then jumps back into the main loop.

Now to expand a little on the LINERUN/LINESCAN routines. They have a common heart, the "statement loop" at 0D79 that looks along a Basic line, identifying the first significant character in each statement. This is usually a command token, which is used to find an associated address from a table pointed to by the system variable CMDADDRT at 0DBD. The table, which is in the upper ROM at FD65, gives the start addresses of each command. A jump is then made to the appropriate routine at 0DD3. After the statement has been handled, a return is made to NEXTSTAT at 0DD4 and the statement loop is repeated if there are more statements. If not, the routine terminates if it is "syntax time", or goes to the next line in "run time".

As an example of the handling of a command, I will use RESTORE. The address table gives the address of the RESTORE routine as 348F. Here we find a CALL SYNTAX3, which is a routine that evaluates the (optional) expression expected after RESTORE. During syntax checking, the invisible forms of any numbers are inserted after the visible forms, but the SYNTAX3 routine never returns to where it was called from - instead it POPs a return address and then RETs to NEXTSTAT.

When a program is running, on the other hand, SYNTAX3 leaves the result of the expression (or zero if there isn't an expression) on the floating-point calculator stack, and does a normal RETurn to the RESTORE routine at CALL GETINT. CALL GETINT removes the result from the FPC stack, compressing it into a double register. The value in the register is used to find the address of the RESTOREd line number so that the DATA pointer can be set up correctly at RESTORE3.

RESTORE is a simple command but illustrates general rules: there is a first part which is common to syntax checking and running, after which an early return is made to NEXTSTAT during syntax checking, or a further, active part of the routine is executed if the program is running. A final RET goes back to NEXTSTAT. Numeric and string expressions that may follow a command are stacked on the calculator stack and removed as required during the "active" routine. (The last-evaluated expression will be the first item removed - the radius of a CIRCLE, for example.)

THE RESTARTS

The Z80 can use a short, fast version of CALL for 8 particular addresses. These RESTART (shortened to RST) instructions take just 1 byte instead of the usual 3 for a CALL, and 11 instead of17 clock tics. The programmer tries to put commonly-used routines at these special addresses. On the SAM (which in this respect is similar to the Spectrum) the special addresses and the associated routines are:

0000 (RST 0) - Well, the computer starts executing here when it is first turned on, so "start-up" code has to go here, and it is not much use as a RST.

0008 (RST 08H) - Handles errors. At other places in the source code, you will see things like:

```
0500 CF     RST 08H
0501 16     DB 22        ;"END OF FILE"
```

The RST 08H CALLS 0008, but the return address is never used to RET to 0501 - instead, it is used by the error-handler to read the byte at 0501 (i.e. 16H) and generate the associated error message from a list. My REM after the semi-colon reminds me of the message involved.

0010 (RST 10H) - PRINT the character in the A register.

0018 (RST 18H) - Get the current character (pointed to by CHAD) from a Basic line.

0020 (RST 20h) - Get the next character by advancing CHAD.

0028 (RST 28H) - Call the floating point calculator. A series of code bytes for the desired operations will follow the RST 28H instruction. In the source, I have usually used DB CALC as a synonym for RST 28H. The calculator "language" provides a convenient and concise way of calculating e.g. SIN, COS, SQR etc. The calculator is in the first part of the upper ROM.

0030 (RST 30H) - CALL or JP to the upper ROM. The address to go to follows the RST 30H instruction. 8000 is subtracted from it to give a JP - otherwise, you get a CALL. Some examples occur in the Jump Table starting at 0100.

0038 (RST 38H) - Maskable Interrupts come here, so this is not much use as a RST. (Non-maskable interrupts goto 0066.)

EXAMINING YOUR ROMS

If you would like to compare the contents of your ROM chip wit hthe values given in the source listing, you can use the EXAM program which is included on this disk to do it.

```
0000                ;SAM COUPE ROM 3.0 SOURCE CODE Copyright Andrew J.A. Wright 1989-90
0000                ;MAIN.SAM
0000                        ORG 0,0
0000
0000 0000=          PAGE0   EQU 0
0000 0001=          PAGE1   EQU 1
0000 001F=          PAGE1F  EQU 1FH
0000
0000                        ORG 0000H,0
0000
0000 F3             L0000:  DI
0001 C3B000                 JP MINITH
0004
0004 E1                     POP HL              ;04 FOWIA
0005 E9             HLJUMP: JP (HL)
0006
0006 FDE9           IYJUMP: JP (IY)
0008
0008                ;RST 08H - ERRORS
0008
0008 00                     NOP                 ;SOME HARDWARE MIGHT LIKE THIS...
0009 D9                     EXX
000A C3CE37                 JP ERROR2
000D
000D 77             NRWRITE: LD (HL),A
000E C9                     RET
000F
000F 1E                     DB 30               ;0FH. ROM VERSION NUMBER
0010
0010                ;RST 10H - PRINT A
0010
0010 C39E01                 JP RST102
0013
0013                ;0013H - PRINT BC FROM (DE)
0013
0013 C32702         PRINTSTR: JP SOP2           ;13
0016
0016 C5             BCJUMP: PUSH BC             ;16
0017 C9                     RET
0018
0018                ;RST 18 - GET CHAR. SKIP ALL CONTROL CODES EXCEPT CR.
0018
0018 2A975A         GETCHAR: LD HL,(CHAD)       ;18
001B C5                     PUSH BC
001C C3C100                 JP GETCHAR1
001F
001F 00                     DB 0
0020
0020                ;RST 20H - NEXT CHAR        ;20
0020
0020 2A975A         NEXTCHAR: LD HL,(CHAD)
0023 23                     INC HL
0024 C5                     PUSH BC
0025 C3BE00                 JP NEXTCHAR1
0028
0028 C3E600                 JP FPCP2            ;28 FLOATING POINT CALCULATOR
002B
002B 10FE           DELAYB: DJNZ DELAYB         ;2B
002D
002D DDE9           IXJUMP: JP (IX)             ;2D
002F
002F 00                     DS 1,0
0030
0030 C3CF01                 JP RST30L2          ;30
0033
0033 D5                     PUSH DE             ;33 DEJUMP
0034 C9                     RET
0035
0035 10FE           DELYB:  DJNZ DELYB
0037
0037 C9                     RET
0038
0038 F5                     PUSH AF             ;38H INTERRUPTS
0039 C5                     PUSH BC
003A DBF9                   IN A,(STATPORT)     ;READ IN ABOUT 36Ts (56) 6 (9) USEC
003C 4F                     LD C,A
003D DBFA                   IN A,(250)
003F 47                     LD B,A              ;B=LMPR, C=STATUS
0040 E5                     PUSH HL
0041 3E5F                   LD A,PAGE1F+40H
0043 D3FA                   OUT (250),A         ;BOTH ROMS ON, PAGE ZERO IN SECTIO B
0045 2A705B                 LD HL,(ANYIV)
0048 E9                     JP (HL)
0049
0049 ED73D25A       ANYI:   LD (SPSTORE),SP     ;ARRIVE IN ABOUT 113 (139) T 19 (23)
004D 31004C                 LD SP,INTSTK
0050 CD2DD4                 CALL INTS
0053 F9                     LD SP,HL
0054 D3FA                   OUT (250),A
0056 E1                     POP HL
0057 C1                     POP BC
0058 F1                     POP AF
0059 FB                     EI
005A C9                     RET
005B
005B 00                     DB 0
005C
005C D3FB                   OUT (251),A
005E E9                     JP (HL)
005F
```

```
005F 78          DELBC:     LD A,B
0060 B1                     OR C
0061 0B                     DEC BC
0062 20FB                   JR NZ,DELBC
0064
0064 C9                     RET
0065
0065 00                     NOP
0066
0066             ;NON-MASKABLE INTERRUPT
0066
0066 F5                     PUSH AF
0067 E5                     PUSH HL           ;REGS SAVED IN ORIG PAGE - MAY CORRUPT 4 BYTES
0068                                          ;IF EG SP BEING USED TO CLS
0068 DBFA                   IN A,(250)
006A 67                     LD H,A
006B 3E1F                   LD A,PAGE1F
006D D3FA                   OUT (250),A       ;ROM0 ON, ROM1 OFF, PAGE 0 AT 4000H
006F 7C                     LD A,H
0070 32D95A                 LD (NMILRP),A     ;SAVE ORIG LRPORT STATUS
0073 ED73D75A               LD (NMISP),SP
0077 318851                 LD SP,NMISTK      ;PUT STACK SOMEWHERE SAFE
007A 2AE05A                 LD HL,(NMIV)
007D 7C                     LD A,H
007E B5                     OR L
007F C40500                 CALL NZ,HLJUMP    ;NORMALLY SUPER-BREAK
0082
0082 ED7BD75A               LD SP,(NMISP)
0086 3AD95A                 LD A,(NMILRP)
0089 D3FA                   OUT (250),A
008B E1                     POP HL
008C F1                     POP AF
008D ED45                   RETN
008F
008F EDB0                   LDIR
0091 C9                     RET
0092
0092 EDB8                   LDDR
0094 C9                     RET
0095
0095 EDB1                   CPIR
0097 C9                     RET
0098
0098 EDB9                   CPDR
009A C9                     RET
009B
009B EDB3                   OTIR
009D C9                     RET
009E
009E EDBB                   OTDR
00A0 C9                     RET
00A1
00A1             ;READ, SKIP NUMBER
00A1
00A1 7E          RDCN:      LD A,(HL)
00A2
00A2             ;SKIP 5-BYTE INVIS. NUMBER IN BASIC LINE. IF SKIP, TAKES ABOUT 34 Ts. vs 54 ZX
00A2
00A2 FE0E        NUMBER:    CP 0EH
00A4 C0                     RET NZ          ;RET IF NOT NUMBER MARKER
00A5
00A5 3E06                   LD A,6
00A7 85                     ADD A,L
00A8 6F                     LD L,A
00A9 7E                     LD A,(HL)
00AA D0                     RET NC          ;RET IF HL OK AND A=NEXT CHAR
00AB
00AB 24                     INC H
00AC
00AC 7E          NRREAD:    LD A,(HL)
00AD C9                     RET
00AE
00AE 1A          RDDE:      LD A,(DE)
00AF C9                     RET
00B0
00B0 06FA        MINITH:    LD B,250
00B2
00B2 0B          IDEL:      DEC BC
00B3 78                     LD A,B
00B4 B1                     OR C
00B5 20FB                   JR NZ,IDEL        ;DELAY APPROX 1.2 MSEC *B
00B7
00B7 3E5F                   LD A,40H+PAGE1F
00B9 D3FA                   OUT (250),A       ;ROM1 ON
00BB C3AEEB                 JP MNINIT
00BE
00BE 22975A      NEXTCHAR1: LD (CHAD),HL
00C1
00C1 DBFA        GETCHAR1:  IN A,(250)
00C3 47                     LD B,A
00C4 E6BF                   AND 0BFH          ;FORCE BIT 6 LOW
00C6 D3FA                   OUT (250),A       ;ROM1 OFF
00C8
00C8 7E          GTCH1:     LD A,(HL)
00C9 FE21                   CP 21H
00CB 3806                   JR C,GTCH3         ;JR IF 00-20H
00CD
00CD 0EFA        GTCH2:     LD C,250
00CF ED41                   OUT (C),B         ;ORIG ROM1 STATUS
00D1 C1                     POP BC
00D2 C9                     RET
00D3
00D3 FE0D        GTCH3:     CP 0DH
```

```
00D5 28F6                JR Z,GTCH2          ;RET IF CR
00D7
00D7 23                  INC HL              ;SKIP SPACES AND CONTROL CODES
00D8 22975A              LD (CHAD),HL
00DB 18EB                JR GTCH1
00DD
00DD 2A975A     NXCHAR:  LD HL,(CHAD)
00E0 23                  INC HL
00E1 22975A              LD (CHAD),HL
00E4 7E                  LD A,(HL)
00E5 C9                  RET
00E6
00E6
00E6            ;FLOATING-POINT CALCULATOR
00E6
00E6 DDE3       FPCP2:   EX (SP),IX        ;SAVE PTR IN USE BY ANY CALLING RST 28H ROUTINE
00E8                                       ;MAKE PTR=ADDR AFTER RST 28H
00E8 ED43865B            LD (BCREG),BC
00EC DBFA                IN A,(250)
00EE F5                  PUSH AF           ;ORIG PORT STATUS
00EF F640                OR 40H            ;BIT FOR ROM1=HI (ACTIVE)
00F1 D3FA                OUT (250),A
00F3 CDB7C7              CALL FPCMAIN
00F6 F1                  POP AF
00F7 DDE3                EX (SP),IX        ;GET ORIG IX AS PTR, (SP) PTS. PAST EXIT OR EXIT2
00F9
00F9 ED4B865B   LRPOUT:  LD BC,(BCREG)
00FD D3FA                OUT (250),A
00FF C9                  RET
0100
0100                     DS 0100H-$,0
0100
0100
0100            ;JUMP TABLE AT 0100H
0100
0100 F7                  RST 30H
0101 5652                DW JSCRN-8000H    ;0100
0103 C35E02              JP JSVIN          ;0103
0106 F7                  RST 30H
0107 7768                DW HEAPROOM-8000H ;0106
0109 C3801E              JP WKROOM         ;0109
010C C31C1E              JP MKRBIG         ;010C  OPEN ABC AT HL
010F C3AD19              JP CALBAS         ;010F  CALL BASIC LINE
0112 C36311              JP SETSTRM        ;0112  SET STREAM IN A REG
0115 C3B43D              JP POMSG          ;0115  O/P MSG A FROM LIST AT DE
0118 C3E43A              JP EXPT1NUM       ;0118  EXPECT A NUMERIC EXPR. AT (CHAD)
011B C3A13A              JP EXPTSTR        ;011B  EXPECT A STRING EXPR AT (CHAD)
011E C3083B              JP EXPTEXPR       ;011E  EXPECT AN EXPRESSION AT (CHAD)
0121 C32E1D              JP GETINT         ;0121  UNSTACK WORD FROM CALCULATOR STACK TO
0124                                       ;      BC. HL=BC, A=C
0124 C3011D              JP STKFETCH       ;0124  GET STRING PARAMS. A=START PAGE, DE=START
0127                                       ;      BC=LEN
0127 C3F01C              JP STKSTORE       ;0127  STACK STRING PARAMS
012A C3E43E              JP SBUFFET        ;012A  UNSTACK STRING PARAMS AND COPY TO BUFFER IN
012D                                       ;      SYS PAGE. ERROR IF >255 BYTES
012D
012D C35E2A              JP FARLDIR        ;012D  MOVE (PAGCOUNT/MODCOUNT) BYTES FROM PAGE A,
0130                                       ;      HL TO PAGE C, DE, USING LDIR
0130 C34E2A              JP FARLDDR        ;0130
0133
0133 C3AB29              JP JPUT           ;0133
0136 C31029              JP JGRAB          ;0136
0139 C3A422              JP JPLOT          ;0139
013C C39E21              JP JDRAW          ;013C
013F C31421              JP JDRAWTO        ;013F
0142 C36B20              JP JCIRCLE        ;0142
0145 C3CF24              JP JFILL          ;0145
0148 C3B824              JP JBLITZ         ;0148
014B C35709              JP JROLL          ;014B
014E C3AE06              JP CLSBL          ;014E CLEAR ENTIRE SCREEN IF A=0, ELSE CLEAR WINDOW
0151 C3B506              JP CLSLOWER       ;0151
0154 F7                  RST 30H
0155 5D6E                DW JPALET-8000H   ;0154
0157                                       ;A=LINE (OR FFH IF NONE) B/C=COLOURS, E=PAL. ENTRY.
0157 F7                  RST 30H
0158 8A53                DW JOPSCR-8000H   ;0157
015A
015A F7         MODET:   RST 30H
015B EB56                DW MODPT2-8000H   ;015A
015D F7                  RST 30H
015E 0C72                DW JTCOPY-8000H   ;015D  TEXT COPY
0160 F7                  RST 30H
0161 0B72                DW JGCOPY-8000H   ;0160  GRAPHICS COPY
0163
0163 C3521E              JP RECLAIM2       ;0163
0166 C3BD1C              JP KBFLUSH        ;0166
0169 C3B11C              JP READKEY        ;0169
016C                                       ;READ KEYBOARD, FLUSH BUFFER (INKEY$). Z, NC=NONE
016C                                       ;CY, NZ IF A=KEY
016C C30A05              JP KYIP2          ;016C **
016F                                       ;WAIT FOR A QUEUED KEY IN A.
016F
016F F7                  RST 30H
0170 1B6F                DW BEEPP2-8000H   ;016F
0172                                       ;DO DE-1 CYCLES AT PERIOD HL 8-T UNITS
0172
0172 F7                  RST 30H
0173 0866                DW SABYTES-8000H  ;0172
0175
0175 F7                  RST 30H
0176 0E66                DW LDBYTES-8000H  ;0175
0178
0178 F7         JLDVD:   RST 30H
```

```
0179 A462                  DW LDVD2-8000H    ;0178  LOAD (IF CY) OR VERIFY CDE AT HL DISC/TAPE
017B
017B C34520                JP EDGE2          ;017B  TAPE EDGE TIMER
017E
017E F7          JPFSTRS:  RST 30H
017F 9759                  DW PFSTRS-8000H   ;017E  STR$ OF FPCS TO BUFFER
0181
0181 F7          SENDA:    RST 30H
0182 5D68                  DW SNDA2-8000H    ;0181 SEND BYTE IN A TO PRINTER
0184
0184             ;NEW TO V11
0184
0184 F7                    RST 30H
0185 3072                  DW IMSCSR-8000H   ;0184 SCREEN$
0187
0187 C36527                JP GRCOMP         ;0187 GRAPHIC COPY SR
018A
018A F7          JGTTOK:   RST 30H
018B B056                  DW GETTOKEN-8000H ;018A MATCH FOR A-1 WORDS FROM LIST AT HL+1
018D
018D F7                    RST 30H
018E DB52                  DW JCLSCR-8000H   ;018D
0190
0190             ;MODE 1/MODE 2/MODE 3/MODE 4
0190             ;DOES FULL CLS, SETS UP EXPANSION TABLE IF NEEDED
0190
0190 CD583A      MODECMD:  CALL SYNTAX6      ;INSIST ON A NUMBER
0193
0193 112204                LD DE,0400H+34
0196 CD5F1F                CALL LIMDB        ;ALLOW ORIG OF 1-4, DEC
0199 18BF                  JR MODET
019B
019B 1E30        OTCD:     LD E,30H
019D 83                    ADD A,E
019E
019E DDE5        RST102:   PUSH IX
01A0 E5                    PUSH HL
01A1 D5                    PUSH DE
01A2 C5                    PUSH BC
01A3 2A515C                LD HL,(CURCHL)
01A6 CDC701                CALL HLJPI
01A9 C1                    POP BC
01AA D1                    POP DE
01AB E1                    POP HL
01AC DDE1                  POP IX
01AE C9                    RET
01AF
01AF 08          S16OP:    EX AF,AF'
01B0 F5                    PUSH AF
01B1 01FD39                LD BC,S16OSR
01B4 CD0802                CALL R1OF2        ;CALL S16 O/P (IN ROM0) WITH ROM1 OFF
01B7 F1                    POP AF
01B8 08                    EX AF,AF'
01B9 C9                    RET
01BA
01BA D9          INPUTAD:  EXX
01BB E5                    PUSH HL
01BC 2A515C                LD HL,(CURCHL)
01BF 23                    INC HL
01C0 23                    INC HL
01C1 CDC701                CALL HLJPI
01C4 E1                    POP HL
01C5 D9                    EXX
01C6 C9                    RET
01C7
01C7 5E          HLJPI:    LD E,(HL)
01C8 23                    INC HL
01C9 56                    LD D,(HL)
01CA EB                    EX DE,HL
01CB E9                    JP (HL)
01CC
01CC F7          PRMAIN:   RST 30H
01CD 005C                  DW PROM1-8000H    ;JP MAIN PRINT ROUTINE IN ROM1
01CF
01CF E3          RST30L2:  EX (SP),HL        ;GET CALLER'S ADDRESS IN HL
01D0 F5                    PUSH AF
01D1 7C                    LD A,H
01D2 FE40                  CP 40H
01D4 301C                  JR NC,RST30L4     ;JR IF CALLED FROM OUTSIDE ROM0 - USER
01D6
01D6 ED43475B              LD (BCSTORE),BC
01DA 4E                    LD C,(HL)
01DB 23                    INC HL
01DC 46                    LD B,(HL)
01DD 23                    INC HL
01DE CB78                  BIT 7,B
01E0 2006                  JR NZ,RST30L3     ;JR IF NORMAL 'CALL ROM1'
01E2
01E2 CBF8                  SET 7,B           ;ELSE BIT 7 SHOWED 'JP ROM1'
01E4 F1                    POP AF
01E5 E1                    POP HL            ;ORIG HL - JUNK 1 RET ADDR
01E6 1802                  JR R1ONCLBC
01E8
01E8 F1          RST30L3:  POP AF
01E9 E3                    EX (SP),HL
01EA
01EA 08          R1ONCLBC: EX AF,AF'
01EB DBFA                  IN A,(250)
01ED F5                    PUSH AF
01EE F640                  OR 40H
01F0 181B                  JR R1OFON
01F2
01F2 F1          RST30L4:  POP AF
```

```
01F3 E3                 EX (SP),HL
01F4 E5                 PUSH HL
01F5 2AF25A             LD HL,(RST30V)
01F8 E3                 EX (SP),HL
01F9 C9                 RET             ;'RET' TO VECTOR, ALL REGS INTACT
01FA
01FA            ;CALL PARAM ADDR WITH ROM1 OFF. JUNK CALLING ADDR SO EQUIV. OF 'JUMP'
01FA            ;(ORIG ROM1 STATUS STILL RESTORED, THOUGH)
01FA
01FA E3         R1OFFJP:    EX (SP),HL
01FB 4E                     LD C,(HL)
01FC 23                     INC HL
01FD 46                     LD B,(HL)
01FE E1                     POP HL
01FF 1806                   JR R1OFFCLBC
0201
0201            ;CALL PARAM ADDR WITH ROM1 OFF. CALL CAN COME FROM ANYWHERE
0201            ;ORIG ROM1 STATUS RESTORED AT END
0201
0201 E3         R1OFFCL:    EX (SP),HL
0202 4E                     LD C,(HL)
0203 23                     INC HL
0204 46                     LD B,(HL)
0205 23                     INC HL
0206 E3                     EX (SP),HL      ;BC=PARAMETER
0207
0207            ;R1OFFCLBC. CALL BC WITH ROM1 OFF. ALL REGS CARRIED IN EXCEPT AF' (CORRUPT)
0207            ;ALL REGS CARRIED OUT EXCEPT AF'. ORIG ROM1 STATUS RESTORED AT END
0207
0207 08         R1OFFCLBC: EX AF,AF'
0208
0208 DBFA       R1OF2:      IN A,(250)
020A F5                     PUSH AF         ;ORIG URPORT STATUS
020B E6BF                   AND 0BFH        ;ROM1 BIT OFF
020D
020D D3FA       R1OFON:     OUT (250),A     ;ROM1 OFF/ON
020F 08                     EX AF,AF'
0210 CD1902                 CALL LDBCJP
0213 08                     EX AF,AF'
0214 F1                     POP AF
0215 D3FA                   OUT (250),A
0217 08                     EX AF,AF'
0218 C9                     RET
0219
0219 C5         LDBCJP:     PUSH BC
021A ED4B475B               LD BC,(BCSTORE)
021E C9                     RET
021F
021F
021F            ;TURN ROM 1 OFF, JP (BC)
021F
021F DBFA       R1XJP:      IN A,(250)
0221 E6BF                   AND 0BFH
0223 D3FA                   OUT (250),A
0225 C5                     PUSH BC
0226 C9                     RET
0227
0227            ;STRING O/P - FROM 0013H
0227            ;ENTRY: DE PTS TO DATA, SWITCHED IN. BC=LEN
0227            ;ACTION: DO NOTHING IF BC=0, ELSE PRINT BC BYTES FROM (DE)
0227            ;EXIT: DE PTS TO JUST PAST LAST BYTE ON EXIT, BC=0, HL CORRUPT
0227
0227 D5         SOP2:       PUSH DE
0228 2A515C                 LD HL,(CURCHL)  ;FETCH AN OUTPUT ADDRESS POINTER
022B 5E                     LD E,(HL)
022C 23                     INC HL
022D 56                     LD D,(HL)
022E EB                     EX DE,HL
022F D1                     POP DE          ;DE=SRC, HL=CHANNEL
0230 DBFA                   IN A,(250)
0232 F5                     PUSH AF
0233 E6BF                   AND 0BFH        ;ROM1 OFF
0235 D3FA                   OUT (250),A
0237 7E                     LD A,(HL)
0238 FE40                   CP 40H
023A 2011                   JR NZ,SOP3      ;JR IF SPECIAL STRING OUTPUT NOT PROVIDED BY
023C                                        ;CHANNEL - USE MULTIPLE CALLS OF RST 10H.
023C 23                     INC HL
023D 23                     INC HL
023E 23                     INC HL          ;SKIP '40H, JR XX' TO PT TO STRING O/P ROUTINE
023F CD0500                 CALL HLJUMP
0242 180E                   JR SOP4
0244
0244 1A         SOPL:       LD A,(DE)
0245 D7                     RST 10H
0246 13                     INC DE
0247 7A                     LD A,D
0248 FEC0                   CP 0C0H
024A D4EB3F                 CALL NC,INCURPDE
024D
024D 78         SOP3:       LD A,B
024E B1                     OR C
024F 0B                     DEC BC
0250 20F2                   JR NZ,SOPL
0252
0252 F1         SOP4:       POP AF
0253 D3FA                   OUT (250),A
0255 C9                     RET
0256
0256 C5         R1OFRD:     PUSH BC
0257 CD0102                 CALL R1OFFCL
025A AC00                   DW NRREAD
025C C1                     POP BC
```

```
025D C9                     RET
025E
025E D9          JSVIN:     EXX
025F E1                     POP HL              ;RET ADDR
0260 5E                     LD E,(HL)
0261 23                     INC HL
0262 56                     LD D,(HL)
0263 23                     INC HL
0264 E5                     PUSH HL             ;RET ADDR TO CALLER
0265 4F                     LD C,A              ;SAVE ENTRY A BRIEFLY
0266 DBFA                   IN A,(250)
0268 47                     LD B,A              ;ENTRY LRPORT
0269 3E1F                   LD A,1FH
026B F3                     DI
026C D3FA                   OUT (250),A         ;SYS PAGE IN AT 4000H, ROM0 ON, ROM1 OFF
026E ED73D55A               LD (JVSP),SP
0272 31C04E                 LD SP,ISPVAL-40H
0275 FB                     EI
0276 C5                     PUSH BC
0277 217F02                 LD HL,JSVIN2        ;RET ADDR TO PT 2
027A E5                     PUSH HL
027B D5                     PUSH DE             ;PARAM TO CALL
027C 79                     LD A,C              ;ENTRY A
027D D9                     EXX
027E C9                     RET                 ;TO PARAM ADDR WITH MAIN REGS INTACT
027F                                            ;THEN 'RET' TO VARSIN2
027F 08          JSVIN2:    EX AF,AF'
0280 F1                     POP AF
0281 F3                     DI
0282 ED7BD55A               LD SP,(JVSP)
0286 D3FA                   OUT (250),A         ;ORIG
0288 FB                     EI
0289 08                     EX AF,AF'
028A C9                     RET
028B
028B
028B
028B             ;INITIAL STREAM DISPLACEMENTS
028B
028B 1A1501060B STRMTAB:    DB 26,21,1,6,11     ;B,$,K,S,R (FIXED) FOR STREAMS FB,FC,FD,FE,FF
0290 01010610               DB 1,1,6,16         ;K,K,S,P FOR STREAMS 0,1,2,3
0294
0294
0294                 ;MAIN FILE - MAIN.SAM
0294                        INCLUDE VARS.SAM
0294
0294             ;VARS.SAM
0294
0294 5A00=       VAR2       EQU 5A00H    ;MUST START AT PAGE BOUNDARY
0294
0294             ;NEXT 18 BYTES INITED TOGETHER FROM 'CHIT'
0294
0294 5A00=       LNCUR      EQU VAR2+00H ;CURRENT LINE CURSOR CHAR (USUALLY '>')
0294 5A01=       KURCHAR    EQU VAR2+01H ;(2) CURSOR CHARACTERS - LOWER CASE/UPPER CASE
0294 5A03=       BIN1DIG    EQU VAR2+03H ;USED BY BIN$ - USUALLY '1'
0294 5A04=       BIN0DIG    EQU VAR2+04H ;               USUALLY '0'
0294 5A05=       INSTHASH   EQU VAR2+05H ;NORMALLY '#'
0294 5A06=       PSLD       EQU VAR2+06H ;(2) DEVICE LETTER/NUMBER
0294                                     ;NUMBER=TAPE SPEED/DISC NUMBER/NET STATION
0294 5A08=       SPEEDINK   EQU VAR2+08H ;INK FLASH COUNTER RELOAD VALUE
0294 5A09=       LINIPTR    EQU VAR2+09H ;(2) PTR TO LINE INTERRUPT PAL CHNG TABLE
0294 5A0B=       XCMDP      EQU VAR2+0BH ;(3) PAGE/ADDR OF FIRST EXTERNAL CMD LIST, OR FFXXXX
0294 5A0E=       PRRHS      EQU VAR2+0EH ;PRINTER RHS LIMIT - 79
0294 5A0F=       AFTERCR    EQU VAR2+0FH ;0A OR NUL ACCORDING TO WHETHER AUTO LF NEEDED
0294 5A10=       LPTPRT1    EQU VAR2+10H ;(2) PRINTER CONTROL PORT/01H STROBE VALUE
0294                  ;ALL reserved for DUMP
0294 5A2F=       TABVAR     EQU VAR2+2FH ;0 IF TAB=16, ELSE TAB=8
0294 5A30=       M23LSC     EQU VAR2+30H ;(2) M2/3 LOWER SCREEN COLOURS
0294 5A32=       SOFE       EQU VAR2+32H ;FLAG FOR SCREEN OFF ENABLE/DISABLE 0=ON
0294 5A33=       TPROMPTS   EQU VAR2+33H ;BIT 0=1 TO SUPPRESS PRINTED NAMES DURING LOAD
0294                                     ;BIT 1=1 TO SUPPRESS PROMPTS DURING SAVE
0294
0294             ;START OF BLOCK SAVED WITH SWITCHED-OUT SCREEN
0294
0294 5A34=       BGFLG      EQU VAR2+34H ;BLOCK GRAPHICS FLAG
0294 5A35=       FL6OR8     EQU VAR2+35H ;00=6 BIT CHARS IN MODE 2, NZ=8 BIT
0294
0294 5A36=       CSIZE      EQU VAR2+36H ;(2) HEIGHT/WIDTH
0294
0294 5A38=       UWRHS      EQU VAR2+38H ;STARTS AT 31
0294 5A39=       UWLHS      EQU VAR2+39H ;STARTS AT 0
0294 5A3A=       UWTOP      EQU VAR2+3AH ;STARTS AT 0
0294 5A3B=       UWBOT      EQU VAR2+3BH ;STARTS AT 18 (19 LINES IN UPPER, 2 IN LOWER SCR, 9 PIX)
0294
0294             ;LOWER WINDOW
0294
0294 5A3C=       LWRHS      EQU VAR2+3CH
0294 5A3D=       LWLHS      EQU VAR2+3DH
0294 5A3E=       LWTOP      EQU VAR2+3EH
0294 5A3F=       LWBOT      EQU VAR2+3FH ;STARTS AT 20
0294
0294 5A40=       MODE       EQU VAR2+40H
0294 5A41=       YCOORD     EQU VAR2+41H ;0-191, 0 AT TOP
0294 5A42=       XCOORD     EQU VAR2+42H ;(2) 0-255 (FAT) OR 0-511 (THIN)
0294 5A42=       RLINE      EQU XCOORD
0294
0294             ;PERMANENT GRAPHICS/PRINT VARS
0294
0294 5A44=       THFATP     EQU VAR2+44H ;00=THIN, NZ=FAT PIXELS
0294 5A45=       ATTRP      EQU VAR2+45H ;ATTR USED BY MODES 0 AND 1
0294 5A46=       MASKP      EQU VAR2+46H
0294 5A47=       PFLAGP     EQU VAR2+47H ;BIT 4=PAPER 9, BIT 6=INK 9
```

```
0294 5A48=      M23PAPP    EQU VAR2+48H ;NIBBLES OR DOUBLE BITS MATCH
0294 5A49=      M23INKP    EQU VAR2+49H ;00-FF. NIBBLES MUST MATCH UNLESS A DOTTED LINE WANTED.
0294                                    ;MUST BE NEXT TO OVER FOR LD DE,(M3INKT)
0294 5A4A=      OVERP      EQU VAR2+4AH ;0-1. NORMAL OR OVER 1 (OR-ING)
0294 5A4B=      INVERP     EQU VAR2+4BH ;00/FF FOR NORMAL/INVERSE  ;
0294 5A4C=      GOVERP     EQU VAR2+4CH ;0-3. NORMAL, XOR, OR, AND ;THESE 2 LINKED
0294                                    ;(GRAPHICS OVER - USED BY PUT)
0294
0294            ;TEMPORARY GRAPHICS/PRINT VARS
0294
0294 5A4D=      THFATT     EQU VAR2+4DH ;COPIED FROM THFATP WHEN MODE=2, ELSE SET TO NZ (FAT)
0294 5A4E=      ATTRT      EQU VAR2+4EH
0294 5A4F=      MASKT      EQU VAR2+4FH
0294 5A50=      PFLAGT     EQU VAR2+50H ;BIT 4=PAPER 9, BIT 6=INK 9
0294 5A51=      M23PAPT    EQU VAR2+51H ;NIBBLES OR DOUBLE BITS MATCH
0294 5A52=      M23INKT    EQU VAR2+52H ;00-FF. NIBBLES MUST MATCH UNLESS A DOTTED LINE WANTED
0294 5A53=      OVERT      EQU VAR2+53H ;0-1. NORMAL OR OVER 1 (OR-ING)
0294 5A54=      INVERT     EQU VAR2+54H ;00/FF FOR NORMAL/INVERSE    ;
0294 5A55=      GOVERT     EQU VAR2+55H ;0-3. NORMAL, XOR, OR, AND   ;LINKED
0294
0294            ;CURRENT WINDOW (TEMP)
0294
0294 5A56=      WINDRHS    EQU VAR2+56H
0294 5A57=      WINDLHS    EQU VAR2+57H
0294 5A58=      WINDTOP    EQU VAR2+58H
0294 5A59=      WINDBOT    EQU VAR2+59H
0294
0294 5A5A=      WINDMAX    EQU VAR2+5AH ;(2) U. WINDOW LOWEST BOTTOM/MAX RHS
0294 5A5C=      ORGOFF     EQU VAR2+5CH ;GRAPHICS ORIGIN OFFSET
0294 5A5D=      LSOFF      EQU VAR2+5DH ;LOWER SCREEN BIT OFFSET ** THESE 3 MOVED HERE
0294                       .14 SPARE
0294 5A6C=      SPOSNU     EQU VAR2+6CH ;(2) SCREEN POSN (UPPER) WINDLHS, WINDTOP AFTER CLS
0294 5A6E=      SPOSNL     EQU VAR2+6EH ;(2) SCREEN POSN (LOWER) 0,19 AFTER CLS
0294
0294            ;END OF BLOCK SAVED WITH SCREEN
0294
0294 5A70=      PRPOSN     EQU VAR2+70H ;(2) PRINTER POSN (1 BYTE EXTRA FOR CONVENIENT LD)
0294 5A72=      OPCHAR     EQU VAR2+72H ;USED BY LPRINT - CURRENT O/P CHAR
0294 5A73=      DEVICE     EQU VAR2+73H ;0=US, 1=LS, 2=PRINTER, 3=
0294 5A74=      CLET       EQU VAR2+74H ;CURRENT CHANNEL LETTER K/S/P/B/T/$ ETC
0294 5A75=      IFTYPE     EQU VAR2+75H ;LONG/SHORT IF
0294 5A76=      REFFLG     EQU VAR2+76H ;Z IF REF VAR BEING WORKED ON
0294 5A77=      CURDISP    EQU VAR2+77H ;CURRENT DISPLAY
0294 5A78=      CUSCRNP    EQU VAR2+78H ;CURRENT SCREEN PAGE
0294 5A79=      CURP       EQU VAR2+79H ;CURRENT UPPER RAM PORT
0294 5A7A=      CLRP       EQU VAR2+7AH ;CURRENT LOWER RAM PORT (TEMP STORES DURING PAGING)
0294 5A7B=      CSA        EQU VAR2+7BH ;(2) CURRENT STATEMENT ADDR
0294 5A7D=      FIRST      EQU VAR2+7DH ;(2)
0294 5A7F=      LAST       EQU VAR2+7FH ;(2) LINE NUMBERS IN EG LIST X TO Y. ALSO USED BY
0294                                    ;ARRAY SLICER
0294
0294            ;POINTERS THAT ARE ADJUSTED BY MAKEROOM, RECLAIM:
0294
0294 5A81=      SAVARSP    EQU VAR2+81H
0294 5A82=      SAVARS     EQU VAR2+82H ;(2)     ;SAVARS/NUMEND/NVARS MUST BE IN ORDER
0294
0294 5A84=      NUMENDP    EQU VAR2+84H          ;NUMEND/NVARS/DATADD MUST BE IN ORDER
0294 5A85=      NUMEND     EQU VAR2+85H ;(2)
0294
0294 5A87=      NVARSP     EQU VAR2+87H
0294 5A88=      NVARS      EQU VAR2+88H ;(2)
0294
0294 5A8A=      DATADDP    EQU VAR2+8AH
0294 5A8B=      DATADD     EQU VAR2+8BH ;(2)
0294
0294 5A8D=      WKENDP     EQU VAR2+8DH
0294 5A8E=      WKEND      EQU VAR2+8EH ;(2)
0294
0294 5A90=      WORKSPP    EQU VAR2+90H
0294 5A91=      WORKSP     EQU VAR2+91H ;(2)
0294
0294 5A93=      ELINEP     EQU VAR2+93H
0294 5A94=      ELINE      EQU VAR2+94H ;(2)
0294
0294 5A96=      CHADP      EQU VAR2+96H
0294 5A97=      CHAD       EQU VAR2+97H ;(2)
0294
0294 5A99=      KCURP      EQU VAR2+99H
0294 5A9A=      KCUR       EQU VAR2+9AH ;(2)
0294
0294 5A9C=      NXTLINEP   EQU VAR2+9CH
0294 5A9D=      NXTLINE    EQU VAR2+9DH  ;(2)
0294
0294 5A9F=      PROGP      EQU VAR2+09FH
0294 5AA0=      PROG       EQU VAR2+0A0H ;(2)
0294
0294 5AA2=      XPTRP      EQU VAR2+0A2H
0294 5AA3=      XPTR       EQU VAR2+0A3H ;(2)
0294
0294 5AA5=      DESTP      EQU VAR2+0A5H
0294 5AA6=      DEST       EQU VAR2+0A6H ;(2)
0294
0294 5AA8=      PRPTRP     EQU VAR2+0A8H
0294 5AA9=      PRPTR      EQU VAR2+0A9H ;(2) PROC POINTER
0294
0294            ;END OF ADJUSTED PTRS
0294
0294 5AAB=      DPPTRP     EQU VAR2+0ABH
0294 5AAC=      DPPTR      EQU VAR2+0ACH ;(2) DEF PROC POINTER
0294
0294 5AAE=      CLAPG      EQU VAR2+0AEH
0294 5AAF=      CLA        EQU VAR2+0AFH ;(2)
0294
```

```
0294 5AB1=      DFTFB       EQU VAR2+0B1H ;**
0294 5AB2=      STRNO       EQU VAR2+0B2H ;CURRENT STREAM NO.
0294 5AB3=      LDCO        EQU VAR2+0B3H ;LD ZX CODE OFFSET (PAGES)
0294                        SPARE
0294 5AB5=      OPSTORE     EQU VAR2+0B5H ;(2)
0294 5AB7=      DMPFG       EQU VAR2+0B7H ;IF NZ PRINT O/P DUMPED
0294 5AB8=      LISTFLG     EQU VAR2+0B8H ;0/1/2 FOR LIST FORMAT 0/1/2
0294 5AB9=      LSTFT       EQU VAR2+0B9H ;TEMPORARY VERSION OF LISTFLG USED BY CHANNEL 'R'
0294 5ABA=      INQUFG      EQU VAR2+0BAH ;IN QUOTES FLAG. BIT 0=1 IF IN QUOTES. OUTLINE ZEROS
0294                                      ;BIT SO INITIAL STATE IS 'OUTSIDE' AND TOKENS PRINTED
0294                                      ;EXCEPT IN QUOTES. PRINT SETS TO 1 SO ALWAYS UDGS.
0294 5ABB=      SPROMPT     EQU VAR2+0BBH ;IF NZ NO 'SCROLL?' PROMPTS
0294 5ABC=      OLDSPCS     EQU VAR2+0BCH ;SPACE STATUS OF PREVIOUS LINE **
0294 5ABD=      INDOPFG     EQU VAR2+0BDH ;INDENTED O/P FLAG
0294
0294 5ABE=      NXTSPCS     EQU VAR2+0BEH
0294 5ABF=      CURSPCS     EQU VAR2+0BFH
0294 5AC0=      NXTHSPCS    EQU VAR2+0C0H
0294 5AC1=      CURTHSPCS   EQU VAR2+0C1H
0294
0294 5AC2=      KPOS        EQU VAR2+0C2H ;(2) CURSOR SCREEN POSN
0294
0294            ;NEXT 4 MUST STAY IN ORDER
0294
0294 5AC4=      SOFFCT      EQU VAR2+0C4H ;COUNTER FOR SCREEN OFF
0294 5AC5=      SOFLG       EQU VAR2+0C5H ;FLAG FOR 'SCREEN HAS BEEN TURNED OFF' (NZ) OR ON (Z)
0294 5AC6=      SPEEDIC     EQU VAR2+0C6H ;COUNTER FOR FLASHING INKS
0294 5AC7=      PALFLAG     EQU VAR2+0C7H ;BIT 0 SHOWS WHICH PAL TABLE IN USE.
0294
0294 5AC8=      TEMPW1      EQU VAR2+0C8H ;(2)
0294 5ACA=      TEMPW2      EQU VAR2+0CAH ;(2)
0294 5ACC=      TEMPW3      EQU VAR2+0CCH ;(2)
0294 5ACE=      TEMPB1      EQU VAR2+0CEH
0294 5ACF=      TEMPB2      EQU VAR2+0CFH
0294 5AD0=      TEMPB3      EQU VAR2+0D0H ;USED FOR FINAL FILL PARAM
0294
0294            ;SYSTEM PAGE ONLY
0294
0294 5AD1=      LASTSTAT    EQU VAR2+0D1H ;STATPORT VALUE ON LAST INTERRUPT
0294 5AD2=      SPSTORE     EQU VAR2+0D2H ;(2) SP STORE EXCLUSIVE TO INTERRUPTS
0294
0294 5AD5=      JVSP        EQU VAR2+0D5H ;(2) JSVIN SP STORE
0294 5AD7=      NMISP       EQU VAR2+0D7H ;(2) NMI SP STORE
0294 5AD9=      NMILRP      EQU VAR2+0D9H ;(1) LRPORT VALUE WHEN NMI OCCURRED
0294
0294 5ADA=      VECTBS      EQU VAR2+0DAH ;VECTOR BASE ADDR (0) USED BY LINK ROUTINE
0294 5ADA=      DMPV        EQU VAR2+0DAH ;(2)
0294 5ADC=      SETIYV      EQU VAR2+0DCH ;(2)
0294 5ADE=      PRTOKV      EQU VAR2+0DEH ;(2)
0294 5AE0=      NMIV        EQU VAR2+0E0H ;(2)
0294 5AE2=      FRAMIV      EQU VAR2+0E2H ;(2)
0294 5AE4=      LINIV       EQU VAR2+0E4H ;(2)
0294 5AE6=      COMSV       EQU VAR2+0E6H ;(2)
0294 5AE8=      MIPV        EQU VAR2+0E8H ;(2)
0294 5AEA=      MOPV        EQU VAR2+0EAH ;(2)
0294 5AEC=      EDITV       EQU VAR2+0ECH ;(2)
0294 5AEE=      RST8V       EQU VAR2+0EEH ;(2)
0294 5AF0=      RST28V      EQU VAR2+0F0H ;(2)
0294 5AF2=      RST30V      EQU VAR2+0F2H ;(2)
0294 5AF4=      CMDV        EQU VAR2+0F4H ;(2)
0294 5AF6=      EVALUV      EQU VAR2+0F6H ;(2)
0294 5AF8=      LPRTV       EQU VAR2+0F8H ;(2)
0294 5AFA=      MTOKV       EQU VAR2+0FAH ;(2)
0294 5AFC=      MOUSV       EQU VAR2+0FCH ;(2)
0294 5AFE=      KURV        EQU VAR2+0FEH ;(2)
0294
0294 5B00=      CEXTAB      EQU VAR2+0100H ;(32) COLOUR IS APPLIED TO THIS DATA, SO EG. F0F0
0294                                       ;MIGHT BECOME A3A3, OR IF INVERSE, 3A3A.
0294
0294 5B20=      EXTAB       EQU VAR2+0120H ;(32) 16 WORDS OF MODE 3 PRINT NIBBLE->WORD DATA.
0294                                       ;(OR 16 BYTES OF MODE 2 DOUBLED DATA)
0294                                       ;EACH WORD (BYTE) IS THE EXPANSION OF A NIBBLE, SO
0294                                       ;E.G. ENTRY 0A BIN 1010 =1111000011110000 OR 11001100
0294
0294 5B40=      COMPFLG     EQU VAR2+0140H ;FLAG BITS USED BY LABEL/FN/PROC COMPILER
0294 5B41=      BREAKDI     EQU VAR2+0141H ;NZ IF BREAK BETWEEN STATEMENTS DISABLED
0294 5B42=      ERRSTAT     EQU VAR2+0142H ;
0294 5B43=      ERRLN       EQU VAR2+0143H ;(2) LINE TO GOTO ON ERROR
0294 5B45=      ONERRFLG    EQU VAR2+0145H ;BIT 7=TEMP ON, BIT 0=PERM ON
0294 5B46=      ONSTORE     EQU VAR2+0146H ;ON CMD'S STATEMENT NO.
0294 5B47=      BCSTORE     EQU VAR2+0147H ;(2) USED BY RST 30H
0294 5B49=      M3PAPP      EQU VAR2+0149H ;(2)
0294 5B4B=      M3LSC       EQU VAR2+014BH ;(2)
0294 5B4D=      TEMPW4      EQU VAR2+014DH ;(2) USED BY POINTERS
0294 5B4F=      TEMPW5      EQU VAR2+014FH ;(2) USED BY POINTERS
0294                        ;1 SPARE
0294 5B52=      LPT         EQU VAR2+0152H ;(30) 1 BYTE PRE SCREEN LINE, MARKED IF HAS A
0294                                       ;LINE NUMBER ON IT
0294
0294 5B70=      ANYIV       EQU VAR2+0170H ;ANY INTERRUPT VECTOR
0294 5B72=      RNSTKE      EQU VAR2+0172H ;(2) RENAME STACK PTR (PARPRO)
0294 5B74=      CURCMD      EQU VAR2+0174H ;CODE OF CMD BEING EXECUTED
0294 5B75=      LTDFF       EQU VAR2+0175H ;LET/DEFAULT FLAG
0294 5B76=      STRM16NM    EQU VAR2+0176H ;(11) TLBYTE/NAME OF VAR THAT STREAM 16 WRITES TO.
0294 5B81=      GRARF       EQU VAR2+0181H ;GRAPHICS RECORD FLAG (0=OFF)
0294 5B82=      DHADJ       EQU VAR2+0182H ;DOUBLE HEIGHT ADJ. 0 UNLESS BOTTOM OF DH CHAR O/PED
0294 5B83=      PAGCOUNT    EQU VAR2+0183H ;PAGE COUNTER USED BY FARLDIR
0294 5B84=      MODCOUNT    EQU VAR2+0184H ;(2) MOD 16K COUNTER USED BY FARLDIR
0294 5B86=      BCREG       EQU VAR2+0186H ;(2) FPC'S BC REG
0294 5B88=      AUTOFLG     EQU VAR2+0188H
0294 5B89=      AUTOSTEP    EQU VAR2+0189H ;(2)
0294 5B89=      RSTEP       EQU AUTOSTEP   ;RENUM STEP
0294
```

```
0294 5B8B=      LSPTR      EQU VAR2+018BH ;(2) LINE SCAN PTR
0294 5B8D=      LNPTR      EQU VAR2+018DH ;(1)
0294 5B8E=      MSEDP      EQU VAR2+018EH ;(8) 018E-0195
0294 5B8F=      BUTSTAT    EQU VAR2+018FH ;MOUSE BUTTON STATUS
0294
0294 5B96=      MXCRD      EQU VAR2+0196H ;(2) MOUSE X COORD
0294 5B98=      MYCRD      EQU VAR2+0198H ;(2) MOUSE Y COORD
0294
0294            ;USED BY PRINTFP:
0294
0294 5B9A=      FRACLIM    EQU VAR2+019AH ;
0294 5B9B=      NPRPOS     EQU VAR2+019BH ;(2)
0294 5B9D=      DIGITS     EQU VAR2+019DH ;   MUST
0294 5B9E=      EPOWER     EQU VAR2+019EH ;   ALL
0294 5B9F=      DECPNTED   EQU VAR2+019FH ;   STAY IN THIS ORDER!
0294 5BA0=      PRNBUFF    EQU VAR2+01A0H ;(16) ALLOWS -0.0000123456789
0294                                      ;     OR     -1.2345678E-35
0294 5BB0=      BCDBUFF    EQU VAR2+01B0H ;(5)
0294
0294 5BB5=      OTHER      EQU VAR2+01B5H ;NET DESTINATION STATION NUMBER
0294 5BB6=      DCT        EQU VAR2+01B6H ;DISC ERROR COUNTER
0294
0294 5BB7=      SLDEV      EQU VAR2+01B7H ;(2) DEVICE LETTER/NUMBER (TEMP)
0294                                      ;NUMBER=TAPE SPEED/DISC NUMBER/NET STATION
0294 5BB9=      OVERF      EQU VAR2+01B9H ;'SAVE OVER' FLAG. 0 IF SAVE OVER, ELSE NZ
0294 5BBA=      INSLV      EQU VAR2+01BAH ;(2)
0294 5BBC=      STRLOCN    EQU VAR2+01BCH ;(2) USED BY LOOKVARS
0294 5BBE=      TVDATA     EQU VAR2+01BEH ;(2)
0294 5BC0=      DOSER      EQU VAR2+01C0H ;(2) JUMP AFTER DOS EXECUTION
0294 5BC2=      DOSFLG     EQU VAR2+01C2H ;Z IF NO DOS LOADED
0294 5BC3=      DOSCNT     EQU VAR2+01C3H ;BIT 0 IS SET IF DOS IN CONTROL
0294 5BC4=      BSTKEND    EQU VAR2+01C4H ;(2)
0294
0294            ;NEXT 26 BYTES INITED TOGETHER FROM MAIT
0294
0294 5BC6=      BASSTK     EQU VAR2+01C6H ;(2)
0294 5BC8=      HEAPEND    EQU VAR2+01C8H ;(2)
0294 5BCA=      HPST       EQU VAR2+01CAH ;(2)
0294 5BCC=      FPSBOT     EQU VAR2+01CCH ;(2) START OF FPCS
0294 5BCE=      DKDEF      EQU VAR2+01CEH ;(2) KEY DEFS
0294 5BD0=      DKLIM      EQU VAR2+01D0H ;(2) LIMIT OF DEF KEY BUFFER
0294 5BD2=      PATOUT     EQU VAR2+01D2H ;(2) ADDR OF 'PRINTABLE CHARS' O/P
0294 5BD4=      ERRMSGS    EQU VAR2+01D4H ;(2)
0294 5BD6=      UMSGS      EQU VAR2+01D6H ;(2)
0294 5BD8=      KBTAB      EQU VAR2+01D8H ;(2)
0294 5BDA=      CMDADDRT   EQU VAR2+01DAH ;(2) START OF CMD ADDR TABLE IN ROM0
0294 5BDC=      MNOP       EQU VAR2+01DCH ;(2) ADDR OF MAIN O/P ROUTINE
0294 5BDE=      MNIP       EQU VAR2+01DEH ;(2) ADDR OF MAIN I/P ROUTINE
0294
0294 5BE0=      PAGER      EQU VAR2+01E0H ;(14) RESERVED FOR PAGING S.R.
0294
0294 5BEE=      KBUFF      EQU VAR2+01EEH ;(18) - 2 TABLES OF 72 BITS
0294
0294 5C00=      LHM1       EQU 5C00H ;USED BY KEYSCAN (AS 'LASTH-1')
0294            ;LASTH      EQU 5C01H ;LAST KEY HIT. 0 IF NO KEY. STOPS CHANGING IF KEYS
0294                                  ;ARE NOT BEING READ, WHEN BUFFER FILLS. RES 5,(FLAGS)
0294                                  ;IS SEEN AS A READ.
0294 5C02=      KDATA      EQU 5C02H
0294 5C03=      LKPB       EQU 5C03H ;(2)
0294 5C05=      REPCT      EQU 5C05H
0294 5C06=      LASTKV     EQU 5C06H ;(2)
0294 5C08=      LASTK      EQU 5C08H ;KEY FROM BUFFER QUEUE HEAD. KEEPS LAST KEY VALUE.
0294                                  ;NEEDS PERIODIC RESETS OF BIT 5,(FLAGS) OR BUFFER FILLS.
0294 5C09=      REPDEL     EQU 5C09H
0294 5C0A=      REPPER     EQU 5C0AH
0294                       ;1 SPARE
0294 5C10=      STREAMS    EQU 5C10H ;(42 - USES 5C0C-5C35 FOR STREAMS -5 TO 15. -4=16)
0294 5C36=      CHARS      EQU 5C36H ;(2)
0294 5C38=      RASP       EQU 5C38H
0294 5C39=      PIP        EQU 5C39H
0294 5C3A=      ERRNR      EQU 5C3AH
0294 5C3B=      FLAGS      EQU 5C3BH
0294 5C3C=      TVFLAG     EQU 5C3CH
0294 5C3D=      ERRSP      EQU 5C3DH ;(2)
0294 5C3F=      LISTSP     EQU 5C3FH ;(2)
0294
0294 5C42=      NEWPPC     EQU 5C42H ;(2)
0294 5C44=      NSPPC      EQU 5C44H
0294 5C45=      PPC        EQU 5C45H ;(2)
0294 5C47=      SUBPPC     EQU 5C47H
0294 5C48=      BORDCR     EQU 5C48H ;ATTRIBUTES FOR LOWER SCREEN IN MODES 1/2
0294 5C49=      EPPC       EQU 5C49H ;(2)
0294 5C4B=      BORDCOL    EQU 5C4BH ;VALUE TO SEND TO BORDER PORT
0294 5C4F=      CHANS      EQU 5C4FH ;(2)
0294 5C51=      CURCHL     EQU 5C51H ;(2)
0294 5C53=      DEFADDP    EQU 5C53H
0294 5C54=      DEFADD     EQU 5C54H ;(2)
0294 5C56=      NLASTH     EQU 5C56H ;(3)
0294                       ;8 SPARE
0294
0294 5C61=      ZIPLIB     EQU 5C61H ;(2) SIMON N. GOODWIN'S
0294 5C63=      ZIPTEMP    EQU 5C63H ;(2) COMPILER VARS
0294 5C65=      STKEND     EQU 5C65H
0294 5C67=      KPFLG      EQU 5C67H  ;FUNCTION KEYS IF EVEN, NUMBER PAD IF ODD
0294 5C68=      MEM        EQU 5C68H
0294 5C6A=      FLAGS2     EQU 5C6AH
0294 5C6C=      SDTOP      EQU 5C6CH
0294 5C6E=      OLDPPC     EQU 5C6EH
0294 5C70=      OSPPC      EQU 5C70H
0294 5C71=      FLAGX      EQU 5C71H
0294 5C72=      STRLEN     EQU 5C72H
0294 5C76=      SEED       EQU 5C76H
0294 5C78=      FRAMES     EQU 5C78H  ;(3)
```

```
0294 5C7B=      UDG         EQU 5C7BH  ;(2)
0294 5C7D=      HUDG        EQU 5C7DH  ;(2)
0294 5C7F=      FRAMES34    EQU 5C7FH  ;(2)
0294 5C82=      OLDPOS      EQU 5C82H
0294 5C8C=      SCRCT       EQU 5C8CH
0294
0294 5C8D=      KBQB        EQU 5C8DH  ;(8) KEYBOARD QUEUE ** MOVED HERE
0294 5C95=      KBQP        EQU 5C95H  ;(2) KEYBOARD QUEUE POINTERS. LOW=END, HI=HEAD
0294                            ;6 SPARE
0294 5C9D=      SCPTR       EQU 5C9DH  ;(2) ADDR OF CURRENT SCREEN IN SCLIST
0294
0294            ;NOT CLEARED BY NEW:
0294
0294 5C9F=      FISCRNP     EQU 5C9FH  ;PAGE OF SCREEN 1
0294 5CA0=      SCLIST      EQU 5CA0H  ;(16) SCREENS LIST. MODE/PAGE OF SCREENS 1-16, OR FFH
0294 5CB0=      LASTPAGE    EQU 5CB0H  ;LAST PAGE RESERVED BY BASIC
0294 5CB1=      RAMTOPP     EQU 5CB1H
0294 5CB2=      RAMTOP      EQU 5CB2H  ;(2)
0294 5CB4=      PRAMTP      EQU 5CB4H  ;LAST PAGE PRESENT IN MACHINE
0294
0294                 ;SPARE
0294
0294 00C4=      KEYWNO      EQU 0C4H
0294 0070=      TSPEED      EQU 112
0294
0294 003B=      PITOK       EQU 3BH
0294 0021=      PI          EQU PITOK-1AH  ;SEE BELOW
0294 004A=      INSTOK      EQU 4AH
0294 0030=      INSTR       EQU INSTOK-1AH ;USED TO SEPARATE N/$ IMMED CODES
0294 0042=      FNTOK       EQU 42H ;USED BY TPRINT
0294 0043=      BINTOK      EQU 43H ;USED BY TPRINT AND CALC5BY
0294 004C=      SCRNTOK     EQU 4CH ;USED BY SAVE/LOAD
0294 0053=      SINTOK      EQU 53H ;USED BY TPRINT
0294 0060=      INTOK       EQU 60H ;
0294 006C=      CODETOK     EQU 6CH ;USED BY SAVE/LOAD
0294 0070=      CHRSTOK     EQU 70H ;USED BY COPY
0294 007A=      MODTOK      EQU 7AH ;USED BY TPRINT
0294 0080=      ANDTOK      EQU 80H ;USED BY TPRINT
0294
0294 0085=      USINGTOK    EQU 085H
0294 0087=      ATTOK       EQU 087H
0294 0088=      TABTOK      EQU 088H
0294 008A=      WHILETOK    EQU 08AH
0294 008B=      UNTILTOK    EQU 08BH
0294 008C=      LINETOK     EQU 08CH
0294 008D=      THENTOK     EQU 08DH
0294 008E=      TOTOK       EQU 08EH
0294 008F=      STEPTOK     EQU 08FH
0294
0294            ;FORMATTOK  EQU 091H
0294            ;ERASETOK   EQU 092H
0294 0094=      SAVETOK     EQU 094H
0294 0095=      LOADTOK     EQU 095H
0294 0096=      MERGETOK    EQU 096H
0294 0097=      VERIFYTOK   EQU 097H
0294            ;RECORDTOK  EQU 0EFH
0294
0294
0294 0080=      BTHK        EQU 128 ;DOS BOOT (DOS CAN IGNORE, OR INTERPRET AS ALHK)
0294
0294              ;IX=HDR, IX+50H=HDL
0294
0294 0081=      FOPHK       EQU 129 ;DOS OPEN (GET HEADER)
0294 0082=      LDHK        EQU 130 ;DOS LOAD
0294 0083=      VFYHK       EQU 131 ;DOS VERIFY
0294 0084=      SVHK        EQU 132 ;DOS SAVE
0294
0294 0086=      OSHK        EQU 134 ;DOS OPEN STREAM
0294 0087=      CSHK        EQU 135 ;DOS CLOSE STREAM (HDR+1)=LETTER
0294 0088=      ALHK        EQU 136 ;DOS LOAD AUTO-LOAD FILE - FOLLOWS BOOT
0294 0089=      DIRHK       EQU 137
0294
0294
0294 008B=      DVHK        EQU 139 ;DOS DVAR
0294 008C=      EOFHK       EQU 140 ;DOS EOF
0294 008D=      PTRHK       EQU 141 ;DOS PTR
0294 008E=      PATHHK      EQU 142 ;DOS PATH$
0294
0294 00E0=      COMM        EQU 224 ;DISC PORTS
0294 00E1=      TRCK        EQU 225
0294 00E2=      SECT        EQU 226
0294 00E3=      DTRQ        EQU 227
0294 0009=      DRES        EQU 9
0294 0059=      STPIN       EQU 59H
0294 0079=      STPOUT      EQU 79H
0294 0080=      DRSEC       EQU 80H
0294
0294
0294            ;SAVE/LOAD EQUATES
0294
0294 000F=      HFG         EQU 15            ;DISP TO HEADER FLAG
0294 001A=      HDT         EQU 26            ;DISP TO HEADER DATE/TIME
0294 001F=      HDN         EQU 31            ;DISP TO HEADER NUMBERS
0294 0050=      HDRL        EQU 80            ;HDR BUFFER LEN
0294 000A=      NMLEN       EQU 10            ;MAX FILE NAME LEN
0294
0294 0039=      YOSDISP     EQU 57
0294 0043=      YRGDISP     EQU 67
0294 004D=      XOSDISP     EQU 77
0294 0057=      XRGDISP     EQU 87
0294
0294 E003=      RSBUFF      EQU 0E003H
0294 8000=      SBO         EQU 8000H
```

```
0294 C000=      SBN         EQU 0C000H
0294
0294 4000=      HPEND       EQU 4000H          ;HEAPEND
0294
0294                                           ;ROOM FOR DO/LOOP/PROC STACK
0294
0294 4AFF=      BSTACK      EQU 4AFFH          ;BASSTK
0294
0294 4B00=      HDR         EQU 4B00H          ;HEADER LEN=50H. ALSO USED FOR PARPRO RENAME STK
0294 4B50=      HDL         EQU 4B50H          ;ADDRESS OF LOADED HEADER BUFFER
0294
0294 4C00=      INTSTK      EQU 4C00H          ;USES DOWN TO 49EEH NORMALLY
0294 4C00=      BUFF256     EQU 4C00H          ;MUST BE PAGE-ALIGNED
0294 4D00=      FPSB        EQU 4D00H          ;FPSBOT
0294 4D00=      CDBUFF      EQU 4D00H          ;CODE BUFFER FOR E.G. MULTI-LDI. MAX LEN=0181H
0294                                           ;MULTI RRD, RLD, LDD (ROLL, SCROLL, CLS)
0294
0294                                           ;STACK USED DOWN TO 4E98H
0294 4F00=      ISPVAL      EQU 4F00H          ;INITIAL SP VALUE
0294
0294 4F00=      INSTBUF     EQU 4F00H          ;BUFFER FOR ROM1 XFER CODE, ETC. 0200H
0294
0294 50C0=      MSGBUFF     EQU INSTBUF+01C0H ;MAX=MGT MSG
0294
0294 5080=      FILBUFF     EQU 5080H
0294 5100=      ALLOCT      EQU 5100H          ;MUST START AT PAGE EDGE. 32 BYTES - 1 PER PAGE
0294                                           ;PLUS TERMINATOR
0294 5121=      MEMVAL      EQU 5121H
0294 513F=      TLBYTE      EQU 513FH
0294 5140=      NMBUFF      EQU 5140H
0294 5140=      FIRLET      EQU NMBUFF
0294
0294 5188=      NMISTK      EQU 5188H
0294 5188=      SCRNBUF     EQU 5188H          ;8 BYTES USED BY SCREEN$ FOR COMP. FORM
0294 5190=      CHARSVAL    EQU 5190H
0294 55D8=      PALTAB      EQU 55D8H
0294 5600=      LINICOLS    EQU 5600H
0294 5800=      DKBU        EQU 5800H
0294 58E0=      KTAB        EQU 58E0H
0294
0294 FEB0=      PVBUFF      EQU 0FEB0H         ;IN SECOND SCREEN PAGE. HOLDS PRINT VARS OF NON-
0294                                           ;DISPLAYED SCREEN. BGFLG-SPOSNL AT FEB0-FEEB,
0294                                           ;CEXTAB/EXTAB AT FF7C-FFBB. FEEC-FF7B NOT USED.
0294                                           ;FF7C-FFD7 NOT USED.
0294 FEB0=      FILLSTK     EQU PVBUFF
0294 FFD8=      PALBUF      EQU 0FFD8H         ;PALETTE OF NON-DISPLAYED SCREEN (ADDRESSED AT
0294                                           ;BFD8H FOR CONVENIENCE)
0294
0294 00FF=      SNDPORT     EQU 0FFH
0294 00FE=      KEYPORT     EQU 0FEH
0294 00FD=      MDIPORT     EQU 0FDH
0294 00FC=      VIDPORT     EQU 0FCH
0294 00FB=      URPORT      EQU 0FBH
0294 00FA=      LRPORT      EQU 0FAH
0294 00F9=      STATPORT    EQU 0F9H
0294 00F8=      CLUTPORT    EQU 0F8H
0294
0294            ;PIADVAL    EQU 80H            ;mode 1 screen at 8000H
0294            ;ATADVAL    EQU 98H            ;mode 1 attributes
0294
0294
0294
0294
0294                        INCLUDE EDITOR.SAM
0294
0294            ;EDITOR.SAM
0294
0294 CDBD05     EDER:       CALL KSCHK         ;Z IF K/S CHANNEL
0297 C26704                 JP NZ,ERRCHK
029A
029A CD8804                 CALL WARNBZ        ;BUZZ AND RE-PRESENT LINE IF K OR S CHANNEL
029D 181A                   JR EDAG
029F
029F AF         EDCX:       XOR A
02A0 32A45A                 LD (XPTR+1),A
02A3 323A5C                 LD (ERRNR),A       ;NEEDED BY "INPUT LINE"
02A6
02A6 2AEC5A     EDITOR:     LD HL,(EDITV)
02A9 7C                     LD A,H
02AA B5                     OR L
02AB C40500                 CALL NZ,HLJUMP
02AE
02AE 2A3D5C                 LD HL,(ERRSP)
02B1 E5                     PUSH HL
02B2 CD893D                 CALL POFETCH
02B5 ED53825C               LD (OLDPOS),DE
02B9
02B9 219402     EDAG:       LD HL,EDER
02BC E5                     PUSH HL
02BD ED733D5C               LD (ERRSP),SP
02C1 CD7C05                 CALL AULN          ;AUTOMATIC LINE NUMBER ENTERED INTO LINE IF WANTED
02C4
02C4 21C402     EDLP:       LD HL,EDLP
02C7 E5                     PUSH HL
02C8 CD3103                 CALL EDFK          ;GET A KEY, OR ENTER USER-DEFINED KEY TEXT
02CB F5                     PUSH AF
02CC CD9904                 CALL NOISE
02CF F1                     POP AF
02D0 FE16                   CP 16H
02D2 3046                   JR NC,ADCH1        ;ENTER CHARS >=16H
02D4
02D4 FE07                   CP 7
02D6 3842                   JR C,ADCH1         ;ENTER CHARS 00-06H
```

```
02D8
02D8 FE10                  CP 10H
02DA 300E                  JR NC,TWOKYS      ;JR IF 10H-15H (INK-OVER)
02DC
02DC             ;DEAL WITH EDITING KEYS 07-0FH
02DC
02DC 216903                LD HL,EKPT-7      ;ED KEY PTR TABLE
02DF 4F                    LD C,A
02E0 5F                    LD E,A
02E1 1600                  LD D,0
02E3 19                    ADD HL,DE
02E4 5E                    LD E,(HL)
02E5 19                    ADD HL,DE
02E6 E5                    PUSH HL
02E7 C32F1F                JP ADDRKC
02EA
02EA CD1A03      TWOKYS:   CALL ADCH1        ;INSERT CONTROL CODE
02ED CDF004                CALL WAITKEY      ;GET PARAM IN A
02F0 1828                  JR ADCH1          ;JR TO INSERT PARAM
02F2
02F2             ;CHANNEL "R" - INSERT CHARS AT KCUR, EXPANDING KEYWORDS. USED BY EDKY
02F2
02F2 FE85        ADDCHAR:  CP 85H
02F4 3019                  JR NC,ADCH07
02F6
02F6 213B5C                LD HL,FLAGS       ;INSERT 00-85H (INCLUDES CONTROL CODES)
02F9 FE20                  CP 20H
02FB 280E                  JR Z,ADCH05
02FD
02FD CB86                  RES 0,(HL)        ;"LEADING SPACE NEEDED"
02FF FE3A                  CP ":"
0301 2017                  JR NZ,ADCH1
0303
0303 3AB95A                LD A,(LSTFT)      ;TEMP LIST FLG
0306 A7                    AND A
0307 3E3A                  LD A,":"
0309 280F                  JR Z,ADCH1
030B
030B CBC6        ADCH05:   SET 0,(HL)        ;NO LEADING SPACE NEEDED IF LAST CHAR=SPACE, OR
030D                                         ;LAST CHAR =":" AND PRETTY LISTING ON.
030D 180B                  JR ADCH1
030F
030F 4F          ADCH07:   LD C,A
0310 3ABA5A                LD A,(INQUFG)
0313 0F                    RRCA
0314 79                    LD A,C
0315 3803                  JR C,ADCH1       ;INSERT CHARS 85H+ IF TOKENS NOT TO BE EXPANDED
0317
0317 F7                    RST 30H
0318 BE5C                  DW PRGR802-8000H  ;PRINT KEYWORD (85-FE) OR FN (FFXX)
031A
031A             ;ENTRY HERE ALLOWS ANYTHING TO BE INSERTED
031A
031A 47          ADCH1:    LD B,A
031B CD493C                CALL R1OSR        ;ROM1 OFF, URPORT SAVED
031E C5                    PUSH BC           ;CHAR
031F CD2F1F                CALL ADDRKC       ;ADDR KCUR
0322 010100                LD BC,1
0325 CD111E                CALL MKRMCH       ;MAKE ROOM FOR ONE CHAR. ALLOW ALL SPACE TO BE USED
0328 F1                    POP AF
0329 77                    LD (HL),A
032A 23                    INC HL
032B 229A5A                LD (KCUR),HL      ;KEEP KCURP UNCHANGED SO SAME BASE AS ELINE OR WS
032E C3423C                JP POPOUT         ;RESTORE PREV LR AND  URPORT STATUS
0331
0331
0331             ;GET A KEY, AND IF IT IS A DEF KEY, ENTER ASSOCIATED TEXT INTO LINE
0331
0331 CDF004      EDFK:     CALL WAITKEY
0334 FEC0                  CP 192
0336 D8                    RET C             ;RET IF NOT IN USER-DEFINED KEY RANGE
0337
0337 FECA                  CP 202
0339 300B                  JR NC,EDFK1       ;JR IF NOT A KEYPAD KEY
033B
033B 4F                    LD C,A
033C 3A675C                LD A,(KPFLG)
033F 1F                    RRA
0340 79                    LD A,C
0341 3003                  JR NC,EDFK1       ;JR IF NUMBER FLAG NOT SET
0343
0343 D690                  SUB 144           ;192-201 -> "0"-"9"
0345 C9                    RET
0346
0346 CDAA05      EDFK1:    CALL FNDKYD       ;LOOK FOR DEFINITION
0349 7A                    LD A,D            ;A=KEY CODE AGAIN
034A D8                    RET C             ;RET IF NONE
034B
034B E5                    PUSH HL           ;DEF. START
034C 09                    ADD HL,BC
034D 2B                    DEC HL            ;HL=DEF END
034E 7E                    LD A,(HL)
034F FE3A                  CP ":"
0351 2001                  JR NZ,EDFK2
0353
0353 0B                    DEC BC            ;LOP OFF LAST CHAR IF IT IS ":"
0354
0354 F5          EDFK2:    PUSH AF           ;SAVE LAST CHAR
0355 C5                    PUSH BC           ;LEN
0356 CD2F1F                CALL ADDRKC
0359 CD1B1E                CALL MAKEROOM     ;OPEN ROOM FOR DEF KEY TEXT
035C EB                    EX DE,HL          ;DE=ROOM
035D C1                    POP BC            ;LEN
```

```
035E F1                     POP AF
035F E1                     POP HL              ;SRC
0360 F5                     PUSH AF
0361 EDB0                   LDIR
0363 EB                     EX DE,HL
0364 229A5A                 LD (KCUR),HL        ;CURSOR POSN AFTER DEF KEY TEXT
0367 CD9904                 CALL NOISE
036A F1                     POP AF
036B D1                     POP DE              ;RET ADDR
036C C26504                 JP NZ,EDENT         ;ENTER LINE IF LAST CHAR WASN"T ":"
036F
036F C9                     RET                 ;BACK TO EDLP
0370
0370            ;ED KEY PTR TABLE
0370
0370 09         EKPT:       DB EDKY-EKPT        ;7  EDIT
0371 68                     DB EDLT-EKPT-1      ;8  LEFT
0372 79                     DB EDRT-EKPT-2      ;9  RIGHT
0373 84                     DB EDDN-EKPT-3      ;10 DOWN
0374 83                     DB EDUP-EKPT-4      ;11 UP
0375 D0                     DB EDDLL-EKPT-5     ;0C DEL LEFT
0376 EF                     DB EDENT-EKPT-6     ;0D ENTER
0377 C8                     DB EDDLR-EKPT-7     ;0E DEL RIGHT
0378 C2                     DB EDKPX-EKPT-8     ;0F KEY PAD TOGGLE
0379
0379            ;"PRINT" LINE TO ELINE EDITING BUFFER
0379
0379 3A715C     EDKY:       LD A,(FLAGX)
037C E620                   AND 20H
037E C2A504                 JP NZ,CLEARSP       ;JP IF INPUT - CLEAR INPUT LINE
0381
0381 CD7910                 CALL EVALLINO       ;SKIP ELINE NUMBER, GET IN BC. CY IF TOO BIG
0384 380D                   JR C,EDKY2
0386
0386 DF                     RST 18H
0387 FE0D                   CP 0DH
0389 2008                   JR NZ,EDKY2         ;JR IF LINE NEITHER EMPTY NOR JUST LINE NO.
038B
038B            ;FROM EDIT (CMD)
038B
038B 78         EDKY1:      LD A,B
038C B1                     OR C
038D 2804                   JR Z,EDKY2
038F
038F ED43495C               LD (EPPC),BC        ;E.G. ENTER 123 (EDIT) SETS EPPC TO 123
0393
0393 CDA504     EDKY2:      CALL CLEARSP        ;CLEAR ELINE OR INPUT LINE
0396 2A495C                 LD HL,(EPPC)
0399 CD4D1A                 CALL FNDLINE
039C CDD604                 CALL LNNM           ;GET LINE NUMBER OR ZERO
039F 7A                     LD A,D
03A0 B3                     OR E
03A1 C8                     RET Z               ;DON"T EDIT LINE ZERO
03A2
03A2 ED5B515C               LD DE,(CURCHL)
03A6 D5                     PUSH DE
03A7 3A4A5C                 LD A,(EPPC+1)
03AA F5                     PUSH AF
03AB E5                     PUSH HL             ;LSB OF LINE NO
03AC 3EFF                   LD A,0FFH
03AE 324A5C                 LD (EPPC+1),A       ;ENSURE NO ">" IS PRINTED
03B1 CD6311                 CALL SETSTRM        ;"R" O/P
03B4 21B85A                 LD HL,LISTFLG
03B7 7E                     LD A,(HL)
03B8 32B95A                 LD (LSTFT),A        ;PUT PRTY LISTING STATUS WHERE CHAN "R" CAN SEE IT.
03BB 3600                   LD (HL),0           ;PRETTY LISTING OFF - COLONS STAY COLONS!
03BD E3                     EX (SP),HL
03BE 2B                     DEC HL              ;PT TO LINE START
03BF F7                     RST 30H
03C0 14F3                   DW OUTLINE
03C2 E1                     POP HL
03C3 3AB95A                 LD A,(LSTFT)
03C6 77                     LD (HL),A           ;PREV LIST FLAG RESTORED.
03C7 F1                     POP AF
03C8 324A5C                 LD (EPPC+1),A
03CB 2A945A                 LD HL,(ELINE)
03CE 010500                 LD BC,5
03D1 09                     ADD HL,BC
03D2 229A5A                 LD (KCUR),HL        ;CURSOR ADDR IS AFTER 5-DIGIT LINE NUMBER
03D5 E1                     POP HL
03D6 C37011                 JP CHANFLAG
03D9
03D9            ;CURSOR LEFT (ALSO CALLED BY DELETE LEFT)
03D9
03D9 CDC204     EDLT:       CALL SETDE          ;GET DE=START OF LINE (HL=CURSOR ADDR). NC
03DC
03DC 2B         EDLT2:      DEC HL              ;KCUR MOVES LEFT
03DD ED52                   SBC HL,DE
03DF 19                     ADD HL,DE
03E0 D8                     RET C               ;RET IF AT LINE START ALREADY (NEW POSN WOULD BE
03E1                                            ;FF SAVARS TERMINATOR)
03E1 2810                   JR Z,EDRLC          ;JR IF NEW POSN WOULD BE START OF LINE
03E3
03E3 2B                     DEC HL              ;ELSE LOOK AT PREV. CHAR (*NOT* SAVARS TERM)
03E4 7E                     LD A,(HL)
03E5 23                     INC HL
03E6 3C                     INC A
03E7 28F3                   JR Z,EDLT2          ;MOVE AGAIN IF CURSOR WOULD POINT TO FN CODE AFTER
03E9                                            ;FF PREFIX
03E9
03E9 1808                   JR EDRLC            ;CHANGE KCUR UNLESS ALREADY AT LINE START
03EB
03EB 7E         EDRT:       LD A,(HL)
```

```
03EC 23                 INC HL
03ED FE0D               CP 0DH
03EF C8                 RET Z               ;NO MOVE RIGHT IF AT LINE END
03F0
03F0 3C                 INC A
03F1 28F8               JR Z,EDRT           ;MOVE AGAIN IF HIT FFH FN PREFIX
03F3
03F3 229A5A     EDRLC:  LD (KCUR),HL
03F6 C9                 RET
03F7
03F7            EDDN:                       ;A=0A
03F7 79         EDUP:   LD A,C              ;KEY 0B/0A
03F8 21715C             LD HL,FLAGX
03FB CB6E               BIT 5,(HL)
03FD 200A               JR NZ,EDUD2         ;JR IF INPUT MODE
03FF
03FF CD351F             CALL ADDRELN
0402 7E                 LD A,(HL)
0403 FE0D               CP 0DH
0405 79                 LD A,C
0406 CA4308             JP Z,FUPDN          ;JR IF EDIT MODE, ELINE EMPTY - MOVE ">" CURSOR
0409
0409 21C35A     EDUD2:  LD HL,KPOS+1        ;CURSOR LINE
040C FE0B               CP 0BH
040E 2802               JR Z,EDUD25
0410
0410 34                 INC (HL)
0411 34                 INC (HL)
0412
0412 35         EDUD25: DEC (HL)            ;CURSOR LINE ADJUSTED UP
0413 118504             LD DE,CUOP
0416 CD3504             CALL KOPSET         ;ALTER CHANNEL K TO SPECIAL O/P
0419 2A915A             LD HL,(WORKSP)
041C 2B                 DEC HL
041D 3A715C             LD A,(FLAGX)
0420 E620               AND 20H
0422 2803               JR Z,EDUD3          ;JR WITH HL=END OF ELINE IF EDIT MODE
0424
0424 2A8E5A             LD HL,(WKEND)       ;ELSE GET END OF INPUT LINE
0427
0427 2B         EDUD3:  DEC HL
0428 229A5A             LD (KCUR),HL        ;SET KCUR TO END OF LINE IN CASE NO MATCH OCCURS.
042B CD9904             CALL NOISE
042E CD7605             CALL EDPRT          ;PRINT LINE, SETTING KCUR TO MATCH WITH KPOS
0431 ED5BDC5B           LD DE,(MNOP)        ;FBA7 EQU
0435
0435 2A4F5C     KOPSET: LD HL,(CHANS)
0438 183F               JR DETOHL
043A
043A            ;TOGGLE KEYPAD BIT (FUNCTION/NUMBER PAD)
043A
043A 21675C     EDKPX:  LD HL,KPFLG
043D 34                 INC (HL)
043E C9                 RET
043F
043F            ;EDITOR DELETE RIGHT (KEY 0E)
043F
043F 7E         EDDLR:  LD A,(HL)
0440 FE0D               CP 0DH
0442 2005               JR NZ,EDDLC
0444
0444 C9                 RET                 ;RET IF HIT LINE END
0445
0445            ;EDITOR DELETE LEFT
0445
0445 CDD903     EDDLL:  CALL EDLT
0448 D8                 RET C               ;RET IF AT START OF LINE
0449
0449 010200     EDDLC:  LD BC,2             ;2 BYTES TO DELETE
044C 7E                 LD A,(HL)
044D 3C                 INC A
044E 2812               JR Z,EDDL3          ;IF FN, DELETE FN PREFIX AND CODE
0450
0450            ;CARRIAGE RETURN USED BY LOCAL!
0450
0450 0D         CARET:  DEC C               ;ELSE DEL 1 BYTE
0451 2B                 DEC HL
0452 7E                 LD A,(HL)           ;CHAR BEFORE CHAR TO DEL. (PERHAPS FF SAVARS TERM.)
0453 23                 INC HL
0454 FE16               CP 16H
0456 300A               JR NC,EDDL3         ;JR IF >OVER
0458
0458 FE10               CP 10H
045A 3806               JR C,EDDL3          ;JR IF NOT INK-OVER
045C
045C 23                 INC HL
045D 229A5A             LD (KCUR),HL        ;PT KCUR PAST PARAM, SO IT WILL BE DELED NEXT TIME
0460 2B                 DEC HL
0461 2B                 DEC HL              ;DELETE CONTROL CODE NOW, NOT PARAM
0462
0462 C3521E     EDDL3:  JP RECLAIM2         ;RECLAIM AT (HL)
0465
0465
0465 F1         EDENT:  POP AF              ;JUNK EDLP
0466 F1                 POP AF              ;JUNK WARNBZ
0467
0467 E1         ERRCHK: POP HL
0468
0468 223D5C     RESESP: LD (ERRSP),HL
046B 3A3A5C             LD A,(ERRNR)
046E A7                 AND A
046F C8                 RET Z               ;RET IF NO ERRORS
0470
```

```
0470 F9                     LD SP,HL
0471 C9                     RET                 ;ELSE RET TO ERROR HANDLER
0472
0472              ;CONTROL CODE PARAM HANDLING ROUTINES
0472
0472 ED5BB55A     RESTOP:   LD DE,(OPSTORE)
0476
0476 2A515C       POCHNG:   LD HL,(CURCHL)
0479
0479 73           DETOHL:   LD (HL),E
047A 23                     INC HL
047B 72                     LD (HL),D
047C C9                     RET
047D
047D 11A73D       PRERESTOP: LD DE,CCRESTOP
0480 32BF5B                 LD (TVDATA+1),A
0483 18F1                   JR POCHNG
0485
0485              ;SPECIAL PRINT O/P ROUTINE TO CHECK IF LS POSN MATCHES DESIRED CURSOR POSN AS
0485              ;EACH CHAR IS PRINTED. IF SO, KCUR IS SET TO ADDR OF THE CHAR IN THE LINE.
0485
0485 F7           CUOP:     RST 30H
0486 1F75                   DW CUOPP-8000H    ;JP ROM1
0488
0488              ;WARNING BUZZ - EDITOR ERRORS
0488
0488 3A735A       WARNBZ:   LD A,(DEVICE)
048B 3D                     DEC A
048C 20D9                   JR NZ,ERRCHK      ;REPORT ERROR IF NOT LOWER SCREEN DEVICE
048E
048E              ;CALLED BY LINE ENTRY ERRORS
048E
048E 2607         RSPNS:    LD H,7
0490 AF                     XOR A
0491 323A5C                 LD (ERRNR),A      ;"NO ERROR"
0494 3A385C                 LD A,(RASP)
0497 1806                   JR NS2
0499
0499              ;PIP NOISE
0499
0499 21FA00       NOISE:    LD HL,250
049C 3A395C                 LD A,(PIP)
049F
049F 5F           NS2:      LD E,A
04A0 1600                   LD D,0
04A2
04A2 F7           BEEPER:   RST 30H
04A3 1B6F                   DW BEEPP2-8000H
04A5
04A5              ;RECLAIM ELINE-WORKSP-1 IF EDITING, OR WORKSP-WKEND-1 IF EDITING
04A5
04A5 CDC204       CLEARSP:  CALL SETDE
04A8 2805                   JR Z,CLRSP2       ;JR IF EDIT MODE
04AA
04AA 2A8E5A                 LD HL,(WKEND)
04AD 1804                   JR CLRSP3
04AF
04AF 2A915A       CLRSP2:   LD HL,(WORKSP)
04B2 2B                     DEC HL            ;PT TO 0D IN ELINE (NOT DELETED)
04B3
04B3 2B           CLRSP3:   DEC HL
04B4 CD4F1E                 CALL RECLAIM1     ;CLEAR ELINE
04B7
04B7 DBFB         SETKC:    IN A,(URPORT)
04B9
04B9 E61F         SETKC2:   AND 1FH
04BB 32995A                 LD (KCURP),A
04BE 229A5A                 LD (KCUR),HL
04C1 C9                     RET
04C2
04C2              ;GET DE=START OF ELINE IF IN EDIT MODE (Z), ELSE DE PTS TO INPUT LINE IN WKSPACE
04C2
04C2 E5           SETDE:    PUSH HL
04C3 3A715C                 LD A,(FLAGX)
04C6 E620                   AND 20H
04C8 2004                   JR NZ,SETDE2
04CA
04CA CD351F                 CALL ADDRELN
04CD BF                     CP A
04CE
04CE F5           SETDE2:   PUSH AF
04CF C42B1F                 CALL NZ,ADDRWK
04D2 EB                     EX DE,HL
04D3
04D3              ;USED BY DEF KEYCODE
04D3
04D3 F1           PPRET:    POP AF
04D4 E1                     POP HL
04D5 C9                     RET
04D6
04D6              ;ENTRY: USUALLY, HL PTS TO LINE NUMBER, DE TO LINE NO OF PREVIOUS LINE, AND
04D6              ;EXITS WITH DE=LINE NUMBER. HOWEVER, IF HL HIT PROG TERMINATOR, PREVIOUS LINE NO
04D6              ;IS RETURNED, AND IF NO LINES EXIST, DE=0
04D6
04D6 7E           LNNM:     LD A,(HL)
04D7 3C                     INC A
04D8 2006                   JR NZ,LNNM2       ;JR IF NOT PROG TERMINATOR
04DA
04DA EB                     EX DE,HL          ;LOOK AT PREV LINE.
04DB 7E                     LD A,(HL)
04DC 3C                     INC A
04DD 57                     LD D,A
04DE 5F                     LD E,A
```

```
04DF C8                     RET Z             ;DE=0 IF NULL PROGRAM
04E0
04E0 56          LNNM2:     LD D,(HL)
04E1 23                     INC HL
04E2 5E                     LD E,(HL)
04E3 C9                     RET
04E4
04E4             ;USED BY SCROLL/SAVE TO TAPE
04E4
04E4 CDBD1C      GTKBK:     CALL KBFLUSH
04E7
04E7             ;USED BY GET AND CONTROL KEY ENTRY ROUTINE
04E7
04E7 CD5D0E      WKBR:      CALL BRKCR        ;CHECK BREAK, STOP IF SO
04EA CD0A05                 CALL KYIP2
04ED 28F8                   JR Z,WKBR
04EF
04EF C9                     RET
04F0
04F0 213C5C      WAITKEY:   LD HL,TVFLAG
04F3 7E                     LD A,(HL)
04F4 E620                   AND 20H
04F6 2002                   JR NZ,WTKY2       ;JR IF THE LOWER SCREEN IS GOING TO BE CLEARED
04F8
04F8 CBDE                   SET 3,(HL)        ;"LINE TO BE PRINTED TO LS" ON 1ST CALL OF INPUTAD
04FA
04FA CDBA01      WTKY2:     CALL INPUTAD      ;CALL I/P OF CURRENT CHANNEL - OFTEN KYIP BELOW
04FD D8                     RET C             ;RET IF GOT A KEY IN A
04FE 28FA                   JR Z,WTKY2        ;JR IF NO KEY, AND NO ERRORS
0500
0500 CF                     RST 08H
0501 16                     DB 22             ;"END OF FILE"
0502
0502
0502             ;KEYBOARD INPUT - KYIP ADDR IS IN CHANNELS THAT ALLOW KB INPUT
0502             ;EXIT: CY IF GOT KEY IN A. NC,Z IF NO KEY. NC,NZ IF END OF FILE.
0502
0502 3A3C5C      KYIP:      LD A,(TVFLAG)
0505 E608                   AND 08H
0507 C47605                 CALL NZ,EDPRT     ;PRINT LINE TO LS IF REQUIRED
050A
050A             ;USED BY GET
050A
050A 213B5C      KYIP2:     LD HL,FLAGS
050D A7                     AND A             ;NC
050E CB6E                   BIT 5,(HL)
0510 C8                     RET Z             ;RET IF NO KEY PRESSED
0511
0511 3A085C                 LD A,(LASTK)
0514 CBAE                   RES 5,(HL)
0516 F5                     PUSH AF
0517 23                     INC HL
0518 CB6E                   BIT 5,(HL)        ;TVFLAG
051A C4B506                 CALL NZ,CLSLOWER
051D F1                     POP AF
051E FE16                   CP 16H
0520 3F                     CCF
0521 D8                     RET C             ;ACCEPT ANYTHING >=16H
0522
0522 FE06                   CP 6
0524 2842                   JR Z,KYCL         ;JR IF CAPS LOCK
0526
0526 FE10                   CP 10H
0528 D8                     RET C             ;ACCEPT 00-0FH
0529
0529 32025C                 LD (KDATA),A      ;SAVE CONTROL CODE 10H-15H (INK-OVER)
052C 113105                 LD DE,KYPM
052F 182E                   JR KYCZ           ;RETURN WITH CONTROL CODE, AND ALTER I/P SO NEXT
0531                                          ;KEY IS ADJUSTED AND CHECKED FOR RANGE VS CC.
0531 213B5C      KYPM:      LD HL,FLAGS
0534 7E                     LD A,(HL)
0535 E620                   AND 20H
0537 C8                     RET Z             ;RET (NC, Z) IF NO KEY
0538
0538 3A085C                 LD A,(LASTK)
053B CBAE                   RES 5,(HL)        ;"NO KEY"
053D D630                   SUB 30H
053F 3833                   JR C,KYPN         ;ONLY ACCEPT 30H+
0541
0541 FE08                   CP 8
0543 302F                   JR NC,KYPN        ;LIMIT TO 0-7
0545
0545 47                     LD B,A
0546 3A025C                 LD A,(KDATA)      ;10-15H
0549 FE12                   CP 12H
054B 380D                   JR C,KYPM6        ;JR IF INK OR PAPER; 0-7 OK
054D
054D FE15                   CP 15H
054F 78                     LD A,B            ;PARAM
0550 2804                   JR Z,KYPM5        ;JR IF OVER
0552
0552 FE02                   CP 2
0554 301E                   JR NC,KYPN        ;0/1 ONLY FOR FLASH/BRIGHT/INVERSE
0556
0556 FE04        KYPM5:     CP 4
0558 301A                   JR NC,KYPN        ;0-3 ONLY FOR OVER
055A
055A 78          KYPM6:     LD A,B
055B ED5BDE5B               LD DE,(MNIP)      ;RESTORE NORMAL I/P
055F
055F 37          KYCZ:      SCF
0560 2A4F5C                 LD HL,(CHANS)     ;KEY I/P CHANNEL ZAP
0563 23                     INC HL
```

```
0564 23                    INC HL
0565 C37904                JP DETOHL
0568
0568 216A5C      KYCL:     LD HL,FLAGS2
056B 7E                    LD A,(HL)
056C EE08                  XOR 8             ;REVERSE CAPS LOCK
056E 77                    LD (HL),A
056F
056F 213C5C                LD HL,TVFLAG
0572 CBDE                  SET 3,(HL)        ;"COPY LINE TO SCREEN"
0574
0574 BF          KYPN:     CP A              ;NC,Z - NO KEY
0575 C9                    RET
0576
0576             ;PRINT EDIT/INPUT LINE TO LOWER SCREEN
0576
0576 F7          EDPRT:    RST 30H
0577 9474                  DW EDPTR2-8000H
0579
0579             ;FORCE NORMAL OUTPUT. USED BY E.G CURSOR OUTPUT TO ENSURE NORMAL OUTPUT EVEN
0579             ;IF PRINTING BETWEEN CONTROL CODE AND PARAMETER.
0579
0579 F7          FONOP:    RST 30H
057A 1360                  DW FONOP2-8000H
057C
057C             ;AUTOMATIC LINE NUMBER ENTRY
057C
057C 3A715C      AULN:     LD A,(FLAGX)
057F E620                  AND 20H
0581 C0                    RET NZ            ;RET IF INPUT MODE
0582
0582 3A885B                LD A,(AUTOFLG)
0585 A7                    AND A
0586 C8                    RET Z             ;RET IF AUTO OFF
0587
0587 CD351F                CALL ADDRELN
058A 7E                    LD A,(HL)
058B FE0D                  CP 0DH
058D C0                    RET NZ            ;RET IF LINE NOT EMPTY
058E
058E 2A495C                LD HL,(EPPC)
0591 ED4B895B              LD BC,(AUTOSTEP)
0595 09                    ADD HL,BC
0596 7C                    LD A,H
0597 FEFF                  CP 0FFH
0599 C8                    RET Z             ;RET IF LINE NUMBER TOO BIG
059A
059A E5                    PUSH HL
059B 3EFF                  LD A,0FFH
059D CD6311                CALL SETSTRM      ;CHANNEL "R"
05A0 C1                    POP BC
05A1 F7                    RST 30H
05A2 ABF5                  DW PRNUMB1
05A4
05A4 AF          STRM0:    XOR A
05A5 C36311                JP SETSTRM
05A8
05A8             ;FNDKYD - FIND DEF KEY DATA
05A8             ;ENTRY: A=KEY CODE (192-255)
05A8             ;EXIT: HL PTS TO START OF DEFINITION, BC=LEN, D=KEY CODE. CY IF NOT FOUND
05A8             ;USES AF,BC,D,HL
05A8             ;DEFINITIONS TERMINATED BY FFH
05A8
05A8             ;ENTRY AT DKTR FINDS POSN OF TERMINATOR,+3
05A8
05A8 3EFF        DKTR:     LD A,0FFH
05AA
05AA 57          FNDKYD:   LD D,A
05AB 2ACE5B                LD HL,(DKDEF)
05AE
05AE 7E          FDKL:     LD A,(HL)         ;CODE
05AF 23                    INC HL
05B0 4E                    LD C,(HL)
05B1 23                    INC HL
05B2 46                    LD B,(HL)         ;BC=LEN
05B3 23                    INC HL
05B4 C601                  ADD A,1
05B6 D8                    RET C             ;RET IF NOT FOUND - (TERMINATOR)
05B7
05B7 3D                    DEC A
05B8 BA                    CP D
05B9 C8                    RET Z             ;RET IF FOUND
05BA
05BA 09                    ADD HL,BC         ;PT TO NEXT DEFINITION
05BB 18F1                  JR FDKL
05BD
05BD
05BD             ;Z IF K OR S CHANNEL. USED BY EDITOR AND INPUT
05BD
05BD 3A745A      KSCHK:    LD A,(CLET)
05C0 FE4B                  CP "K"
05C2 C8                    RET Z
05C3
05C3 FE53                  CP "S"
05C5 C9                    RET
05C6
05C6                       INCLUDE LIST.SAM       ;AUTOLIST, LIST, SPACAN,
05C6             ;LIST.SAM   LIST, CLS
05C6
05C6 CDB506      AUTOLIST: CALL CLSLOWER
05C9 21495C                LD HL,EPPC
05CC CD3709                CALL REALN        ;MAKE SURE EPPC HAS THE NUMBER OF A REAL LINE
05CF
```

```
05CF             ;USED BY AUTO
05CF
05CF 216C5C      AUL2:      LD HL,SDTOP
05D2 CD3709                 CALL REALN          ;DITTO FOR SDTOP
05D5
05D5 CDF006                 CALL CLSUP          ;"S"
05D8 3E10                   LD A,10H
05DA 323C5C                 LD (TVFLAG),A       ;"AUTOLIST, UPPER SCREEN"
05DD 216A5C                 LD HL,FLAGS2
05E0 CBC6                   SET 0,(HL)          ;"SCREEN IS NOT CLEAR"
05E2 2A6C5C                 LD HL,(SDTOP)
05E5 ED5B495C               LD DE,(EPPC)
05E9 ED537F5A               LD (LAST),DE        ;LAST LINE THAT *MUST* BE LISTED IS EPPC
05ED A7                     AND A
05EE ED52                   SBC HL,DE
05F0 EB                     EX DE,HL            ;HL=EPPC
05F1 302F                   JR NC,AUL4          ;JR IF EPPC IS ABOVE OR EQU TO TOP LINE IN
05F3                                            ;AN AUTOLIST - ALTER SDTOP
05F3 CD4D1A                 CALL FNDLINE        ;ELSE ENSURE THAT SDTOP IS NOT TOO FAR ABOVE EPPC
05F6 DBFB                   IN A,(URPORT)
05F8 F5                     PUSH AF
05F9 E5                     PUSH HL             ;HL SHOULD BE ABOUT 0100 HIGHER THAN ADDR FOR SDTOP
05FA 2A6C5C                 LD HL,(SDTOP)       ;SO THAT IT APPEARS WITHOUT SCROLLING
05FD CD4D1A                 CALL FNDLINE
0600 D1                     POP DE              ;EPPC ADDR
0601 0D                     DEC C
0602 15                     DEC D               ;ADDR IS 512 BEFORE EPPC IN LISTING
0603 F1                     POP AF              ;EPPC PAGE
0604
0604 0EFB        AULLP:     LD C,URPORT
0606 ED40                   IN B,(C)            ;B=SDTOP PAGE
0608 B8                     CP B
0609 2005                   JR NZ,AUL25         ;JR IF SDTOP AND EPPC PAGES DON"T MATCH
060B
060B ED52                   SBC HL,DE           ;SBC SDTOP ADDR, EPPC-0200H
060D 19                     ADD HL,DE           ;HL=SDTOP ADDR AGAIN
060E 300E                   JR NC,AUL3          ;JR IF DIFF <100H
0610
0610 23          AUL25:     INC HL
0611 23                     INC HL
0612 4E                     LD C,(HL)
0613 23                     INC HL
0614 46                     LD B,(HL)
0615 09                     ADD HL,BC
0616 F5                     PUSH AF
0617 CDEF3F                 CALL CHKHL
061A F1                     POP AF
061B 23                     INC HL              ;NEXT POSSIBLE SDTOP ADDR TO CONSIDER
061C 18E6                   JR AULLP
061E
061E 56          AUL3:      LD D,(HL)
061F 23                     INC HL
0620 5E                     LD E,(HL)
0621 EB                     EX DE,HL            ;NEW SDTOP LINE NO.
0622
0622 226C5C      AUL4:      LD (SDTOP),HL
0625 ED733F5C               LD (LISTSP),SP      ;SO "SCROLL" CAN ABORT AT END OF SCREEN.
0629 CD8406                 CALL LIST5          ;LIST LINES FROM LINE HL, INDENTED
062C
062C 213C5C      AULX:      LD HL,TVFLAG        ;ALWAYS COME BACK HERE, EVEN IF "SCROLL" ABORTED
062F CBA6                   RES 4,(HL)
0631 01CD08                 LD BC,IOPOF
0634 C31F02                 JP R1XJP            ;ENSURE INDENTED OP IS OFF, TURN ROM1 OFF
0637
0637 0E03        LLIST:     LD C,03H
0639 21                     DB 21H              ;"JR+2"
063A
063A 0E02        LIST:      LD C,02H
063C AF                     XOR A
063D 323C5C                 LD (TVFLAG),A       ;NON-AUTOLIST (BIT 4=0)
0640 CDC63A                 CALL RUNFLG
0643 79                     LD A,C
0644 DC6311                 CALL C,SETSTRM
0647 DF                     RST 18H
0648 FE91                   CP 91H              ;FORMATTOK
064A 200D                   JR NZ,LIST1
064C
064C CD573A                 CALL SSYNTAX6       ;SKIP, GET NUMBER
064F
064F 111E03                 LD DE,0300H+30
0652 CD621F                 CALL LIMBYTE
0655 32B85A                 LD (LISTFLG),A      ;0 IF LIST FORMAT 0 - NO INDENT
0658 C9                     RET
0659
0659 CD3211      LIST1:     CALL PRHSH1         ;POSSIBLE "#"
065C DF                     RST 18H
065D CD6C3A                 CALL COMMASC        ;COMMA OR SEMICOLON
0660 2001                   JR NZ,LIST2
0662
0662 E7                     RST 20H             ;SKIP ,/;
0663
0663 CD4B3B      LIST2:     CALL BRKLSSL        ;CALL BRACKETLESS SLICER SR - POSSIBLE ST TO FIN
0666 213B5C                 LD HL,FLAGS         ;CY IF ERROR
0669 CB7E                   BIT 7,(HL)
066B C8                     RET Z               ;RET IF NOT RUNNING
066C
066C F5                     PUSH AF             ;CY SHOWS ERROR  !!
066D A7                     AND A
066E 2004                   JR NZ,LIST3         ;JR IF NOT 1-NUMBER SLICER
0670                                            ;ELSE TURN EG LIST 10 INTO LIST 10 TO END (BUT
0670                                            ; DELETE KEEPS DELETE 10 AS IS)
0670 3D                     DEC A               ;A=FF
0671 32805A                 LD (LAST+1),A       ;LAST LINE=HIGH NUMBER
```

```
0674
0674 2A7D5A     LIST3:      LD HL,(FIRST)     ;GET FIRST LINE
0677 7C                     LD A,H
0678 B5                     OR L
0679 2805                   JR Z,LIST4        ;IGNORE OUT OF RANGE CAUSED BY LIST 0
067B
067B F1                     POP AF            ;CY SHOWS IF OUT OF RANGE
067C DA391D                 JP C,IOORERR
067F
067F F5                     PUSH AF
0680
0680 F1         LIST4:      POP AF
0681 22495C                 LD (EPPC),HL      ;NEW EPPC=FIRST LISTED LINE
0684
0684 CD4D1A     LIST5:      CALL FNDLINE
0687 018D06                 LD BC,LSTLNS
068A C3C508                 JP IOPCL
068D
068D F7         LSTLNS:     RST 30H
068E E472                   DW LSTR1-8000H
0690
0690            ;CANCEL SPACES
0690
0690 AF         SPACAN:     XOR A
0691 32BE5A                 LD (NXTSPCS),A
0694 32C05A                 LD (NXTHSPCS),A
0697 C9                     RET
0698
0698 AF         MCLS:       XOR A
0699 1813                   JR CLSBL
069B
069B FE23       CLS:        CP "#"            ;28 BYTES FOR CLS #
069D 2003                   JR NZ,CLSNH
069F
069F F7                     RST 30H
06A0 7D6D                   DW CLSHS-8000H
06A2
06A2 CD503A     CLSNH:      CALL SYNTAX3
06A5
06A5 CD331D                 CALL GETBYTE      ;0 IF CLEAR ENTIRE SCREEN
06A8 1E06                   LD E,6            ;BLITZ CODE FOR "CLS"
06AA CD6628                 CALL GRAREC
06AD 79                     LD A,C            ;PARAM OR 0
06AE
06AE            ;BLITZ ENTRY
06AE
06AE FE01       CLSBL:      CP 1
06B0 2840                   JR Z,CLU1
06B2
06B2 CDF206                 CALL CLU1
06B5
06B5 213F5A     CLSLOWER:   LD HL,LWBOT
06B8 7E                     LD A,(HL)
06B9 2B                     DEC HL
06BA 96                     SUB (HL)          ;SUB LWTOP
06BB 3D                     DEC A
06BC 2822                   JR Z,CLSL2        ;JR IF LW IS ONLY 2 LINES HIGH
06BE
06BE                                          ;DEFINE A WINDOW COVERING LW OVERLAP ONTO UPPER
06BE                                          ;WINDOW (ABOVE NORMAL 2-LINE SIZE)
06BE 7E                     LD A,(HL)         ;LWTOP
06BF 32585A                 LD (WINDTOP),A
06C2 23                     INC HL
06C3 7E                     LD A,(HL)         ;LWBOT
06C4 3D                     DEC A
06C5 2B                     DEC HL
06C6 77                     LD (HL),A         ;LWTOP IS 1 LESS THAN LWBOT
06C7 3D                     DEC A
06C8 32595A                 LD (WINDBOT),A
06CB 2A3C5A                 LD HL,(LWRHS)
06CE 22565A                 LD (WINDRHS),HL   ;RHS AND LHS
06D1 2A485A                 LD HL,(M23PAPP)
06D4 22515A                 LD (M23PAPT),HL
06D7 3A455A                 LD A,(ATTRP)
06DA 324E5A                 LD (ATTRT),A      ;ENSURE UW COLOURS
06DD CD6D0B                 CALL CLSWIND      ;TEMP OVERLAP WINDOW CLEARED
06E0
06E0 CD6111     CLSL2:      CALL STREAMFD     ;"K"
06E3 3E01                   LD A,1
06E5 323C5C                 LD (TVFLAG),A     ;LS, DO NOT CLEAR ON KEYSTROKE, NOT AUTOLIST
06E8 CD4007                 CALL CLWC         ;CLEAR WINDOW AND RESET CHANNEL
06EB 24                     INC H
06EC 226E5A                 LD (SPOSNL),HL    ;POSN=LHS, TOP+1
06EF C9                     RET
06F0
06F0
06F0 3E01       CLSUP:      LD A,1            ;"CLEAR WINDOW, NOT ENTIRE SCREEN"
06F2
06F2 F5         CLU1:       PUSH AF
06F3
06F3 EF                     DB CALC           ;GRAPHICS COORDS 0,0
06F4 E1                     DB STKZERO
06F5 E1                     DB STKZERO
06F6 33                     DB EXIT
06F7
06F7 CD2E20                 CALL SETESP       ;ENSURE NO ERROR ON CLS EVEN IF XOS ETC ODD
06FA CD9327                 CALL GTFCOORDS    ;GET PHYSICAL COORDS FROM XOS ETC AND 0,0
06FD 3803                   JR C,CLU2         ;JR IF THINPIX - HL=X, B=Y
06FF
06FF 69                     LD L,C
0700 2600                   LD H,0            ;HL=X NOW
0702
0702 22425A     CLU2:       LD (XCOORD),HL
```

```
0705 78                  LD A,B
0706 32415A              LD (YCOORD),A
0709 E1                  POP HL
070A 223D5C              LD (ERRSP),HL
070D AF                  XOR A
070E 32A45A              LD (XPTR+1),A     ;RESET ANY "?" ERROR PTR
0711 323A5C              LD (ERRNR),A      ;"OK"
0714 3C                  INC A
0715 328C5C              LD (SCRCT),A
0718 CD5E11              CALL STREAMFE     ;"S"
071B F1                  POP AF
071C A7                  AND A
071D 2007                JR NZ,CLS2
071F
071F CD5307              CALL CLSE         ;CLEAR ENTIRE SCREEN IF CLS OR CLS 0
0722 CD4307              CALL CLWC2        ;RESET CHANNEL
0725 BF                  CP A              ;Z
0726
0726 C44007     CLS2:    CALL NZ,CLWC      ;CLEAR WINDOW, RESET CHANNEL
0729 226C5A              LD (SPOSNU),HL
072C 216A5C              LD HL,FLAGS2
072F CB86                RES 0,(HL)        ;"SCREEN IS CLEAR"
0731
0731             ;ZERO TABLE ENTRIES
0731
0731 21525B              LD HL,LPT
0734 061E                LD B,30
0736 AF                  XOR A
0737
0737 77          ZTEL:   LD (HL),A
0738 23                  INC HL
0739 10FC                DJNZ ZTEL
073B
073B 3D                  DEC A
073C 328D5B              LD (LNPTR),A      ;FF SHOWS NO CURSOR LINE
073F C9                  RET
0740
0740 CD6D0B     CLWC:    CALL CLSWIND
0743
0743 ED5B515C   CLWC2:   LD DE,(CURCHL)
0747 21DC5B              LD HL,MNOP
074A 010400              LD BC,4
074D EDB0                LDIR              ;REFRESH CHANNEL O/P AND I/P ADDRESSES IN CASE
074F                                       ;THEY HAVE BEEN ALTERED E.G. FOR COLOUR I/P
074F 2A575A              LD HL,(WINDLHS)   ;GET TOP LEFT OF WINDOW (FOR PRINT POSN)
0752 C9                  RET
0753
0753             ;CLEAR ENTIRE SCREEN - QUICKLY
0753
0753 3A405A      CLSE:   LD A,(MODE)
0756 FE02                CP 2
0758 301F                JR NC,CLS1
075A
075A 2698                LD H,98H          ;END OF MODE 0/1 PATTERN DATA
075C 1E00                LD E,0            ;CLEAR WITH ZEROS
075E 010280              LD BC,8002H       ;DO 1800H BYTES
0761 F5                  PUSH AF
0762 CD8207              CALL CLSG         ;CLEAR M0/M1 PATTERN DATA
0765 F1                  POP AF            ;MODE
0766 ED5B455A            LD DE,(ATTRP)
076A 26B8                LD H,0B8H         ;END OF MODE 1 ATTRS
076C 010280              LD BC,8002H       ;DO 1800H BYTES
076F A7                  AND A
0770 2010                JR NZ,CLSG        ;JR IF MODE 1
0772
0772 269B                LD H,9BH          ;END OF MODE 0 ATTRS
0774 010130              LD BC,3001H       ;DO 0300H BYTES
0777 1809                JR CLSG
0779
0779 26E0        CLS1:   LD H,0E0H
077B ED5B485A            LD DE,(M23PAPP)
077F 010600              LD BC,0006H       ;CLEAR DFFF-8000H WITH M3PAPP
0782
0782 CDA83F      CLSG:   CALL SPSSR        ;STORE PAGE, SELECT SCREEN
0785 53                  LD D,E
0786 F3                  DI
0787 ED73C85A            LD (TEMPW1),SP
078B 2E00                LD L,0
078D F9                  LD SP,HL
078E
078E D5          CLSLP:  PUSH DE
078F D5                  PUSH DE
0790 D5                  PUSH DE
0791 D5                  PUSH DE
0792 D5                  PUSH DE
0793 D5                  PUSH DE
0794 D5                  PUSH DE
0795 D5                  PUSH DE           ;DO 16 BYTES AT A TIME AT ABOUT 7 Ts/BYTE
0796 10F6                DJNZ CLSLP
0798
0798 0D                  DEC C
0799 20F3                JR NZ,CLSLP       ;DO 1000H BYTES
079B
079B ED7BC85A            LD SP,(TEMPW1)
079F FB                  EI
07A0 C3BF3F              JP RCURPR         ;RESET CURRENT UR PAGE
07A3
07A3
07A3             ;BASIC"S PRINT COMMAND
07A3
07A3 0E03        LPRINT: LD C,3
07A5 21                  DB 21H
07A6
```

```
07A6 0E02       PRINT:      LD C,2
07A8 CDC63A                 CALL RUNFLG
07AB 79                     LD A,C
07AC 21BA5A                 LD HL,INQUFG
07AF CBC6                   SET 0,(HL)          ;"IN QUOTES" - MEANS KEYWORDS NOT EXPANDED
07B1 DC6311                 CALL C,SETSTRM
07B4 CD8E11                 CALL TEMPS
07B7
07B7 DF         PRINT2:     RST 18H
07B8 CDED07                 CALL PRTERM
07BB 280F                   JR Z,PRINT3         ;JR IF E.G. PRINT :
07BD
07BD CDD807     MPRSEPLP:   CALL PRSEPR
07C0 C8                     RET Z               ;RET IF TERMINATOR FOUND
07C1
07C1 30FA                   JR NC,MPRSEPLP      ;LOOP UNTIL A NON-SEPARATOR IS FOUND
07C3
07C3 CDF807                 CALL PRITEM
07C6 CDD807                 CALL PRSEPR
07C9 C8                     RET Z
07CA 30F1                   JR NC,MPRSEPLP
07CC
07CC FE29       PRINT3:     CP ")"
07CE C8                     RET Z               ;AVOID CR AFTER EMBEDDED PRINT ITEMS IN INPUT
07CF                                            ;E.G. INPUT "OLD VALUE:";(X);" NEW:";X
07CF
07CF            ;ENTRY FROM INPUT. CR IF RUNNING
07CF
07CF 0E0D       RUNCR:      LD C,0DH
07D1
07D1            ;PRINT C IF RUNNING
07D1
07D1 CDC63A     PRCIFRN:    CALL RUNFLG
07D4 D0                     RET NC
07D5
07D5 79                     LD A,C
07D6 D7                     RST 10H
07D7 C9                     RET
07D8
07D8            ;PRINT SEPARATORS - CONSIDER ;/,/"
07D8
07D8            ;EXIT: Z IF SEPARATOR FOLLOWED BY TERMINATOR, NZ, CY IF NOT SEPARATOR, NZ, NC
07D8            ;IF SEPARATOR NOT FOLLOWED BY TERMINATOR.
07D8
07D8 DF         PRSEPR:     RST 18H
07D9 FE3B                   CP ";"
07DB 280F                   JR Z,PRSEPR3
07DD
07DD 0E06                   LD C,6
07DF FE2C                   CP ","
07E1 2806                   JR Z,PRSEPR2
07E3
07E3 FE27                   CP "'"
07E5 37                     SCF
07E6 C0                     RET NZ
07E7
07E7 0E0D                   LD C,0DH
07E9
07E9 CDD107     PRSEPR2:    CALL PRCIFRN        ;PRINT C IF RUNNING - CHR$ 6 FOR COMMA, CR FOR """
07EC
07EC E7         PRSEPR3:    RST 20H             ;SKIP SEPARATOR
07ED
07ED FE29       PRTERM:     CP ")"
07EF C8                     RET Z
07F0
07F0 FE3A                   CP ":"
07F2 C8                     RET Z
07F3
07F3 FE0D                   CP 0DH
07F5 37                     SCF
07F6 3F                     CCF                 ;NC WITHOUT ALTERING ZERO FLAG
07F7 C9                     RET
07F8
07F8
07F8 DF         PRITEM:     RST 18H
07F9 FE88                   CP TABTOK
07FB 200B                   JR NZ,PRITEM2
07FD
07FD CD573A                 CALL SSYNTAX6
0800
0800 CD2E1D                 CALL GETINT
0803 51                     LD D,C
0804 3E17                   LD A,17H
0806 1815                   JR ATSR4
0808
0808 FE87       PRITEM2:    CP ATTOK
080A 2017                   JR NZ,PRITEM4
080C
080C CD5E3A                 CALL SSYNTAX8
080F
080F CD331D                 CALL GETBYTE
0812 F5                     PUSH AF             ;COL
0813 CD331D                 CALL GETBYTE        ;C=ROW
0816 51                     LD D,C
0817 F1                     POP AF
0818 5F                     LD E,A              ;D=ROW, E=COL
0819 FE                     DB 0FEH             ;"JR+1"
081A
081A            ;PRINT AT LINE A, COL E
081A
081A 57         ATSR2:      LD D,A
081B
081B 3E16       ATSR3:      LD A,16H            ;"AT" CONTROL CODE
```

```
081D
081D D7          ATSR4:     RST 10H
081E 7A                     LD A,D
081F D7                     RST 10H
0820 7B                     LD A,E
0821 D7                     RST 10H
0822 C9                     RET
0823
0823 CD6613      PRITEM4:   CALL CITEMSR
0826 D0                     RET NC
0827
0827 FE23                   CP "#"
0829 CA3511                 JP Z,PRHSH2
082C
082C CD083B                 CALL EXPTEXPR
082F D0                     RET NC              ;RET IF NOT RUNNING
0830
0830 DBFB                   IN A,(URPORT)
0832 F5                     PUSH AF
0833 2004                   JR NZ,PRITEM5       ;JR IF NUMERIC EXPR.
0835
0835 CDDC3F                 CALL GETSTRING      ;UNSTACK STRING, SEL PAGE
0838 BF                     CP A                ;Z
0839
0839 C47E01      PRITEM5:   CALL NZ,JPFSTRS     ;GET STR$ OF NUMBER AS BC BYTES AT (DE)
083C
083C CD1300                 CALL PRINTSTR
083F F1                     POP AF
0840 D3FB                   OUT (URPORT),A
0842 C9                     RET
0843
0843
0843
0843             ;LPT = LINE PTR TABLE
0843             ;TABLE OF 30 BYTES, EACH 0 IF NO LINE NUMBER ON THAT LINE, OR FF IF THERE IS.
0843             ;LNPTR VAR. HOLDS NO OF LINE WITH CURSOR, OR >3FH IF NONE
0843
0843 F5          FUPDN:     PUSH AF             ;DIRECTION CODE - 0BH=UP, 09H=DOWN
0844 CD5E11                 CALL STREAMFE       ;"S" CHANNEL
0847 C1                     POP BC
0848 CD4E08                 CALL FUPDN2
084B C3A405                 JP STRM0
084E
084E 3A8D5B      FUPDN2:    LD A,(LNPTR)
0851 FE40                   CP 40H
0853 D2C605                 JP NC,AUTOLIST      ;JP IF THERE IS NO CURSOR ON-SCREEN
0856
0856 78                     LD A,B
0857 FE0B                   CP 0BH              ;UP
0859 207A                   JR NZ,LPD           ;JR IF DOWN
085B
085B             ;CURSOR UP
085B
085B 2A495C                 LD HL,(EPPC)
085E E5                     PUSH HL
085F CD4D1A                 CALL FNDLINE
0862 62                     LD H,D
0863 6B                     LD L,E              ;HL AND DE PT TO LINE *BEFORE* EPPC. DE USED LATER
0864 46                     LD B,(HL)
0865 23                     INC HL
0866 4E                     LD C,(HL)           ;BC=PREV LINE NO
0867 ED43495C               LD (EPPC),BC
086B E1                     POP HL              ;OLD EPPC
086C A7                     AND A
086D ED42                   SBC HL,BC
086F C8                     RET Z               ;RET IF AT TOP OF PROG - EPPC UNCHANGED
0870
0870 D5                     PUSH DE
0871 F7                     RST 30H
0872 5CDE                   DW EROC2            ;GET A/C=LNPTR, H=TABLE MSB, ERASE OLD CURSOR
0874 D1                     POP DE
0875
0875 A7          LPUL:      AND A
0876 280C                   JR Z,MWDN           ;JR IF AT WINDOW TOP ALREADY - MOVE WINDOW DOWN
0878
0878 3D                     DEC A
0879 C652                   ADD A,>LPT          ;ADD OFFSET
087B 6F                     LD L,A              ;HL=ADDR IN TABLE (MUST BE INSIDE A PAGE)
087C D652                   SUB A,>LPT
087E 46                     LD B,(HL)
087F 04                     INC B
0880 107C                   DJNZ LPC            ;JR IF FOUND A LINE WITH A LINE NUMBER
0882
0882 18F1                   JR LPUL             ;ELSE LOOP
0884
0884 D5          MWDN:      PUSH DE             ;START OF NEW EPPC
0885
0885             ;GET SCREEN LINES TAKEN BY PROGRAM LINE STARTING AT DE (LEN OF NEW EPPC)
0885
0885 3C                     INC A               ;A=1
0886 32B75A                 LD (DMPFG),A        ;"DUMP OUTPUT"
0889 2A6C5A                 LD HL,(SPOSNU)
088C E5                     PUSH HL
088D 2A575A                 LD HL,(WINDLHS)     ;L=LHS
0890 3E41                   LD A,41H
0892 91                     SUB C               ;LINES FROM TOP OF PREV CURSOR POSN
0893 67                     LD H,A
0894 226C5A                 LD (SPOSNU),HL      ;START AT HI LN, PAST ANY WINDBOT, NO SCROLLING
0897
0897 EB                     EX DE,HL            ;PT HL TO FIRST CHAR
0898 CDBF08                 CALL IOUTLNC        ;DUMMY PRINT OF LINE TO GET LENGTH
089B AF                     XOR A
089C 32B75A                 LD (DMPFG),A
```

```
089F 21585A               LD HL,WINDTOP
08A2 3A595A               LD A,(WINDBOT)
08A5 96                   SUB (HL)
08A6 3C                   INC A
08A7 67                   LD H,A
08A8 3A6D5A               LD A,(SPOSNU+1)   ;END LINE POSN
08AB D640                 SUB 40H           ;A=LINES TAKEN BY PROGRAM LINE
08AD BC                   CP H              ;CP WINDOW HEIGHT
08AE 3801                 JR C,MWDN2        ;JR IF LINE <WIND HEIGHT
08B0
08B0 7C                   LD A,H
08B1
08B1 E1          MWDN2:   POP HL
08B2 226C5A               LD (SPOSNU),HL
08B5 CD8A0B               CALL EDRSADN      ;SCROLL DOWN "A" LINES
08B8 2A575A               LD HL,(WINDLHS)   ;L=LHS, H=TOP
08BB 226C5A               LD (SPOSNU),HL    ;PRINT POSN AT 0,0
08BE E1                   POP HL            ;LINE START
08BF
08BF CD9006      IOUTLNC: CALL SPACAN       ;**
08C2
08C2 01D208      IOUTLN:  LD BC,OUTLINC
08C5
08C5 3E01        IOPCL:   LD A,1
08C7 32BD5A               LD (INDOPFG),A
08CA CD1600               CALL BCJUMP       ;CALL BC WITH INDENT O/P ON
08CD
08CD AF          IOPOF:   XOR A
08CE 32BD5A               LD (INDOPFG),A
08D1 C9                   RET
08D2
08D2 F7          OUTLINC: RST 30H
08D3 1473                 DW OUTLINE-8000H
08D5
08D5             ;CURSOR DOWN
08D5
08D5 2A495C      LPD:     LD HL,(EPPC)
08D8 E5                   PUSH HL
08D9 CD4309               CALL ADVEPPC      ;MOVE EPPC DOWN ONE - NEW EPPC IN DE
08DC E1                   POP HL
08DD A7                   AND A
08DE ED52                 SBC HL,DE
08E0 C8                   RET Z             ;RET IF EPPC HASN"T CHANGED - END OF PROG
08E1
08E1 19                   ADD HL,DE
08E2 E5                   PUSH HL           ;OLD EPPC
08E3 ED5B585A             LD DE,(WINDTOP)
08E7 7A                   LD A,D            ;BOT
08E8 93                   SUB E
08E9 3C                   INC A
08EA F5                   PUSH AF           ;LINE NUMBER OF BOTTOM LINE+1, REL. TO WINDOW
08EB F7                   RST 30H
08EC 5CDE                 DW EROC2          ;GET A/C=LNPTR, H=TABLE MSB, ERASE OLD CURSOR
08EE D1                   POP DE            ;D=BOTTOM LINE+1
08EF
08EF 3C          LPDL:    INC A
08F0 BA                   CP D
08F1 301C                 JR NC,MWUP        ;JR IF AT WINDOW BOT ALREADY - MOVE WINDOW UP
08F3
08F3 C652                 ADD A,>LPT        ;ADD OFFSET
08F5 6F                   LD L,A            ;HL=ADDR IN TABLE (MUST BE INSIDE A PAGE)
08F6 D652                 SUB A,>LPT
08F8 46                   LD B,(HL)         ;B=0 IF NO LINE NO, OR FF
08F9 04                   INC B
08FA 05                   DEC B
08FB 28F2                 JR Z,LPDL         ;LOOP IF NO LINE NO. ON LINE
08FD
08FD E1                   POP HL            ;JUNK OLD EPPC
08FE
08FE 4F          LPC:     LD C,A
08FF 3A585A               LD A,(WINDTOP)
0902 81                   ADD A,C
0903 328D5B               LD (LNPTR),A      ;NEW LINE WITH CURSOR
0906 79                   LD A,C
0907 1E05                 LD E,5
0909 CD1A08               CALL ATSR2
090C F7                   RST 30H
090D 6475                 DW PRLCU-8000H
090F
090F CD3E09      MWUP:    CALL ADVSTOP      ;MOVE TOP OF SCREEN PROG LINE DOWN BY ONE
0912 AF                   XOR A
0913 32535A               LD (OVERT),A      ;OVER 0 SO OVERPRINTING LOOKS OK
0916 5F                   LD E,A            ;LH COLUMN
0917 218C5C               LD HL,SCRCT
091A 73                   LD (HL),E         ;PREVENT SCROLL PROMPT
091B 21585A               LD HL,WINDTOP
091E 3A8D5B               LD A,(LNPTR)
0921 96                   SUB (HL)
0922 CD1A08               CALL ATSR2
0925 E1                   POP HL            ;OLD EPPC
0926 CD4D1A               CALL FNDLINE
0929 3ABC5A               LD A,(OLDSPCS)
092C 32BE5A               LD (NXTSPCS),A    ;** RESET SPACES TO SAME FOR 2ND PRINT
092F CDC208               CALL IOUTLN       ;OLD EPPC MAY OVERLAP SCREEN BOTTOM - SO PRINT IT
0932 3E0D                 LD A,0DH          ;AGAIN, PERHAPS FORCING SCROLLING.
0934 D7                   RST 10H           ;CR
0935 188B                 JR IOUTLN         ;PRINT NEW EPPC - IT WILL HAVE A ">" CURSOR
0937
0937
0937             ;ENSURE SYS VAR PTED TO BY HL CONTAINS A LINE NUMBER THAT EXISTS. (IF IT DOES
0937             ;NOT, REPLACE IT WITH NO. OF FOLLOWING "REAL" LINE)
0937
0937 7E          REALN:   LD A,(HL)
```

```
0938 23                    INC HL
0939 E5                    PUSH HL
093A 66                    LD H,(HL)
093B 6F                    LD L,A          ;HL=LINE NO
093C 180E                  JR ADVAC
093E
093E               ;ADVANCE LINE NUMBER IN SYS. VAR TO NEXT LINE IF POSSIBLE
093E               ;EXIT: DE=NEW LINE NUMBER, HL IS UNCHANGED
093E
093E 216C5C        ADVSTOP:  LD HL,SDTOP      ;SCREEN DISPLAY TOP PROGRAM LINE
0941 1803                  JR ADVAN
0943
0943 21495C        ADVEPPC:  LD HL,EPPC
0946
0946 7E            ADVAN:    LD A,(HL)
0947 23                    INC HL
0948 E5                    PUSH HL
0949 66                    LD H,(HL)
094A 6F                    LD L,A          ;HL=LINE NO FROM SYS VAR.
094B 23                    INC HL
094C
094C CD4D1A        ADVAC:    CALL FNDLINE     ;FIND ADDR OF NEXT LINE, OR PROG END
094F CDD604                CALL LNNM        ;GET LINE NO. IN DE (OR PREV LINE NO. IF PROG END)
0952 E1                    POP HL
0953 72                    LD (HL),D
0954 2B                    DEC HL
0955 73                    LD (HL),E
0956 C9                    RET
0957
0957                                            ;TABSR, CLS, PRINT
0957                                            ;FUPDN, IOUTLN, PRLCU, ZTENTS, LNLEN,
0957                                            ;ATSR, REALN, ADVAN
0957                       INCLUDE ROLL.SAM      ;CALCPIX, NEXTUP, NXTDOWN, CLSWIND, EDRS, CRTBF
0957
0957               ;*************************************************************************
0957               ;ROLL.SAM - EG ROLL DIR,PIX,X,Y,W,L
0957               ;              ROLL DIR,PIX
0957               ;              ROLL DIR
0957               ; SCROLL IS SIMILAR BUT NO WRAP-ROUND
0957               ; SCROLL CLEAR -   SCROLL PROMPT OFF
0957               ; SCROLL RESTORE - SCROLL PROMPT ON
0957
0957               ;FROM JUMP TABLE:
0957               ;B=PIX, C=DIR (1-4), HL=TOP LHS COORDS, D=LEN, E=W, A=ROLL/SCROLL
0957
0957 1D            JROLL:    DEC E
0958 32D05A                LD (TEMPB3),A    ;ROLL=FF, SCROLL=00
095B D5                    PUSH DE
095C E5                    PUSH HL
095D 78                    LD A,B
095E F5                    PUSH AF
095F 1865                  JR JROLL2
0961
0961 3EFF          ROLL:     LD A,0FFH
0963 1812                  JR RSCOMM
0965
0965 FEB3          SCROLL:   CP 0B3H          ;CLEARTOK
0967 2804                  JR Z,SETPROMPT   ;JR IF PROMPT TO BE TURNED OFF
0969
0969 D6BA                  SUB 0BAH         ;RESTORETOK - ZERO RESULT FOR "PROMPT ON"
096B 2009                  JR NZ,SCRNINOT
096D
096D 47            SETPROMPT: LD B,A
096E CD143B                CALL SABORTER    ;SKIP CLEAR/RESTORE
0971
0971 78                    LD A,B
0972 32BB5A                LD (SPROMPT),A   ;0=PROMPTS ON
0975 C9                    RET
0976
0976 AF            SCRNINOT: XOR A
0977
0977 32D05A        RSCOMM:   LD (TEMPB3),A    ;ROLL=FF, SCROLL=00
097A CDE43A                CALL EXPT1NUM    ;DIRECTION
097D FE2C                  CP ","
097F 2807                  JR Z,ROLL4       ;GET PIX IF SPECIFIED
0981
0981 CD153B                CALL CHKEND      ;ELSE CHECK END AND USE DEFAULT
0984 3E01                  LD A,1           ;OF 1-PIXEL ROLL
0986 180D                  JR ROLL5
0988
0988 CDE33A        ROLL4:    CALL SEXPT1NUM   ; PIXELS
098B FE2C                  CP ","
098D 2810                  JR Z,ROLL6       ;GET AREA IF SPECIFIED
098F
098F CD153B                CALL CHKEND
0992
0992 CD331D                CALL GETBYTE     ;PIXELS
0995
0995 21FFC0        ROLL5:    LD HL,0C0FFH     ;LEN=192, W-1=255 ARE DEFAULTS
0998 E5                    PUSH HL
0999 210000                LD HL,0000H      ;Y=0, X=0 (TOP LHS) ARE DEFAULTS
099C E5                    PUSH HL
099D 1823                  JR ROLL7
099F
099F CDD33A        ROLL6:    CALL SEXPT4NUMS  ;SKIP X,Y,W,L - SR SHARED WITH GRAB
09A2 CD153B                CALL CHKEND
09A5
09A5 CD331D                CALL GETBYTE     ;L
09A8 A7                    AND A
09A9 2833                  JR Z,IOORHP2     ;LENGTH MUST BE 1-255 INITIALLY
09AB
09AB F5                    PUSH AF          ;LENGTH
```

```
09AC CD2E1D               CALL GETINT       ;WIDTH. LEGAL=2-256
09AF CB81                 RES 0,C           ;EVEN WIDTHS ONLY
09B1 0B                   DEC BC            ;1-255
09B2 78                   LD A,B
09B3 A7                   AND A
09B4 2028                 JR NZ,IOORHP2
09B6
09B6 F1                   POP AF            ;L
09B7 47                   LD B,A            ;L,W-1 IN B,C
09B8 C5                   PUSH BC
09B9 CD8927               CALL GTFIDFCDS    ;B=Y, C=X (FAT COORDS FORCED)
09BC CB81                 RES 0,C           ;EVEN X ONLY (OR FOR THINPIX, MULTIPLES OF 4 ONLY)
09BE C5                   PUSH BC
09BF CD331D               CALL GETBYTE      ;A=PIX
09C2
09C2 F5         ROLL7:    PUSH AF
09C3 CD331D               CALL GETBYTE      ;DIRECTION TO C
09C6
09C6 C5         JROLL2:   PUSH BC
09C7 CD9512               CALL CHKMD23
09CA CDD211               CALL GRATEMPS     ;SET TEMPS FROM PERM LS OR US VARS - USE FOR SCROLL
09CD CDA83F               CALL SPSSR        ;STORE PAGE, SELECT SCREEN
09D0 C1                   POP BC            ;C=DIRECTION
09D1 F1                   POP AF            ;PIX
09D2 E1                   POP HL            ;COORDS
09D3 D1                   POP DE            ;L, W-1
09D4 A7                   AND A
09D5 2807                 JR Z,IOORHP2
09D7
09D7 47                   LD B,A            ;PIX
09D8 79                   LD A,C            ;DIR. 1234=L/U/R/D
09D9 3D                   DEC A
09DA FE04                 CP 4
09DC 3802                 JR C,ROLL75       ;ORIG DIR MUST BE 1-4
09DE
09DE CF         IOORHP2:  RST 08H
09DF 1E                   DB 30
09E0
09E0 7C         ROLL75:   LD A,H            ;TOP=0, BOT=191
09E1 82                   ADD A,D
09E2 38FA                 JR C,IOORHP2
09E4
09E4 FEC1                 CP 193
09E6 30F6                 JR NC,IOORHP2     ;ERROR IF AREA FALLS OFF BOTTOM
09E8
09E8 CB51                 BIT 2,C
09EA 2802                 JR Z,NTRDOWN
09EC
09EC 67                   LD H,A
09ED 25                   DEC H             ;H=Y IF ROLL DOWN
09EE
09EE 7D         NTRDOWN:  LD A,L
09EF 83                   ADD A,E           ;ADD X,W-1
09F0 38EC                 JR C,IOORHP2      ;JR IF OFF SCREEN ON RHS
09F2
09F2 7A                   LD A,D
09F3
09F3 D9                   EXX
09F4 47                   LD B,A            ;B"=LENGTH (1-192)
09F5 D9                   EXX
09F6
09F6 37                   SCF
09F7 CB1C                 RR H
09F9 CB1D                 RR L              ;HL=SCR ADDR (LHS, TOP IF LT, RT OR UP; BOT IF DN)
09FB 7B                   LD A,E
09FC C601                 ADD A,1           ;WIDTH=2-256. CY IF 256
09FE 1F                   RRA               ;GET WIDTH IN BYTES (1-128)
09FF 5F                   LD E,A            ;E=WIDTH IN BYTES
0A00 50                   LD D,B            ;D=PIX
0A01 CB41                 BIT 0,C           ;01=L,10=U,11=R,100=D
0A03 CAD90A               JP Z,RUPDN        ;JP IF UP OR DOWN
0A06
0A06 1D                   DEC E             ;E=W-1, IN BYTES (0-127)
0A07 1036                 DJNZ RLBYTE       ;JR IF MOVING MORE THAN 1 PIX - USE BYTES
0A09
0A09 51                   LD D,C
0A0A
0A0A            ;LEFT OR RIGHT BY 1 PIXEL (1 NIBBLE IN MODE 3)
0A0A            ;C/D=DIR, E=WIDTH-1 IN BYTES, HL=SCREEN ADDR, B"=LENGTH, A=WIDTH
0A0A
0A0A F7                   RST 30H
0A0B FEE9                 DW CRBBFN
0A0D 15                   DEC D             ;DEC DIR
0A0E 2007                 JR NZ,NTNRL       ;JR IF ROLL RIGHT
0A10
0A10 7D                   LD A,L
0A11 83                   ADD A,E
0A12 6F                   LD L,A            ;HL=RHS IF ROLL LEFT BY 1 PIX
0A13 7B                   LD A,E
0A14 ED44                 NEG
0A16 5F                   LD E,A            ;E=NEGATED WIDTH-1 IF ROLL LEFT. ALLOWS PT TO LHS
0A17
0A17 3AD05A     NTNRL:    LD A,(TEMPB3)     ;ROLL=FF, SCROLL=00
0A1A 57                   LD D,A
0A1B 0E80                 LD C,128          ;SCAN LEN
0A1D D9                   EXX
0A1E
0A1E D9         NLRLP:    EXX
0A1F 45                   LD B,L
0A20 7D                   LD A,L
0A21 83                   ADD A,E
0A22 6F                   LD L,A            ;PT TO OTHER END OF LINE. CY IF E -VE (ROLL LEFT)
0A23 3A515A               LD A,(M23PAPT)
```

```
0A26 14                     INC D
0A27 15                     DEC D
0A28 2807                   JR Z,ROLL8          ;JR IF SCROLL - A=BG COLOUR
0A2A
0A2A 7E                     LD A,(HL)           ;GET NIBBLE TO WRAP ROUND
0A2B 3004                   JR NC,ROLL8         ;JR IF ROLL RIGHT
0A2D
0A2D 0F                     RRCA                ;ELSE GET NIBBLE TO OTHER SIDE OF A
0A2E 0F                     RRCA
0A2F 0F                     RRCA
0A30 0F                     RRCA
0A31
0A31 68           ROLL8:    LD L,B              ;HL PTS TO ORIG LINE END AGAIN
0A32 E5                     PUSH HL
0A33 CD004D                 CALL CDBUFF
0A36 E1                     POP HL
0A37 0600                   LD B,0
0A39 09                     ADD HL,BC
0A3A D9                     EXX
0A3B 10E1                   DJNZ NLRLP          ;DO B" SCANS
0A3D
0A3D 183E                   JR RCURPH           ;RESET UR PORT
0A3F
0A3F              ;RIGHT OR LEFT BY BYTES
0A3F
0A3F 04           RLBYTE:   INC B               ;B=PIX (2+)
0A40 78                     LD A,B
0A41 1F                     RRA                 ;BYTES OF MOVEMENT (1+). CALL IT M
0A42 57                     LD D,A
0A43 7B                     LD A,E              ;WIDTH-1 IN BYTES (0-127)
0A44 92                     SUB D               ;BYTES OF MOVEMENT (M)
0A45 DADE09                 JP C,IOORHP2        ;JR IF M ISN"T LESS THAN WIDTH
0A48
0A48 3C                     INC A
0A49 F7                     RST 30H
0A4A 1CEA                   DW CRTBF
0A4C 37                     SCF                 ;CY=LEFT
0A4D 0D                     DEC C               ;Z IF LEFT, NZ IF RIGHT
0A4E 2803                   JR Z,RLBY2
0A50
0A50 7D                     LD A,L
0A51 83                     ADD A,E
0A52 6F                     LD L,A              ;HL PTS TO RHS IF RIGHT AND BYTE MOVING. NC
0A53
0A53 0C           RLBY2:    INC C
0A54 0600                   LD B,0
0A56 3AD05A                 LD A,(TEMPB3)
0A59 FE01                   CP 1                ;C IF SCROLL, NC IF ROLL
0A5B 0D                     DEC C               ;Z IF LEFT, NZ IF RIGHT
0A5C 7A                     LD A,D              ;A=M
0A5D D9                     EXX
0A5E 383D                   JR C,SCROLLLR
0A60
0A60 201E                   JR NZ,RRBMLP
0A62
0A62              ;ROLL LEFT BYTES
0A62
0A62 D9           RLBMLP:   EXX
0A63 4F                     LD C,A              ;BC=M
0A64 E5                     PUSH HL
0A65 E5                     PUSH HL             ;SCRN PTR
0A66 1103E0                 LD DE,RSBUFF
0A69 EDB0                   LDIR                ;SAVE M BYTES FROM LINE START, ADVANCE SRC
0A6B D1                     POP DE              ;ORIG HL
0A6C CD004D                 CALL CDBUFF         ;COPY THE SCAN
0A6F 2103E0                 LD HL,RSBUFF
0A72 04                     INC B               ;B=0
0A73 4F                     LD C,A              ;BC=M
0A74 EDB0                   LDIR                ;WRAP BYTES FROM BUFFER
0A76 E1                     POP HL              ;LINE START
0A77 0E80                   LD C,128
0A79 09                     ADD HL,BC           ;DROP 1 SCAN
0A7A D9                     EXX
0A7B 10E5                   DJNZ RLBMLP
0A7D
0A7D C3BF3F       RCURPH:   JP RCURPR
0A80
0A80              ;ROLL RIGHT BYTES
0A80
0A80 D9           RRBMLP:   EXX
0A81 4F                     LD C,A              ;BC=M
0A82 E5                     PUSH HL
0A83 E5                     PUSH HL             ;SCRN PTR
0A84 1182E0                 LD DE,RSBUFF+127
0A87 EDB8                   LDDR                ;SAVE M BYTES FROM LINE END, MOVE SRC PTR LEFT
0A89 D1                     POP DE              ;ORIG HL
0A8A CD004D                 CALL CDBUFF         ;MOVE THE SCAN
0A8D 2182E0                 LD HL,RSBUFF+127
0A90 04                     INC B
0A91 4F                     LD C,A              ;BC=M
0A92 EDB8                   LDDR                ;WRAP BYTES FROM BUFFER
0A94 E1                     POP HL              ;LINE START
0A95 0E80                   LD C,128
0A97 09                     ADD HL,BC           ;DROP 1 SCAN
0A98 D9                     EXX
0A99 10E5                   DJNZ RRBMLP
0A9B
0A9B 18E0                   JR RCURPH
0A9D
0A9D 201C         SCROLLLR: JR NZ,SRBYPRE       ;JR IF SCROLL RIGHT
0A9F
0A9F              ;SCROLL LEFT BYTES
0A9F
```

```
0A9F D9          SLBMLP:    EXX
0AA0 4F                     LD C,A             ;BC=M
0AA1 E5                     PUSH HL            ;SCRN PTR
0AA2 54                     LD D,H
0AA3 5D                     LD E,L
0AA4 09                     ADD HL,BC
0AA5 CD004D                 CALL CDBUFF        ;MOVE SCAN LEFT
0AA8 47                     LD B,A
0AA9 4F                     LD C,A
0AAA 3A515A                 LD A,(M23PAPT)
0AAD
0AAD 12          SLBBLP:    LD (DE),A          ;BLANK END OF SCAN
0AAE 13                     INC DE
0AAF 10FC                   DJNZ SLBBLP
0AB1
0AB1 79                     LD A,C
0AB2 E1                     POP HL
0AB3 0E80                   LD C,128
0AB5 09                     ADD HL,BC          ;DROP 1 SCAN
0AB6 D9                     EXX
0AB7 10E6                   DJNZ SLBMLP
0AB9
0AB9 18C2        RCURH2:    JR RCURPH
0ABB
0ABB             ;SCROLL RIGHT BYTES.
0ABB
0ABB A7          SRBYPRE:   AND A              ;NC FOR FIRST SBC
0ABC
0ABC D9          SRBMLP:    EXX
0ABD 4F                     LD C,A             ;BC=M
0ABE E5                     PUSH HL            ;SCRN PTR
0ABF 54                     LD D,H
0AC0 5D                     LD E,L
0AC1 ED42                   SBC HL,BC          ;DE PTS TO RHS. MOVE HL SLIGHTLY (M) BYTES LEFT
0AC3 CD004D                 CALL CDBUFF        ;MOVE SCAN RIGHT
0AC6 47                     LD B,A             ;B=M
0AC7 4F                     LD C,A             ;SAVE M BRIEFLY
0AC8 3A515A                 LD A,(M23PAPT)
0ACB
0ACB 12          SRBBLP:    LD (DE),A          ;BLANK LHS (M BYTES)
0ACC 1B                     DEC DE
0ACD 10FC                   DJNZ SRBBLP
0ACF
0ACF 79                     LD A,C             ;A=M
0AD0 E1                     POP HL
0AD1 0E80                   LD C,128
0AD3 09                     ADD HL,BC          ;DROP 1 SCAN. NC
0AD4 D9                     EXX
0AD5 10E5                   DJNZ SRBMLP
0AD7
0AD7 18E0                   JR RCURH2
0AD9
0AD9             ;*******************************************************************************
0AD9             ;RUPDN - ROLL/SCROLL UP OR DOWN
0AD9             ;ENTRY: E=WIDTH IN BYTES, HL=TOP LHS SCRN ADDR (UP) OR BOT LHS (DOWN)
0AD9             ;B"=LENGTH, D=PIX OF DISP, C=DIR (1-4)
0AD9
0AD9 CB51        RUPDN:     BIT 2,C            ;010=UP, 100=DOWN
0ADB 018000                 LD BC,128          ;BC = DISP TO ROW BELOW IF ROLL UP
0ADE 2801                   JR Z,RUPDN2
0AE0
0AE0 05                     DEC B              ;BC=FF80=-128 IF ROLL DOWN
0AE1
0AE1             ;THIS ENTRY FROM EDRS WITH BC SET UP
0AE1
0AE1 7B          RUPDN2:    LD A,E
0AE2 F7                     RST 30H
0AE3 23EA                   DW CRTBFI       ;CREATE BUFFER OF A LDI"S
0AE5 7B                     LD A,E
0AE6 08                     EX AF,AF'
0AE7 7A                     LD A,D             ;PIX
0AE8
0AE8 D9                     EXX
0AE9 4F                     LD C,A             ;C"=PIX
0AEA 78                     LD A,B             ;GET LENGTH FROM B"
0AEB 41                     LD B,C             ;B"=PIX
0AEC D9                     EXX
0AED
0AED 92                     SUB D              ;SUB LEN,PIX
0AEE DADE09                 JP C,IOORHP2       ;ERROR IF MOVEMENT IS GREATER THAN WIN LEN
0AF1
0AF1 F5                     PUSH AF            ;SAVE MAIN BLOCK SCANS, Z IF MOVE=WIND LEN
0AF2 3AD05A                 LD A,(TEMPB3)
0AF5 A7                     AND A
0AF6 C43F0B                 CALL NZ,RSSTBLK    ;STORE WRAPPED AREA IF ROLL
0AF9 F1                     POP AF             ;LINES IN MAIN BLOCK
0AFA 280E                   JR Z,RUPFIN        ;JR IF MOVEMENT=LENGTH OF WINDOW (CLS)
0AFC
0AFC D9                     EXX
0AFD 47                     LD B,A             ;B"=LINES IN MAIN BLOCK (AREA NOT WRAPPED)
0AFE D9                     EXX
0AFF
0AFF 7A                     LD A,D             ;PIX
0B00 54                     LD D,H
0B01 5D                     LD E,L             ;DE IS SCREEN DEST
0B02
0B02 09          ADDDISP:   ADD HL,BC
0B03 3D                     DEC A              ;DEC PIX TO MOVE BY
0B04 20FC                   JR NZ,ADDDISP      ;MOVE HL UP OR DOWN TO START OF MAIN BLOCK
0B06
0B06 CD530C                 CALL RSMOVSR
0B09 EB                     EX DE,HL
0B0A
```

```
0B0A EB          RUPFIN:    EX DE,HL           ;DE=SCRN DEST
0B0B 3AD05A                 LD A,(TEMPB3)
0B0E A7                     AND A
0B0F 2815                   JR Z,SCRUDBLK      ;JR IF SCROLL AND BLANKING OF NEW AREA NEEDED
0B11
0B11                        ;ELSE WRAP DATA FROM BUFFER
0B11 2103E0                 LD HL,RSBUFF
0B14 D9                     EXX
0B15 41                     LD B,C             ;B"=PIX
0B16
0B16             ;DE PTS TO BLOCK, BC=SGNED SCAN LEN, B"=SCANS TO DO
0B16
0B16 D9          SCRUDWRAP: EXX
0B17 D5                     PUSH DE            ;SCRN DEST
0B18 C5                     PUSH BC            ;DISP TO NEXT SCAN
0B19 CD004D                 CALL CDBUFF        ;COPY A SCAN
0B1C C1                     POP BC
0B1D D1                     POP DE
0B1E EB                     EX DE,HL
0B1F 09                     ADD HL,BC          ;MOVE SCRN PTR UP OR DOWN A SCAN
0B20 EB                     EX DE,HL
0B21 D9                     EXX
0B22 10F2                   DJNZ SCRUDWRAP     ;COPY "PIX" SCANS
0B24
0B24 1816                   JR RCUHP
0B26
0B26             ;DE PTS TO BLOCK, BC=SGNED SCAN LEN, C"=SCANS TO DO, M23PAPT=VALUE
0B26
0B26 EB          SCRUDBLK:  EX DE,HL           ;HL PTS TO BLOCK TO CLEAR
0B27 3A515A                 LD A,(M23PAPT)
0B2A D9                     EXX
0B2B 41                     LD B,C             ;B"=PIX
0B2C
0B2C D9          SCRUBOLP:  EXX
0B2D 54                     LD D,H
0B2E 5D                     LD E,L
0B2F 1C                     INC E
0B30 77                     LD (HL),A
0B31 C5                     PUSH BC
0B32 E5                     PUSH HL
0B33 CD024D                 CALL CDBUFF+2
0B36 E1                     POP HL
0B37 C1                     POP BC
0B38 09                     ADD HL,BC          ;MOVE UP OR DOWN A SCAN
0B39 D9                     EXX
0B3A 10F0                   DJNZ SCRUBOLP     ;BLANK "PIX" SCANS AT TOP OR BOTTOM
0B3C
0B3C C3BF3F      RCUHP:     JP RCURPR
0B3F             ;*************************************************************************
0B3F             ;RSSTBLK - STORE BLOCK. ALSO USED BY GRAB
0B3F             ;ENTRY: HL=SCRN ADDR (TOP OR BOT), BC=SGNED SCAN LEN, D=PIX, B"=PIX,A"=WIDTH
0B3F             ;EXIT: SAME EXCEPT B"=0. TEMPW2=SPACE
0B3F
0B3F D5          RSSTBLK:   PUSH DE
0B40 E5                     PUSH HL
0B41 08                     EX AF,AF'          ;WIDTH
0B42 5F                     LD E,A
0B43 210000                 LD HL,0
0B46 7A                     LD A,D             ;A=PIX
0B47 54                     LD D,H             ;DE=WIDTH
0B48
0B48 19          CALCSPLP:  ADD HL,DE
0B49 3D                     DEC A
0B4A 20FC                   JR NZ,CALCSPLP     ;CALC WIDTH*PIX=STRIP MEM USE
0B4C
0B4C 22CA5A                 LD (TEMPW2),HL     ;SAVE SPACE REQUIRED FOR DATA (FOR GRAB)
0B4F 1113E0                 LD DE,RSBUFF+16    ;E013H
0B52 19                     ADD HL,DE
0B53 3002                   JR NC,STSTPOK
0B55
0B55 CF                     RST 08H
0B56 24                     DB 36              ;"Stored area too big"
0B57                                           ;ERROR IF STRIP USES MORE THAN (8K-19 BYTES)
0B57                                           ;(16 BYTES FOR PUT STACK, 3 FOR CC,W,L)
0B57 E1          STSTPOK:   POP HL
0B58 E5                     PUSH HL
0B59 1103E0                 LD DE,RSBUFF
0B5C D9                     EXX
0B5D
0B5D D9          STSTPLP:   EXX
0B5E C5                     PUSH BC            ;SCAN LEN
0B5F E5                     PUSH HL            ;SCRN SRC
0B60 CD004D                 CALL CDBUFF
0B63 E1                     POP HL
0B64 C1                     POP BC
0B65 09                     ADD HL,BC          ;PT TO SCAN ABOVE OR BELOW (BC IS SIGNED SCAN LEN)
0B66 D9                     EXX
0B67 10F4                   DJNZ STSTPLP
0B69
0B69 D9                     EXX
0B6A E1                     POP HL             ;SCRN ADDR
0B6B D1                     POP DE             ;D=PIX
0B6C C9                     RET
0B6D             ;*************************************************************************
0B6D             ;CLS AND EDITOR SCROLL ROUTINES.
0B6D             ;CLEAR WINDOW - VARS DEFINE CHAR WINDOW
0B6D
0B6D 21595A      CLSWIND:   LD HL,WINDBOT
0B70 7E                     LD A,(HL)
0B71 2B                     DEC HL
0B72 96                     SUB (HL)           ;SUB WINDTOP
```

```
0B73 3C                    INC A             ;GET ROWS OF WINDOW LEN
0B74
0B74 0E02                  LD C,2            ;"UP"
0B76 CD660C                CALL EDRSSR       ;GET B"=LEN, D=DISP (SAME)
0B79 3A735A                LD A,(DEVICE)
0B7C A7                    AND A
0B7D 7A                    LD A,D
0B7E 2005                  JR NZ,CLSW1       ;JR IF NOT UPPER SCREEN
0B80
0B80 3A5D5A                LD A,(LSOFF)
0B83 82                    ADD A,D           ;INCLUDE "LEFT OVER" SCANS
0B84 57                    LD D,A
0B85
0B85 D9          CLSW1:    EXX
0B86 47                    LD B,A
0B87 D9                    EXX
0B88 1818                  JR EDRSF
0B8A
0B8A             ;SCROLL WINDOW DOWN BY "A" ROWS
0B8A
0B8A 1600        EDRSADN:  LD D,0            ;SCROLL LPT DOWN
0B8C F5                    PUSH AF
0B8D F7                    RST 30H
0B8E E1F3                  DW STENTS         ;SCROLL LINE PTR TABLE
0B90 F1                    POP AF
0B91 0E04                  LD C,4            ;DOWN
0B93 180A                  JR EDRS
0B95
0B95             ;SCROLL WINDOW UP BY 1 ROW
0B95
0B95 3E01        EDRS1UP:  LD A,1            ;LPT SCROLL BY 1 ROW
0B97 57                    LD D,A            ;SCROLL LPT UP AS D=1
0B98 F7                    RST 30H
0B99 E1F3                  DW STENTS
0B9B 3E01                  LD A,1            ;WINDOW SCROLL BY 1 ROW
0B9D
0B9D 0E02        EDRSAUP:  LD C,2            ;UPWARDS
0B9F
0B9F             ;EDRS - EDITOR"S ROLL/SCROLL ROUTINE
0B9F             ;ENTRY: WINDOW VARS DEFINE CHAR WINDOW, A=ROWS TO SCROLL, C=2 IF UP, 4 IF DN.
0B9F
0B9F CD660C      EDRS:     CALL EDRSSR
0BA2
0BA2 CDA83F      EDRSF:    CALL SPSSR        ;STORE PAGE, SELECT SCREEN
0BA5 3A405A                LD A,(MODE)
0BA8 FE02                  CP 2
0BAA D2D90A                JP NC,RUPDN
0BAD
0BAD CB51                  BIT 2,C
0BAF 012000                LD BC,32
0BB2 DD21040D              LD IX,NEXTDOWN    ;IN CASE MODE 0
0BB6 2807                  JR Z,EDRSM1       ;JR IF UP
0BB8
0BB8 01E0FF                LD BC,-32
0BBB DD21E70C              LD IX,NEXTUP      ;IN CASE MODE 0
0BBF
0BBF 3A405A      EDRSM1:   LD A,(MODE)
0BC2 A7                    AND A
0BC3 2821                  JR Z,EDRSM0       ;JR IF MODE 0
0BC5
0BC5 AF                    XOR A
0BC6 32515A                LD (M23PAPT),A    ;MAKE "PAPER COLOUR" BLANK PIXEL PATTERN
0BC9 C5                    PUSH BC           ;SIGNED SCAN LEN
0BCA D5                    PUSH DE           ;D=PIX,E=WIDTH
0BCB E5                    PUSH HL           ;SCREEN ADDR
0BCC D9                    EXX
0BCD C5                    PUSH BC           ;LEN IN SCANS
0BCE D9                    EXX
0BCF CDE10A                CALL RUPDN2
0BD2 D9                    EXX
0BD3 C1                    POP BC
0BD4 D9                    EXX
0BD5 E1                    POP HL
0BD6 CBEC                  SET 5,H           ;ADD 2000H - PT TO MODE 1 ATTR
0BD8 D1                    POP DE
0BD9 C1                    POP BC
0BDA 3A4E5A                LD A,(ATTRT)
0BDD 32515A                LD (M23PAPT),A    ;SET COLOUR FOR M1 ATTRIBUTE SCROLL
0BE0 CDA83F                CALL SPSSR
0BE3 C3E10A                JP RUPDN2
0BE6
0BE6             ;MODE 0 SCROLL
0BE6             ;D=PIX TO MOVE BY, E=WIDTH, HL=SCRN ADDR, BC=SGNED ATTR ROW LEN, B"=WIND LEN
0BE6
0BE6 7B          EDRSM0:   LD A,E
0BE7 F7                    RST 30H
0BE8 23EA                  DW CRTBFI
0BEA C5                    PUSH BC           ;+/-32
0BEB
0BEB D9                    EXX
0BEC 78                    LD A,B            ;GET PIX OF WINDOW LEN
0BED D9                    EXX
0BEE
0BEE 92                    SUB D             ;SUB WINDOW LEN, PIX TO MOVE BY="MAIN BLOCK" LEN
0BEF DADE09                JP C,IOORHP2      ;ERROR IF MOVED BY MORE THAN WIND LEN
0BF2
0BF2 D9                    EXX
0BF3 47                    LD B,A            ;MAIN BLOCK LEN IN B"
0BF4 D9                    EXX
0BF5
0BF5 7A                    LD A,D            ;PIX. Z IF WIND LEN=MOVEMENT (CLS)
0BF6 F5                    PUSH AF
0BF7 4B                    LD C,E            ;C=WIDTH
```

```
0BF8 42                     LD B,D            ;B=PIX
0BF9 54                     LD D,H
0BFA 5D                     LD E,L            ;DE=SCRN DEST
0BFB 2805                   JR Z,EDRSM0L1
0BFD
0BFD
0BFD CD2D00     EDRSM0P:    CALL IXJUMP       ;MOVE UP OR DOWN A SCAN
0C00 10FB                   DJNZ EDRSM0P      ;PT HL TO SRC
0C02                                          ;B=0 SO BC=WIDTH
0C02 F1         EDRSM0L1:   POP AF            ;PIX. Z/NZ
0C03 F5                     PUSH AF           ;PIX
0C04 D5                     PUSH DE
0C05 E5                     PUSH HL           ;SAVE FOR ATTR SCROLL
0C06 F5                     PUSH AF           ;PIX, Z/NZ
0C07
0C07 D9                     EXX
0C08 2814                   JR Z,EDRSM0L2
0C0A
0C0A D9         EDRSM0LP:   EXX
0C0B E5                     PUSH HL
0C0C D5                     PUSH DE
0C0D C5                     PUSH BC
0C0E CD004D                 CALL CDBUFF
0C11 C1                     POP BC
0C12 E1                     POP HL
0C13 CD2D00                 CALL IXJUMP       ;ADJ DEST PTR
0C16 EB                     EX DE,HL
0C17 E1                     POP HL
0C18 CD2D00                 CALL IXJUMP       ;ADJ SRC PTR
0C1B D9                     EXX
0C1C
0C1C 10EC                   DJNZ EDRSM0LP     ;LOOP FOR ALL SCANS
0C1E
0C1E 42         EDRSM0L2:   LD B,D            ;B"=ROWS FOR ATTR SCROLL
0C1F D9                     EXX
0C20
0C20 EB                     EX DE,HL          ;HL=SCRN DEST
0C21 D1                     POP DE            ;D=PIX TO MOVE BY (AND BLANK)
0C22 1E00                   LD E,00H
0C24
0C24 E5         EDRSM0DL:   PUSH HL           ;SCAN START
0C25 41                     LD B,C            ;USE B AS WIDTH COUNTER
0C26
0C26 73         EDRSM0BL:   LD (HL),E
0C27 2C                     INC L
0C28 10FC                   DJNZ EDRSM0BL
0C2A
0C2A E1                     POP HL
0C2B CD2D00                 CALL IXJUMP
0C2E 15                     DEC D
0C2F 20F3                   JR NZ,EDRSM0DL
0C31
0C31 E1                     POP HL
0C32 CD490C                 CALL CTAA         ;CONVERT SRC TO ATTR ADDR
0C35 EB                     EX DE,HL          ;SRC IN DE
0C36 E1                     POP HL
0C37 CD490C                 CALL CTAA         ;CONVERT DEST
0C3A EB                     EX DE,HL
0C3B F1                     POP AF            ;Z/NZ
0C3C C1                     POP BC            ;+/-32
0C3D C4530C                 CALL NZ,RSMOVSR
0C40 3A4E5A                 LD A,(ATTRT)
0C43 32515A                 LD (M23PAPT),A    ;SET COLOUR FOR M1 ATTRIBUTE SCROLL
0C46
0C46            ;DE PTS TO BLOCK, BC=SGNED SCAN LEN, C"=SCANS TO DO, A"=WIDTH, M23PAPT=VALUE
0C46
0C46 C3260B                 JP SCRUDBLK
0C49
0C49            ;CONVERT HL TO ATTR ADDR
0C49
0C49 7C         CTAA:       LD A,H
0C4A 0F                     RRCA
0C4B 0F                     RRCA
0C4C 0F                     RRCA
0C4D E603                   AND 3
0C4F F698                   OR 98H
0C51 67                     LD H,A
0C52 C9                     RET
0C53
0C53            ;SCROLL M0 ATTRS / MAIN R/S UP/DOWN ROUTINE
0C53
0C53 D9         RSMOVSR:    EXX
0C54
0C54 D9         UPDNLP:     EXX
0C55 E5                     PUSH HL           ;SCRN SRC PTR
0C56 D5                     PUSH DE           ;DEST
0C57 C5                     PUSH BC           ;DISP TO ROW ABOVE OR BELOW
0C58 CD004D                 CALL CDBUFF
0C5B C1                     POP BC            ;DISP
0C5C E1                     POP HL            ;DEST
0C5D 09                     ADD HL,BC         ;ADJUST BY SCAN LEN
0C5E EB                     EX DE,HL
0C5F E1                     POP HL
0C60 09                     ADD HL,BC         ;SRC IS AJUSTED BY SIGNED SCAN LEN
0C61 D9                     EXX
0C62
0C62 10F0                   DJNZ UPDNLP       ;DO B" SCANS
0C64
0C64 D9                     EXX
0C65 C9                     RET
0C66
0C66            ;EDRSSR
0C66            ;ACTION: GET E=WIDTH IN BYTES, HL=TOP LHS SCRN ADDR (UP) OR BOT LHS (DOWN)
```

```
0C66             ;B"=LENGTH, D=PIX OF DISP, KEEP C
0C66
0C66 D9          EDRSSR:     EXX
0C67 4F                      LD C,A             ;C"=ROWS TO MOVE BY
0C68 D9                      EXX
0C69
0C69 CDD10C                  CALL CALCPIX
0C6C F5                      PUSH AF            ;SAVE AMOUNT TO MOVE BY, IN SCANS (PIX)
0C6D AF                      XOR A
0C6E 32D05A                  LD (TEMPB3),A      ;"SCROLL"
0C71 ED5B575A                LD DE,(WINDLHS)    ;DE=TOP/LHS ROW
0C75 CB49                    BIT 1,C
0C77 201F                    JR NZ,EDRS1        ;JR IF SCROLL UP
0C79
0C79 3A595A                  LD A,(WINDBOT)     ;ELSE GET BOTTOM
0C7C 3C                      INC A
0C7D 57                      LD D,A
0C7E CDB73D                  CALL ANYDEADDR     ;GET ADDR 1 SCAN (SIC) LOWER THAN NEEDED - NOW
0C81 3A405A                  LD A,(MODE)        ;BACK UP BY 1
0C84 2180FF                  LD HL,0FF80H       ;MINUS SCAN LEN FOR M2 OR M3
0C87 FE02                    CP 2
0C89 300A                    JR NC,EDRS0        ;JR IF M2 OR M3
0C8B
0C8B 2EE0                    LD L,0E0H          ;MINUS SCAN LEN FOR M1=FFE0
0C8D 3D                      DEC A
0C8E 2805                    JR Z,EDRS0
0C90
0C90 EB                      EX DE,HL
0C91 CDE70C                  CALL NEXTUP        ;IF MODE 0
0C94 FE                      DB 0FEH            ;"JR+1"
0C95
0C95 19          EDRS0:      ADD HL,DE
0C96
0C96 EB                      EX DE,HL           ;DE=DESIRED SCREEN ADDR
0C97 BF                      CP A               ;SET Z
0C98
0C98             ;ENTRY AT EDRS1 IS NZ
0C98
0C98 C4B73D      EDRS1:      CALL NZ,ANYDEADDR  ;USES DE/A ONLY. GETS DE=SCRN ADDR
0C9B 21595A                  LD HL,WINDBOT
0C9E 7E                      LD A,(HL)
0C9F 2B                      DEC HL
0CA0 96                      SUB (HL)           ;SUB WINDTOP
0CA1 3C                      INC A              ;GET ROWS OF WINDOW LEN
0CA2
0CA2 D9                      EXX
0CA3 91                      SUB C
0CA4 57                      LD D,A             ;ROWS TO DO (IN CASE M0 ATTR SCROLL)
0CA5 81                      ADD A,C            ;WINDOW LEN IN ROWS
0CA6 CDD10C                  CALL CALCPIX
0CA9 47                      LD B,A             ;B"=SCANS OF WINDOW LEN
0CAA D9                      EXX
0CAB
0CAB 2B                      DEC HL
0CAC 2B                      DEC HL
0CAD 7E                      LD A,(HL)          ;WINDRHS
0CAE 23                      INC HL
0CAF 96                      SUB (HL)           ;SUB WINDLHS
0CB0 3C                      INC A              ;WIDTH IN CHARS.
0CB1 EB                      EX DE,HL           ;HL=SCREEN ADDR
0CB2 D1                      POP DE             ;D=AMOUNT TO MOVE BY IN SCANS
0CB3 5F                      LD E,A             ;MODE 0 OR 1 USES 1 BYTE/CHAR
0CB4 3A405A                  LD A,(MODE)
0CB7 FE02                    CP 2
0CB9 D8                      RET C
0CBA
0CBA FE03                    CP 3
0CBC 280D                    JR Z,EDRS3
0CBE
0CBE 3A355A                  LD A,(FL6OR8)
0CC1 A7                      AND A
0CC2 7B                      LD A,E
0CC3 2003                    JR NZ,EDRS2        ;JR IF 8 PIXEL, 2 BYTE CHARS
0CC5
0CC5 1C                      INC E
0CC6 CB3B                    SRL E
0CC8
0CC8 83          EDRS2:      ADD A,E            ;A=WIDTH*1.5, ROUNDED UP, WIDTH*2
0CC9 5F                      LD E,A
0CCA C9                      RET
0CCB
0CCB 7B          EDRS3:      LD A,E             ;MODE 3 USES 4 BYTES/CHAR
0CCC 87                      ADD A,A
0CCD 87                      ADD A,A
0CCE 5F                      LD E,A
0CCF C9                      RET
0CD0
0CD0 7A          CALCPIXD:   LD A,D
0CD1
0CD1             ;CALCULATE NO. OF PIXELS IN "A" ROWS. RESULT IN A. ADD 8 IF BOTTOM HALF OF
0CD1             ;DOUBLE-HEIGHT CHARACTER
0CD1
0CD1 C5          CALCPIX:    PUSH BC
0CD2 4F                      LD C,A             ;C=ROWS
0CD3 3A365A                  LD A,(CSIZE)
0CD6 D605                    SUB 5
0CD8 47                      LD B,A             ;B=HEIGHT-5 (AT LEAST 1)
0CD9 79                      LD A,C
0CDA 87                      ADD A,A
0CDB 87                      ADD A,A
0CDC 81                      ADD A,C            ;A=ROWS*5
0CDD
0CDD 81          CLPXL:      ADD A,C
```

```
0CDE 10FD                  DJNZ CLPXL        ;A=ROWS*6 IF B=1, ROWS*7 IF B=2 ETC
0CE0
0CE0 4F                    LD C,A
0CE1 3A825B                LD A,(DHADJ)      ;8 IF BOTTOM HALF OF DOUBLE-HEIGHT CHAR BEING
0CE4                                         ;PRINTED, ELSE 0
0CE4 81                    ADD A,C
0CE5 C1                    POP BC
0CE6 C9                    RET
0CE7
0CE7             ;MODE 0 MOVE HL UP BY 1 SCAN
0CE7
0CE7 25          NEXTUP:   DEC H
0CE8 7C                    LD A,H
0CE9 F6F8                  OR 0F8H
0CEB 3C                    INC A
0CEC C0                    RET NZ
0CED
0CED 7D                    LD A,L
0CEE D620                  SUB 32
0CF0 6F                    LD L,A
0CF1 D8                    RET C
0CF2
0CF2 7C                    LD A,H
0CF3 D6F8                  SUB 0F8H
0CF5 67                    LD H,A
0CF6 C9                    RET
0CF7
0CF7
0CF7             ;GET ADDR OF POSN 1 SCAN BELOW (HL) IN HL, MODES 0/1. USES AF,HL ONLY
0CF7             ;SCREEN$ SR
0CF7
0CF7 3A405A      NXTDOWN:  LD A,(MODE)
0CFA A7                    AND A
0CFB 2807                  JR Z,NEXTDOWN
0CFD
0CFD 3E20                  LD A,20H
0CFF 85                    ADD A,L
0D00 6F                    LD L,A
0D01 D0                    RET NC
0D02
0D02 24                    INC H
0D03 C9                    RET
0D04
0D04             ;USED BY ROLL/SCROLL
0D04
0D04 24          NEXTDOWN: INC H
0D05 7C                    LD A,H
0D06 E607                  AND 07H
0D08 C0                    RET NZ            ;NC=NO CRSSING OF CHAR BORDER
0D09
0D09 7D          NXTDOWN1: LD A,L
0D0A C620                  ADD A,32
0D0C 6F                    LD L,A
0D0D D8                    RET C             ;RET IF NEW THIRD
0D0E
0D0E 7C                    LD A,H
0D0F C6F8                  ADD A,0F8H        ;SET CY
0D11 67                    LD H,A
0D12 C9                    RET
0D13
0D13
0D13                       INCLUDE NMNLP.SAM     ;MAIN LOOP
0D13             ;MAINLP.SAM  - SAM MAIN LOOP
0D13
0D13             ;MAIN PARSER - CHECK LINE FOR SYNTAX
0D13
0D13 213B5C      LINESCAN:     LD HL,FLAGS
0D16 CBBE                      RES 7,(HL)      ;SIGNAL SYNTAX CHECK
0D18 AF                        XOR A
0D19 67                        LD H,A
0D1A 6F                        LD L,A
0D1B 22755A                    LD (IFTYPE),HL  ;LONG IF, "REF TYPE" (FN FLAG) NO
0D1E 32475C                    LD (SUBPPC),A   ;FIRST STATEMENT
0D21 323A5C                    LD (ERRNR),A    ;"OK" ERROR
0D24 CD7910                    CALL EVALLINO   ;SKIP ANY LINE NO.
0D27 3054                      JR NC,STMTLP1   ;JR IF LINE NO. IN RANGE
0D29
0D29 CF          NONSENSE:     RST 08H
0D2A 1D                        DB 29
0D2B
0D2B             ;ENTRY POINT FROM LOOP, RETURN ETC, TO ELINE. C=STAT TO GOTO
0D2B
0D2B 79          LOOPEL:       LD A,C
0D2C
0D2C 32445C      LOOPEL2:      LD (NSPPC),A
0D2F
0D2F             ;ENTRY POINT FOR RUNNING OR CHECKING THE E-LINE
0D2F
0D2F AF          LINERUN:      XOR A
0D30 32B05A                    LD (CLA+1),A    ;ELINE SHOWN BY ODD CURRENT LINE ADDR - ALLOWS
0D33                                           ;EG RETURN TO RECOGNIZE RET ADDR AS ELINE
0D33 21FFFF                    LD HL,0FFFFH    ;FFFF - PRINT LINE NO. WILL GIVE 0
0D36 22455C                    LD (PPC),HL
0D39 CD9110                    CALL AELP       ;ADDRESS ELINE, SET PAGES
0D3C EB                        EX DE,HL
0D3D 2A915A                    LD HL,(WORKSP)
0D40 2B                        DEC HL          ;DE PTS TO ELINE START, HL TO ELINE END
0D41 3A445C                    LD A,(NSPPC)
0D44 C3440E                    JP NEXTLINE
0D47
0D47             ;FINDER
0D47             ;CALLED BY DO AND DEF PROC WITH D=INTERVENING, E=TARGET, BYTE AFTER CALL=ERROR
0D47             ;IN CASE NOT FOUND. NO RETURN EVER MADE. SEARCH STARTS FROM CHAD.
```

```
0D47
0D47 CD9E1D     SEARCH:    CALL SEARCHALL
0D4A D20800                JP NC,0008H          ;JP IF NOT FND - USE BYTE AFTER CALL AS ERR CODE
0D4D                                            ;ELSE CLA AND CHAD HAVE BEEN SET, A=STAT WHERE
0D4D                                            ;TARGET FOUND (LOOP OR END PROC)
0D4D
0D4D            ;ENTRY WITH A=STAT, CHAD SET, CLA=LINE START OR 00?? IF ELINE.
0D4D            ;CHAD IS PAST END PROC, LOOP, LELSE, SELSE, END IF, NEXT (VAR SKIPPED).
0D4D            ;LOOP MIGHT HAVE "WHILE" OR "UNTIL" TO SKIP.
0D4D            ;CONTINUE EXECUTION AT CHAD. EXCHAD2 USED BY FOR, LIF
0D4D
0D4D D1                    POP DE               ;JUNK RET ADDR
0D4E 32475C                LD (SUBPPC),A
0D51
0D51 DBFB       EXCHAD2:   IN A,(URPORT)
0D53 CD9410                CALL STPGS        ;SET CHADP, NXTLNP, CLAPG **
0D56 2AAF5A                LD HL,(CLA)
0D59 24                    INC H
0D5A 25                    DEC H
0D5B 2810                  JR Z,STMHOP       ;JR IF ELINE - PPC AND NXTLINE CORRECT
0D5D
0D5D 56                    LD D,(HL)
0D5E 23                    INC HL
0D5F 5E                    LD E,(HL)            ;GET LINE NO.
0D60 23                    INC HL
0D61 ED53455C              LD (PPC),DE          ;AND SET SYS VAR
0D65 5E                    LD E,(HL)
0D66 23                    INC HL
0D67 56                    LD D,(HL)            ;LINE LEN
0D68 23                    INC HL
0D69 19                    ADD HL,DE
0D6A 229D5A                LD (NXTLINE),HL      ;PTR TO NEXT LINE SET
0D6D
0D6D DF         STMHOP:    RST 18H
0D6E FE8A                  CP WHILETOK
0D70 3812                  JR C,STMTLP2
0D72
0D72 FE8C                  CP UNTILTOK+1
0D74 DCF210                CALL C,SKIPCSTAT     ;IF WHILE/UNTIL, SKIP STATEMENT
0D77
0D77 180B                  JR STMTLP2
0D79
0D79            ;THIS PART IS ALSO MAIN RUN-TIME CONTROL
0D79
0D79 23         STMTLP:       INC HL
0D7A
0D7A 22975A     STMTLP05:     LD (CHAD),HL
0D7D
0D7D            ;FROM ELSE
0D7D
0D7D CD891D     STMTLP1:      CALL SETWORK
0D80 21475C                   LD HL,SUBPPC
0D83 34                       INC (HL)          ;INC STATEMENT NO.
0D84
0D84            ;FROM ELSE (SYNTAX CHECK)
0D84
0D84 2A975A     STMTLP2:      LD HL,(CHAD)
0D87
0D87 7E         STMTLP25:     LD A,(HL)
0D88 FE21                     CP 21H
0D8A 300B                     JR NC,STMTLP3    ;JR IF 21-FF
0D8C
0D8C FE0D                     CP 0DH
0D8E CA120E                   JP Z,LINEEND
0D91
0D91 23                       INC HL           ;SKIP 0-20 EXCEPT CR.
0D92 22975A                   LD (CHAD),HL
0D95 18F0                     JR STMTLP25
0D97
0D97 FE3A       STMTLP3:      CP ":"
0D99 28DE                     JR Z,STMTLP
0D9B
0D9B 11D40D                   LD DE,NEXTSTAT   ;CMDS RETURN TO NEXT STAT AFTER EXECUTION
0D9E
0D9E D5         ON4ENT:       PUSH DE
0D9F 227B5A                   LD (CSA),HL      ;CURRENT STATEMENT ADDR
0DA2 2AF45A                   LD HL,(CMDV)
0DA5 24                       INC H
0DA6 25                       DEC H
0DA7 C40500                   CALL NZ,HLJUMP   ;ALLOWS ADDING OF EXTRA CMDS
0DAA
0DAA 32745B                   LD (CURCMD),A    ;USED BY SAVE/LOAD ETC
0DAD D690                     SUB 90H
0DAF DADB32                   JP C,PROCS
0DB2
0DB2 FE67                     CP 0F7H-90H
0DB4 3059                     JR NC,NONSX
0DB6
0DB6 87                       ADD A,A          ;GET WORD DISPLACEMENT
0DB7 4F                       LD C,A
0DB8 0600                     LD B,0
0DBA 2ADA5B                   LD HL,(CMDADDRT)
0DBD 09                       ADD HL,BC        ;PT TO WORD IN TABLE OF ROUTINE START ADDRS
0DBE 0EFA                     LD C,LRPORT
0DC0 ED40                     IN B,(C)
0DC2 CBF0                     SET 6,B
0DC4 ED41                     OUT (C),B        ;ROM1 ON
0DC6 5E                       LD E,(HL)
0DC7 23                       INC HL
0DC8 56                       LD D,(HL)
0DC9 E7                       RST 20H          ;SKIP TO CHAR PAST CMD
0DCA CB7A                     BIT 7,D
0DCC 2004                     JR NZ,R1CMD      ;JR IF CMD IN ROM1
```

```
0DCE
0DCE CBB0                       RES 6,B
0DD0 ED41                       OUT (C),B       ;ROM1 OFF
0DD2
0DD2 EB         R1CMD:          EX DE,HL
0DD3 E9                         JP (HL)         ;EXECUTE CMD. CMDS CALL ABORTER FOR EARLY RETURN
0DD4                                            ;IN SYNTAX TIME.
0DD4
0DD4 CD630E     NEXTSTAT:       CALL BRKSTOP
0DD7
0DD7 CDD13F     NOBREAK:        CALL R1OCHP     ;ROM1 OFF, SELCHADP
0DDA 3A445C                     LD A,(NSPPC)
0DDD 3C                         INC A
0DDE 2821                       JR Z,STMTNEXT   ;JR IF NSPPC=FF - NO JUMP TO NEW LINE/STAT
0DE0
0DE0 2A425C                     LD HL,(NEWPPC)  ;LINE NUMBER TO JUMP TO
0DE3 24                         INC H
0DE4 CA2F0D                     JP Z,LINERUN    ;JR IF LINE FFXX - E LINE
0DE7
0DE7 25                         DEC H
0DE8 CD351A                     CALL FNDLNHL   ;HL=LINE START
0DEB F5                         PUSH AF
0DEC DBFB                       IN A,(URPORT)
0DEE CD9410                     CALL STPGS
0DF1 F1                         POP AF
0DF2 3A445C                     LD A,(NSPPC)
0DF5 282B                       JR Z,LINEUSE    ;JR IF LINE FOUND (FLAG FROM FNDLNHL)
0DF7
0DF7 A7                         AND A
0DF8 C25B0E                     JP NZ,STATLOST   ;INSIST RETURN AND SUCH ACTUALLY GOTO DESIRED
0DFB                                            ;STATEMENT - ONLY GOTO/GOSUB AVOID THIS
0DFB
0DFB 4E                         LD C,(HL)       ;GET LINE NO. MSB OR STOPPER
0DFC 0C                         INC C
0DFD 2023                       JR NZ,LINEUSE   ;JR IF NOT AT PROGRAM END
0DFF
0DFF CF         OKERR:          RST 08H
0E00 00                         DB 0            ;"OK"
0E01
0E01 DF         STMTNEXT:       RST 18H
0E02
0E02 FE3A       STMTNEXT1:      CP ":"
0E04
0E04 CA790D     STMTLPH:        JP Z,STMTLP
0E07
0E07 FE8D                       CP THENTOK      ;NEEDED BY 'IF'
0E09 28F9                       JR Z,STMTLPH
0E0B
0E0B FE0D                       CP 0DH
0E0D 2803                       JR Z,LINEEND
0E0F
0E0F CF         NONSX:          RST 08H
0E10 1D                         DB 29           ;NONSENSE
0E11
0E11 F1         REMARK:         POP AF          ;JUNK NEXTSTAT
0E12                                            ;IGNORE REST OF LINE
0E12
0E12 CD153B     LINEEND:        CALL ABORTER    ;RET IF SYNTAX CHECK
0E15 1803                       JR LNEND2
0E17
0E17            ;USED BY 'ON' IN RUNTIME
0E17
0E17 CDD13F     OLNEND:         CALL R1OCHP     ;ROM1 OFF, SELCHADP
0E1A
0E1A 2A9D5A     LNEND2:         LD HL,(NXTLINE)  ;GET ADDRESS OF FOLLOWING LINE - FOLLOW ON
0E1D 7E                         LD A,(HL)
0E1E 3C                         INC A
0E1F 28DE                       JR Z,OKERR      ;JR IF AT PROGRAM END. (FF=STOPPER)
0E21
0E21 AF                         XOR A           ;STAT 0/1. LATER, 0 RATHER THAN 1 SHOWS
0E22                                            ;WE ARE DEALING WITH A GOTO (OF A NON-EXISTENT
0E22                                            ;LINE) BUT THAT"S IRREL. HERE.
0E22
0E22 FE01       LINEUSE:        CP 1
0E24 CE00                       ADC A,0         ;0 BECOMES 1, REST SAME
0E26 CB74                       BIT 6,H
0E28 2808                       JR Z,NONEWCLAPG
0E2A
0E2A F5                         PUSH AF
0E2B CDF23F                     CALL INCURPAGE
0E2E
0E2E            ;JPED TO WITH A=PAGE, (SP)=STAT, HL=LINE START
0E2E
0E2E CD9410     RLEPI:          CALL STPGS     ;SET CLAPG, CHADP, NXTLINEP
0E31 F1                         POP AF
0E32
0E32 22AF5A     NONEWCLAPG:     LD (CLA),HL     ;CUR. LINE ADDR RECORDED FOR GOSUB RETURNS ETC.
0E35 56                         LD D,(HL)
0E36 23                         INC HL
0E37 5E                         LD E,(HL)
0E38 23                         INC HL
0E39 ED53455C                   LD (PPC),DE     ;UPDATE PPC
0E3D 5E                         LD E,(HL)
0E3E 23                         INC HL
0E3F 56                         LD D,(HL)
0E40 EB                         EX DE,HL        ;DE PTS TO LINE LEN MSB.
0E41 23                         INC HL          ;HL=TEXT LEN+1
0E42 19                         ADD HL,DE       ;HL PTS TO FIRST CHAR OF NEXT LINE
0E43 13                         INC DE          ;DE PTS TO FIRST CHAR IN LINE
0E44
0E44 229D5A     NEXTLINE:       LD (NXTLINE),HL
0E47 EB                         EX DE,HL        ;HL PTS TO FIRST CHAR IN LINE
0E48 57                         LD D,A          ;STAT NO.
```

```
0E49 3EFF                   LD A,0FFH
0E4B 32445C                 LD (NSPPC),A     ;SIGNAL NO JUMP
0E4E 82                     ADD A,D          ;A=STAT NO.-1
0E4F 32475C                 LD (SUBPPC),A
0E52 CA7A0D                 JP Z,STMTLP05    ;JP IF WE WANT FIRST STAT. -  LD (CHAD),HL
0E55
0E55 CDF810                 CALL SKIPSTATS   ;SKIP D STATS, END WITH CHAD PTING TO :/CR/THEN
0E58 CA020E                 JP Z,STMTNEXT1
0E5B
0E5B CF          STATLOST:  RST 08H
0E5C 1F                     DB 31            ;"Statement doesn't exist"
0E5D
0E5D CD690E      BRKCR:     CALL BRKTST
0E60 C0                     RET NZ
0E61
0E61 CF          BRCERR:    RST 08H
0E62 0E                     DB 14            ;"BREAK - CONTINUE to repeat"
0E63
0E63             ;CHECK BREAK, STOP IF IT IS PRESSED
0E63
0E63 CD690E      BRKSTOP:   CALL BRKTST
0E66 C0                     RET NZ           ;RET IF ESC NOT PRESSED
0E67
0E67 CF                     RST 08H
0E68 0F                     DB 15            ;"BREAK into program"
0E69
0E69             ;Z IF BREAK PRESSED AND BREAKDI=0
0E69
0E69 3EF7        BRKTST:    LD A,0F7H
0E6B DBF9                   IN A,(STATPORT)
0E6D E620                   AND 20H
0E6F C0                     RET NZ           ;RET IF ESC NOT PRESSED
0E70
0E70 3A415B                 LD A,(BREAKDI)   ;NZ IN BREAKDI DISABLES BREAK
0E73 A7                     AND A
0E74 C9                     RET
0E75
0E75             ;*********************************************************************
0E75             ;USED BY AUTO
0E75
0E75 CDCF05      AULL:      CALL AUL2        ;AUTO-LIST, AVOIDING EPPC->CLOSEST
0E78 180D                   JR MAINX         ;MAIN LOOP, AFTER AUTO-LIST CALL
0E7A
0E7A             ;MAIN CONTROLLING LOOP FOR ENTIRE INTERPRETER! ENTRY AT MAINEXEC
0E7A
0E7A CDA010      MAINEADD:  CALL INSERTLN
0E7D 3A3A5C                 LD A,(ERRNR)
0E80 A7                     AND A
0E81 C2ED0E                 JP NZ,MAINER     ;JR IF NO ROOM FOR LINE
0E84
0E84 CDC605      MAINEXEC:  CALL AUTOLIST
0E87
0E87 CD711D      MAINX:     CALL SETMIN
0E8A
0E8A CDA405      MAINELP:   CALL STRM0
0E8D CDA602                 CALL EDITOR
0E90 CD7238                 CALL TOKMAIN     ;TOKENIZE LINE
0E93 CD130D                 CALL LINESCAN    ;CHECK FOR CORRECT SYNTAX/INSERT 5-BYTE FORMS
0E96 3A475C                 LD A,(SUBPPC)
0E99 17                     RLA
0E9A 3005                   JR NC,MAINE1     ;JR IF 127 OR LESS STATEMENTS
0E9C
0E9C 3E21                   LD A,33          ;ELSE "No room for line"
0E9E 323A5C                 LD (ERRNR),A
0EA1
0EA1 3A3A5C      MAINE1:    LD A,(ERRNR)
0EA4 A7                     AND A
0EA5 2811                   JR Z,MAINE2      ;JR IF "ERROR"=0 (OK)
0EA7
0EA7 3A735A                 LD A,(DEVICE)
0EAA 3D                     DEC A
0EAB 2040                   JR NZ,MAINER     ;JR UNLESS DEVICE 1 (LOWER SCRN) IN USE - REPORT
0EAD
0EAD CD351F                 CALL ADDRELN
0EB0 CD6910                 CALL REMOVEFP
0EB3 CD8E04                 CALL RSPNS       ;RASP NOISE, CANCEL ERRNR
0EB6 18D2                   JR MAINELP
0EB8
0EB8 CD7910      MAINE2:    CALL EVALLINO    ;GET LINE NUMBER IN BC, ELSE BC=0, Z SET
0EBB DA290D                 JP C,NONSENSE    ;ERROR IF TOO BIG
0EBE
0EBE 20BA                   JR NZ,MAINEADD   ;INSERT LINE IF THERE IS A LINE NO
0EC0
0EC0 DF                     RST 18H
0EC1 FE0D                   CP 0DH
0EC3 28BF                   JR Z,MAINEXEC    ;JR IF LINE IS JUST <ENTER>
0EC5
0EC5 3A6A5C                 LD A,(FLAGS2)
0EC8 1F                     RRA
0EC9 DCF006                 CALL C,CLSUP
0ECC CDB506                 CALL CLSLOWER
0ECF 3A3A5A                 LD A,(UWTOP)
0ED2 47                     LD B,A
0ED3 3A6D5A                 LD A,(SPOSNU+1)
0ED6 90                     SUB B
0ED7 3C                     INC A            ;LINES USED IN US
0ED8 328C5C                 LD (SCRCT),A
0EDB 213B5C                 LD HL,FLAGS
0EDE CBFE                   SET 7,(HL)       ;"RUNNING"
0EE0 2B                     DEC HL
0EE1 AF                     XOR A
0EE2 77                     LD (HL),A        ;ERROR NR=0
0EE3 3C                     INC A
```

```
0EE4 32445C                 LD (NSPPC),A     ;START WITH A JUMP TO STATEMENT 1
0EE7 CDDE33                 CALL COMPILE     ;IF FLAG SAYS SO, CREATE LABELS, COMPILE DEF FNS
0EEA                                         ;AND DEF PROC ADDRESSES.
0EEA CD2F0D                 CALL LINERUN
0EED
0EED CDD13F     MAINER:     CALL R1OCHP
0EF0 FB                     EI
0EF1 21475C                 LD HL,SUBPPC
0EF4 7E                     LD A,(HL)
0EF5 A7                     AND A
0EF6 2004                   JR NZ,MAINER1
0EF8
0EF8 3A465B                 LD A,(ONSTORE)   ;FETCH REAL SUBPPC IF "ON" FIDDLED WITH SYS VAR
0EFB 77                     LD (HL),A
0EFC
0EFC 210000     MAINER1:    LD HL,0
0EFF 22545C                 LD (DEFADD),HL
0F02 22A35A                 LD (XPTR),HL
0F05 7C                     LD A,H
0F06 23                     INC HL
0F07 22165C                 LD (STREAMS+6),HL ;STREAM ZERO POINTS TO CHANNEL K
0F0A 32715C                 LD (FLAGX),A
0F0D 32885B                 LD (AUTOFLG),A   ;AUTO OFF
0F10 CDA13B                 CALL SETDISP     ;DISPLAY 0
0F13 CD711D                 CALL SETMIN
0F16
0F16 3A3A5C                 LD A,(ERRNR)
0F19 E6EF                   AND 0EFH
0F1B 284A                   JR Z,MAINER3     ;NO "ON ERROR" IF "OK" OR 'STOP' (0/10H)
0F1D
0F1D CDBD1C                 CALL KBFLUSH     ;FLUSH KEYBOARD BUFFER **
0F20 213B5C                 LD HL,FLAGS
0F23 CBFE                   SET 7,(HL)       ;SET RUNNING IN CASE "VAL" TURNED IT OFF
0F25 21455B                 LD HL,ONERRFLG
0F28 CB7E                   BIT 7,(HL)
0F2A CBBE                   RES 7,(HL)       ;BIT 7=TEMP FLAG (OFF). BIT 0=PERM FLAG UNCHANGED
0F2C 2839                   JR Z,MAINER3     ;JR IF TEMP FLAG *WAS* OFF
0F2E
0F2E CD2F10                 CALL ERRHAND2    ;SKIP PRINTING PART OF ERRHAND
0F31 21ED0E                 LD HL,MAINER
0F34 E5                     PUSH HL          ;ERR HANDLER ADDR
0F35 F7                     RST 30H
0F36 70EB                   DW SETUPVARS     ;CREATE LINO, STAT, ERROR
0F38 3A3A5C                 LD A,(ERRNR)
0F3B FE0F                   CP 15            ;"BREAK into program"
0F3D 200B                   JR NZ,MAINER2
0F3F
0F3F CDA519                 CALL CONTINUE2   ;CONTINUE GETS THE CORRECT VALUES IF BREAK
0F42                                         ;INTERRUPTED A JUMP
0F42 22455C                 LD (PPC),HL      ;PPC LDED WITH OLDPPC
0F45 15                     DEC D
0F46 7A                     LD A,D
0F47 32475C                 LD (SUBPPC),A    ;SAME STATEMENT FOR RETURN, NOT NEXT ONE
0F4A
0F4A 2A435B     MAINER2:    LD HL,(ERRLN)
0F4D CD351A                 CALL FNDLNHL
0F50 200E                   JR NZ,STATLH
0F52
0F52 23                     INC HL
0F53 23                     INC HL
0F54 23                     INC HL
0F55 23                     INC HL
0F56 3A425B                 LD A,(ERRSTAT)
0F59 57                     LD D,A
0F5A CDF810                 CALL SKIPSTATS
0F5D E7                     RST 20H          ;PT TO "ON ERROR"
0F5E FEDD                   CP 0DDH          ;ONERRORTOK
0F60
0F60 C25B0E     STATLH:     JP NZ,STATLOST
0F63
0F63 E7                     RST 20H          ;SKIP IT. PT TO E.G. "GOTO 10"/"GOSUB 50"/HANDLER
0F64                                         ;CLA/NXTLINE/PPC ETC MATCH LINE WITH ERROR STILL
0F64 C3870D                 JP STMTLP25
0F67
0F67 CDB506     MAINER3:    CALL CLSLOWER    ;SETS CHANNEL K ALSO
0F6A 213C5C                 LD HL,TVFLAG
0F6D CBEE                   SET 5,(HL)       ;"CLEAR LOWER SCREEN ON KEYSTROKE"
0F6F 2B                     DEC HL           ;PT TO FLAGS
0F70 CBBE                   RES 7,(HL)       ;"NOT RUNNING" SO FNDLN DOESN"T USE CLA DURING
0F72                                         ;EDITING
0F72 3A3A5C                 LD A,(ERRNR)
0F75 CD7B0F                 CALL ERRHAND1    ;PRINT REPORT, ETC.
0F78 C38A0E                 JP MAINELP
0F7B
0F7B FE50       ERRHAND1:   CP 50H
0F7D 2033                   JR NZ,EHZ
0F7F
0F7F            ;MGT MESSAGE GIVEN IF "REPORT" 50H
0F7F
0F7F AF                     XOR A
0F80 CDB03D                 CALL UTMSG       ;"  MILES GORDON TECHNOLOGY plc"
0F83                                         ;"    C 1989  SAM Coup"
0F83 21345A                 LD HL,BGFLG
0F86 3E82                   LD A,82H         ; e WITH AN ACCENT
0F88 77                     LD (HL),A        ;NZ=FOREIGN ON
0F89 D7                     RST 10H
0F8A 3600                   LD (HL),0
0F8C 3E20                   LD A," "
0F8E D7                     RST 10H
0F8F 3AB45C                 LD A,(PRAMTP)
0F92 3C                     INC A            ;16 OR 32
0F93 6F                     LD L,A
0F94 2600                   LD H,0
```

```
0F96 29                 ADD HL,HL          ;32 OR 64
0F97 29                 ADD HL,HL          ;64 OR 128
0F98 29                 ADD HL,HL          ;128 OR 256
0F99 29                 ADD HL,HL          ;256 OR 512
0F9A 44                 LD B,H
0F9B 4D                 LD C,L
0F9C F7                 RST 30H
0F9D ABF5               DW PRNUMB1
0F9F 3E4B               LD A,"K"
0FA1 D7                 RST 10H
0FA2
0FA2 CDB11C     WTFK:   CALL READKEY
0FA5 28FB               JR Z,WTFK          ;WAIT FOR A KEYPRESS
0FA7
0FA7 CDB506             CALL CLSLOWER
0FAA 3EFF               LD A,0FFH
0FAC 320056             LD (LINICOLS),A    ;TURN OFF RAINBOW SCREEN
0FAF C32F10             JP ERRHAND2
0FB2
0FB2 380E       EHZ:    JR C,EH0           ;JR IF NOT DOS ERR CODE
0FB4
0FB4 D651               SUB 51H            ;RANGE NOW 0+
0FB6 4F                 LD C,A
0FB7 3AC25B             LD A,(DOSFLG)
0FBA CDDF3F             CALL SELURPG       ;DOS AT 8000H
0FBD 2A1082             LD HL,(8210H)      ;DOS ERR MSGS
0FC0 79                 LD A,C
0FC1 DD                 DB 0DDH            ;"JR+3"
0FC2
0FC2 2AD45B     EH0:    LD HL,(ERRMSGS)
0FC5
0FC5 EB         EH15:   EX DE,HL
0FC6 F7                 RST 30H
0FC7 4CDD               DW POMSR           ;PRINT MESSAGE TO BUFFER (AND GET BC=LEN)
0FC9 3A565A             LD A,(WINDRHS)
0FCC 91                 SUB C              ;A=SPACE APART FROM MESSAGE, -1
0FCD FE0D               CP 13              ;ALLOW FOR 3, PLUS E.G. ", 12345:11"
0FCF F5                 PUSH AF
0FD0 3006               JR NC,EH1          ;JR IF OK TO PRINT ERROR MSG ON 1 LINE
0FD2
0FD2 3A3E5A             LD A,(LWTOP)
0FD5 326F5A             LD (SPOSNL+1),A    ;PRINT ON TOP LINE
0FD8
0FD8 C5         EH1:    PUSH BC
0FD9 3A3A5C             LD A,(ERRNR)
0FDC F5                 PUSH AF
0FDD F7                 RST 30H
0FDE A8F5               DW PRAREG          ;PRINT ERROR NUMBER AS 1 OR 2 DIGITS
0FE0 3E20               LD A," "
0FE2 D7                 RST 10H
0FE3 F1                 POP AF
0FE4 D602               SUB 2
0FE6 2025               JR NZ,EH2          ;JR IF NOT "MISSING VARIABLE"
0FE8                                       ;ELSE PRINT ITS NAME BEFORE " not found"
0FE8 47                 LD B,A
0FE9 213F51             LD HL,TLBYTE
0FEC 7E                 LD A,(HL)
0FED E61F               AND 1FH
0FEF 4F                 LD C,A             ;BC=LEN OF STR/ARRAY NAME, LEN-1 OF SIMPLE NUM
0FF0 CB6E               BIT 5,(HL)         ;NZ IF NUMERIC ARRAY
0FF2 23                 INC HL             ;PT TO FIRST LETTER
0FF3 54                 LD D,H
0FF4 5D                 LD E,L
0FF5 09                 ADD HL,BC          ;PT PAST END OF NAME (STR/ARRAY) OR TO END IF NUM
0FF6 03                 INC BC             ;TRUE LEN OF SIMPLE NUM VAR NAMES, OR EXTRA ROOM
0FF7                                       ;FOR "$" OR "("
0FF7 200B               JR NZ,PMV1         ;JR IF NUMERIC ARRAY
0FF9
0FF9 3A3B5C             LD A,(FLAGS)
0FFC CB77               BIT 6,A
0FFE 200A               JR NZ,PMV2         ;JR IF A (SIMPLE) NUMERIC VAR
1000
1000 3624               LD (HL),"$"
1002 1806               JR PMV2
1004
1004 3628       PMV1:   LD (HL),"("
1006 23                 INC HL
1007 3629               LD (HL),")"
1009 0C                 INC C              ;ALLOW FOR ")" IN LEN
100A
100A CD1300     PMV2:   CALL PRINTSTR      ;PRINT VAR NAME
100D
100D C1         EH2:    POP BC
100E 11C050             LD DE,MSGBUFF
1011 CD1300             CALL PRINTSTR      ;PRINT ERROR MESSAGE
1014 F1                 POP AF
1015 3E0D               LD A,0DH
1017 3805               JR C,EH3           ;JR IF USING 2 LINES
1019
1019 3E2C               LD A,","
101B D7                 RST 10H
101C 3E20               LD A," "
101E
101E D7         EH3:    RST 10H
101F ED4B455C           LD BC,(PPC)
1023 F7                 RST 30H
1024 ABF5               DW PRNUMB1         ;PRINT LINE NUMBER
1026 3E3A               LD A,":"
1028 D7                 RST 10H
1029 3A475C             LD A,(SUBPPC)
102C F7                 RST 30H
102D A8F5               DW PRAREG
102F
```

```
102F CDA504     ERRHAND2:  CALL CLEARSP
1032 3A3A5C                LD A,(ERRNR)
1035 A7                    AND A
1036 C8                    RET Z              ;NO CONTINUE AFTER "O.K." - RET
1037
1037 D610                  SUB 16             ;RESULT OF 0 IF "STOP statement", FF IF "BREAK
1039                                          ;into program", AND CY
1039 0600                  LD B,0
103B 88                    ADC A,B
103C 2008                  JR NZ,ERRHAND3     ;NO INC OF SUBPPC IF NEITHER STOP OR BREAK
103E
103E 3A745B                LD A,(CURCMD)
1041 FEC1                  CP 193             ;NEXT
1043 2801                  JR Z,ERRHAND3      ;B=0 NOW - CONTINUE AFTER BREAK INTO 'NEXT'
1045                                          ;REPEATS STATMENT
1045 04                    INC B
1046
1046 21445C     ERRHAND3:  LD HL,NSPPC
1049 7E                    LD A,(HL)
104A 36FF                  LD (HL),0FFH       ;CANCEL ANY JUMP
104C 2A425C                LD HL,(NEWPPC)
104F CB7F                  BIT 7,A
1051 2807                  JR Z,ERRHAND4      ;JR IF JUMP WAS ABOUT TO HAPPEN
1053
1053 3A475C                LD A,(SUBPPC)
1056 80                    ADD A,B            ;INCR. SUBPPC BY 1 IF STOP OR BREAK
1057 2A455C                LD HL,(PPC)
105A
105A 24         ERRHAND4:  INC H
105B C8                    RET Z              ;RET IF CONT WOULD HAVE BEEN TO E-LINE
105C
105C 25                    DEC H
105D 226E5C                LD (OLDPPC),HL
1060 32705C                LD (OSPPC),A       ;COPY NSPPC/NEW PPC OR SUBPPC/PPC TO OSPPC/OLDPPC
1063 C9                    RET
1064
1064
1064 E3         DFKNL:     EX (SP),HL         ;JUNK NEXT STAT RETURN, STACK PTR TO REST OF LINE
1065 CD010E                CALL STMTNEXT      ;CHECK SYNTAX FOR REST OF LINE
1068 E1                    POP HL             ;REMOVE FLOATING POINT FROM REST OF LINE, RET TO
1069                                          ;MAIN LOOP
1069
1069               ;REMOVE INVISIBLE 5-BYTE FORMS FROM (HL) TO 0DH
1069
1069 0E06       REMOVEFP:      LD C,6
106B 7E                        LD A,(HL)
106C D60E                      SUB 0EH
106E 47                        LD B,A
106F CC521E                    CALL Z,RECLAIM2
1072 7E                        LD A,(HL)
1073 23                        INC HL
1074 FE0D                      CP 0DH
1076 20F1                      JR NZ,REMOVEFP
1078
1078 C9                        RET
1079
1079
1079               ;GET LINE NO AT START OF ELINE TO BC, SET Z IF LN=0. CY IF TOO BIG.
1079
1079 CD9110     EVALLINO:  CALL AELP      ;ADDR ELINE, SET CHADP ETC
107C 22975A                LD (CHAD),HL
107F F7                    RST 30H
1080 0BE8                  DW SMBW
1082 DF                    RST 18H
1083 CDDE17                CALL INTTOFP
1086 CD3B1D                CALL FPTOBC
1089 D8                    RET C              ;RET IF >64K
108A
108A 78                    LD A,B
108B C601                  ADD A,1
108D D8                    RET C              ;RET IF >65279
108E
108E 78                    LD A,B
108F B1                    OR C
1090 C9                    RET                ;Z IF BC=0
1091
1091 CD351F     AELP:      CALL ADDRELN
1094
1094 E61F       STPGS:     AND 1FH
1096 32AE5A                LD (CLAPG),A
1099 32965A                LD (CHADP),A
109C 329C5A                LD (NXTLINEP),A
109F C9                    RET
10A0
10A0               ;*********************************************************************************
10A0               ;INSERT LINE - INSERT LINE BC FROM ELINE INTO PROGRAM
10A0
10A0 C5         INSERTLN:      PUSH BC            ;LINE NUMBER
10A1                       ;   LD HL,(INSLV)
10A1                       ;   INC H
10A1                       ;   DEC H
10A1                       ;   CALL NZ,HLJUMP
10A1
10A1 CDC333                    CALL SCOMP         ;DEF PROCS/DEF FNS AND LABELS NEED DOING -
10A4                                              ;ANY PRE-PASS OF LABELS ETC IS OBSOLETE.
10A4 2A915A                    LD HL,(WORKSP)
10A7 ED4B975A                  LD BC,(CHAD)       ;BC PTS AFTER LINE NUMBER
10AB 0A                        LD A,(BC)
10AC FE20                      CP " "
10AE 200B                      JR NZ,INSLN3
10B0
10B0 03                        INC BC
10B1 0A                        LD A,(BC)
```

```
10B2 FE0D                       CP 0DH
10B4 2001                       JR NZ,INSLN2      ;JR, LEAVING BC INCED, TO INC CHAD AND DELETE
10B6                                              ;FIRST SPACE IN A LINE LIKE: 10 test. PREVENTS
10B6                                              ;SPACES ACCUMULATING WITH MULTI-EDIT/ENTERS.
10B6 0B                         DEC BC            ;AVOID ANY ACTION IF E.G. 10 (space) CR
10B7
10B7 ED43975A   INSLN2:         LD (CHAD),BC
10BB
10BB 37         INSLN3:         SCF
10BC ED42                       SBC HL,BC         ;HL=LEN OF TEXT, INCLUDING 0DH
10BE 7C                         LD A,H
10BF FE3F                       CP 3FH
10C1 D24B3F                     JP NC,OOMERR      ;LIMIT LINE LEN TO 3EFFH
10C4
10C4 E3                         EX (SP),HL        ;STACK LEN, GET HL=LINE NO
10C5 22495C                     LD (EPPC),HL
10C8 CD461E                     CALL FNORECL      ;FIND/RECLAIM LINE
10CB C1                         POP BC            ;TEXT LEN
10CC 79                         LD A,C
10CD 3D                         DEC A
10CE B0                         OR B
10CF C8                         RET Z             ;RET IF TEXT IS JUST 0DH (LENGTH OF 1)
10D0
10D0 C5                         PUSH BC           ;TEXT LEN
10D1 03                         INC BC
10D2 03                         INC BC
10D3 03                         INC BC
10D4 03                         INC BC            ;GET LEN INCLUDING LN AND LEN BYTES
10D5                    ;       PUSH BC
10D5                    ;       PUSH HL
10D5                    ;       CALL GAPSZ
10D5                    ;       LD (4020H),HL  ;!!
10D5                    ;       POP HL
10D5                    ;       POP BC
10D5
10D5 CD1B1E                     CALL MAKEROOM     ;BC BYTES AT (HL)
10D8 ED4B495C                   LD BC,(EPPC)
10DC 70                         LD (HL),B
10DD 23                         INC HL
10DE 71                         LD (HL),C         ;ENTER LINE NUMBER
10DF 23                         INC HL
10E0 C1                         POP BC            ;TEXT LEN
10E1 71                         LD (HL),C
10E2 23                         INC HL
10E3 70                         LD (HL),B
10E4 CD791F                     CALL SPLITBC
10E7 23                         INC HL
10E8 EB                         EX DE,HL
10E9 DBFB                       IN A,(URPORT)
10EB 4F                         LD C,A            ;CDE=POINTS TO ROOM FOR TEXT
10EC CD3B1F                     CALL ADDRCHAD     ;TEXT START IN AHL
10EF C35E2A                     JP FARLDIR
10F2
10F2                ;SKIPCSTAT - SKIP CURRENT STATEMENT E.G. DATA, LABEL, DEF FN, LOOP UNTIL ETC.
10F2                ;ENTRY: CHAD=POSN
10F2
10F2 DF             SKIPCSTAT: RST 18H
10F3 110001                    LD DE,0100H       ;1 STAT, NOT IN QUOTES
10F6 1806                      JR SKIPS15
10F8
10F8                ;SKIPSTATS - FIND D"TH STATMENT FROM POSN.
10F8                ;ENTRY: D=STATMENTS TO SKIP,+1. HL=POSN
10F8                ;EXIT: CHAD PTS TO BEFORE REQUIRED STAT - TO ":" OR "THEN" OR CR.
10F8                ;Z,NC IF OK, Z, CY IF HIT END OF LINE AS STAT COUNTER REACHED ZERO, NZ, CY IF
10F8                ;HIT END OF LINE BEFORE STAT COUNT REACHED ZERO
10F8
10F8 2B             SKIPSTATS: DEC HL             ;COMP FOR INITIAL INC SO WE DON"T MISS SHORT STATS
10F9
10F9                ;FROM "ON"
10F9
10F9 AF             SKIPS0:    XOR A              ;"NOT IN QUOTES", NC
10FA 5F                        LD E,A
10FB 1820                      JR SKIPS5
10FD
10FD 23             SKIPS1:    INC HL
10FE
10FE 7E             SKIPS15:   LD A,(HL)
10FF FE0E                      CP 0EH
1101 CCA200                    CALL Z,NUMBER
1104
1104 FE22                      CP 22H
1106 2001                      JR NZ,SKIPS2
1108
1108 1D                        DEC E               ;"INSIDE STRING"
1109
1109 FE3A           SKIPS2:    CP ":"
110B 280C                      JR Z,SKIPS4
110D
110D FE8D                      CP THENTOK
110F 2808                      JR Z,SKIPS4
1111
1111 FE0D                      CP 0DH
1113 20E8                      JR NZ,SKIPS1
1115
1115 15                        DEC D               ;Z IF LINE END=DESIRED STAT.
1116 37                        SCF                 ;"HIT LINE END"
1117 1807                      JR SKIPS6
1119
1119 CB43           SKIPS4:    BIT 0,E
111B 20E0                      JR NZ,SKIPS1        ;IF ":"/"THEN"INSIDE STRING, JR, GET NEXT CHAR
111D
111D 15             SKIPS5:    DEC D               ;DEC "STATS TO SKIP" COUNTER
111E 20DD                      JR NZ,SKIPS1
```

```
1120
1120 22975A     SKIPS6:     LD (CHAD),HL         ;PT CHAD TO JUST BEFORE DESIRED STATMENT
1123 C9                     RET
1124
1124 CDC63A     DATA:       CALL RUNFLG
1127 38C9                   JR C,SKIPCSTAT    ;SKIP STAT IF RUNNING
1129
1129 CDED14     DATA1:      CALL SCANSR
112C FE2C                   CP " "
112E C0                     RET NZ            ;RET IF END OF STAT
112F
112F E7                     RST 20H
1130 18F7                   JR DATA1
1132
1132
1132                        INCLUDE MISC1.SAM      ;PRHSH, STRMINFO, SETSTRM, CHANFLAG,
1132              ;MISC1.SAM
1132              ;MODE, TEMPS, DATA, RESTORE
1132              ;CITEM, PERMS, SETSTRM
1132
1132              ;CALLED BY LIST
1132
1132 FE23         PRHSH1:     CP "#"
1134 C0                     RET NZ
1135
1135              ;ENTRY FROM PRINT
1135
1135 CD573A       PRHSH2:     CALL SSYNTAX6     ;NUMBER
1138
1138 CD331D                 CALL GETBYTE
113B CD4311                 CALL STRMINF2
113E 1829                   JR STSM2
1140
1140              ;ENTRY: FPCS HOLDS STREAM NUMBER. EXIT: DE=STREAM PTR, HL PTS TO STRM-H, C=STRM
1140
1140 CD331D       STRMINFO:   CALL GETBYTE      ;IN A AND C
1143
1143 FE11         STRMINF2:   CP 11H
1145 3013                   JR NC,INVSTRM     ;STREAMS 0-16 ARE LEGAL
1147
1147              ;FROM SETSTRM
1147
1147 FE10         STRMINF3:   CP 10H
1149 2003                   JR NZ,STRMINF4
114B
114B 3EFC                   LD A,0FCH         ;TRANSFORM STREAM 16 TO -4 (OUTPUT TO STRING)
114D 4F                     LD C,A
114E
114E C60B         STRMINF4:   ADD A,0BH         ;(07, 0B-1A IF FROM ABOVE, OR 06-0E IF FROM SETSTRM
1150 87                     ADD A,A           ;0EH, 16-34H OR 0CH-1CH
1151 6F                     LD L,A
1152 265C                   LD H,5CH          ;5C0C-5C34H
1154 5E                     LD E,(HL)
1155 23                     INC HL
1156 56                     LD D,(HL)
1157 7A                     LD A,D
1158 B3                     OR E              ;Z IF CLOSED
1159 C9                     RET
115A
115A CF           INVSTRM:  RST 08H
115B 15                     DB 21             ;"Invalid stream number"
115C
115C CF           SNOTOPER: RST 08H
115D 2F                     DB 47             ;"Stream is not open"
115E
115E 3EFE         STREAMFE: LD A,0FEH
1160 21                     DB 21H
1161
1161 3EFD         STREAMFD: LD A,0FDH
1163
1163              ;ENTRY: A=FCH-03H
1163
1163 32B25A       SETSTRM:  LD (STRNO),A      ;KEEP FOR DOS
1166 CD4711                 CALL STRMINF3
1169
1169 28F1         STSM2:    JR Z,SNOTOPER     ;JR IF STREAM NOT OPEN
116B
116B 2A4F5C                 LD HL,(CHANS)
116E 19                     ADD HL,DE         ;PT TO 2ND BYTE OF CHANNEL
116F 2B                     DEC HL
1170
1170 22515C       CHANFLAG: LD (CURCHL),HL
1173 23                     INC HL
1174 23                     INC HL
1175 23                     INC HL
1176 23                     INC HL
1177 7E                     LD A,(HL)         ;CHANNEL LETTER
1178 32745A                 LD (CLET),A       ;KEEP IT FOR INPUT TO LOOK AT
117B 0E02                   LD C,2
117D FE50                   CP "P"
117F 2809                   JR Z,STSMD
1181
1181 0D                     DEC C
1182 FE4B                   CP "K"
1184 2804                   JR Z,STSMD
1186
1186 0D                     DEC C
1187 FE53                   CP "S"
1189 C0                     RET NZ            ;ONLY SET "DEVICE" OR CALL TEMPS FOR K/S/P
118A
118A 79           STSMD:    LD A,C
118B 32735A                 LD (DEVICE),A     ;P=2,K=1,S=0
118E
```

```
118E             ;COPY PERMANENT GRAPHIC VARS TO TEMP VARS.
118E             ;TEMPS FOR PRINT (COLOUR EXPANSION TABLE IF NEEDED)
118E             ;HAVE TO COLOUR TABLE *AFTER* EACH PRINTED INK/PAPER/BRIGHT ALSO - CALL COLEX
118E
118E CDD611      TEMPS:     CALL GTEMPS
1191 2A385A                 LD HL,(UWRHS)
1194 ED5B3A5A               LD DE,(UWTOP)      ;UPPER WINDOW DATA
1198 3A735A                 LD A,(DEVICE)
119B 3D                     DEC A
119C 2007                   JR NZ,TEMUS        ;JR IF NOT LOWER SCREEN
119E
119E 2A3C5A                 LD HL,(LWRHS)
11A1 ED5B3E5A               LD DE,(LWTOP)
11A5
11A5 22565A      TEMUS:     LD (WINDRHS),HL
11A8 ED53585A               LD (WINDTOP),DE
11AC
11AC 3A405A      COLEX:     LD A,(MODE)
11AF D602                   SUB 2
11B1 D8                     RET C              ;RET IF M0/M1
11B2
11B2 0620                   LD B,32            ;COLOUR 32 BYTES IF M3
11B4 2002                   JR NZ,COLEX1       ;JR IF MODE 3
11B6
11B6 0610                   LD B,16            ;MODE 2 O/P NEEDS 16-BYTE COLEX TABLE
11B8
11B8 ED5B515A    COLEX1:    LD DE,(M23PAPT)    ;D=M3INKT, E=M3PAPT
11BC 21205B                 LD HL,EXTAB        ;EXPANSION TABLE
11BF D9                     EXX
11C0 21005B                 LD HL,CEXTAB       ;COLOURED EXPANSION TABLE
11C3 D9                     EXX
11C4 7A                     LD A,D
11C5 AB                     XOR E
11C6 4F                     LD C,A             ;C=PAPER XOR INK
11C7
11C7 79          COLEXLP:   LD A,C
11C8 A6                     AND (HL)           ;USE EXPANDED NIBBLES AS MASKS TO CHOOSE INK/PAP
11C9 2C                     INC L
11CA AB                     XOR E
11CB D9                     EXX
11CC 77                     LD (HL),A
11CD 2C                     INC L
11CE D9                     EXX
11CF 10F6                   DJNZ COLEXLP
11D1
11D1 C9                     RET
11D2
11D2
11D2             ;GRAPHIC TEMPS LEAVES EXPANSION TABLE ALONE (FOR SPEED) AND SETS "UPPER SCRN"
11D2
11D2 AF          GRATEMPS:  XOR A
11D3 32735A                 LD (DEVICE),A
11D6
11D6 21445A      GTEMPS:    LD HL,THFATP
11D9 114D5A                 LD DE,THFATT
11DC 010900                 LD BC,9
11DF EDB0                   LDIR
11E1 3A735A                 LD A,(DEVICE)
11E4 A7                     AND A
11E5 2812                   JR Z,TEMPS1        ;JR IF UPPER SCREEN, ELSE USE LOWER SCRN COLOURS
11E7
11E7 60                     LD H,B
11E8 68                     LD L,B
11E9 22535A                 LD (OVERT),HL      ;OVER 0, INVERSE 0
11EC 3A485C                 LD A,(BORDCR)
11EF 6F                     LD L,A
11F0 224E5A                 LD (ATTRT),HL      ;MASKT WILL BE ZERO
11F3 2A305A                 LD HL,(M23LSC)     ;MODES 2/3 LOWER SCREEN COLOURS
11F6 22515A                 LD (M23PAPT),HL    ;LD PAPER AND INK
11F9
11F9 3A405A      TEMPS1:    LD A,(MODE)
11FC FE02                   CP 2
11FE C8                     RET Z
11FF
11FF 3E01                   LD A,1
1201 324D5A                 LD (THFATT),A      ;NZ=FAT UNLESS MODE 2, THEN=COPY OF THFATP
1204 C9                     RET
1205
1205
1205             ;POKE, DPOKE, CALL, USR, PEEK & DPEEK ALLOW ADDRESSES OF 00000-1FFFF, RELATIVE
1205             ;TO THE CONTEXT BASE PAGE. 0000-3FFF WILL BE ROM0, 4000-7FFF WILL BE BASE PAGE,
1205             ;8000-BFFF WILL BE THE PAGE ABOVE THE BASE PAGE. 10000-13FFF WILL SWITCH PAGE
1205             ;3 ABOVE BASE PAGE INTO 8000H (IRRELEVANT EXCEPT TO NON-RELOCATABLE CALLED OR
1205             ;USRED CODE)
1205
1205 CDE43A      POKE:      CALL EXPT1NUM      ;EVAL ADDR
1208 CD853A                 CALL INSISCOMA
120B CD083B                 CALL EXPTEXPR
120E 2016                   JR NZ,POKE2        ;JR IF NUMERIC, ELSE DO POKE N,A$
1210
1210 D0                     RET NC             ;RET IF SYNTAX TIME
1211
1211 CD011D                 CALL STKFETCH      ;ADE=SRC, BC=LEN
1214 F5                     PUSH AF
1215 D5                     PUSH DE
1216 CD791F                 CALL SPLITBC
1219 CD8C3F                 CALL UNSTLEN
121C 4F                     LD C,A
121D 0D                     DEC C
121E EB                     EX DE,HL
121F CBFA                   SET 7,D            ;CDE=DEST ADDR
1221 E1                     POP HL
1222 F1                     POP AF             ;AHL=SRC ADDR
```

```
1223 C35E2A                JP FARLDIR
1226
1226 3004       POKE2:     JR NC,POKE3
1228
1228 EF                    DB CALC             ;ADDR,N
1229 06                    DB SWOP             ;N,ADDR
122A C8                    DB STOD0            ;N
122B 33                    DB EXIT
122C
122C 110000     POKE3:     LD DE,0             ;EXTRA NUMBERS COUNT
122F DF                    RST 18H
1230 180B                  JR POKE4
1232
1232 D5         POKENL:    PUSH DE             ;EXPR COUNT
1233 CDE33A                CALL SEXPT1NUM
1236 D1                    POP DE
1237 1C                    INC E
1238 CB6B                  BIT 5,E
123A C2290D                JP NZ,NONSENSE      ;LIMIT EXTRAS TO 31
123D
123D FE2C       POKE4:     CP ","
123F 28F1                  JR Z,POKENL
1241
1241 CD153B                CALL CHKEND
1244
1244 D5                    PUSH DE
1245
1245 EF                    DB CALC
1246 D8                    DB RCL0             ;N1,N2,N3...Nn,ADDR
1247 33                    DB EXIT
1248
1248 CD9F3F                CALL NPDPS          ;GET ADJUSTED ADDR TO HL, FORMER PAGE TO A
124B D1                    POP DE              ;DE=0 IF ONLY 1 NUMBER
124C 19                    ADD HL,DE
124D 1C                    INC E
124E
124E E5         PKALP:     PUSH HL             ;DEST ADDR FOR TOP NUMBER ON FPCS
124F D5                    PUSH DE             ;E=NUMBER OF NUMBERS ON FPCS
1250 CD651D                CALL FPTOA
1253 DA391D                JP C,IOORERR
1256
1256 2802                  JR Z,POKE5          ;JR IF +VE
1258
1258 ED44                  NEG
125A
125A D1         POKE5:     POP DE
125B E1                    POP HL
125C 77                    LD (HL),A
125D 2B                    DEC HL
125E 1D                    DEC E
125F 20ED                  JR NZ,PKALP
1261
1261 C9                    RET
1262
1262 CD5F3A     DPOKE:     CALL SYNTAX8        ;EVAL ADDR, NUMBER
1265
1265 CD2E1D                CALL GETINT         ;WORD TO POKE TO BC
1268 CD9F3F                CALL NPDPS
126B 71                    LD (HL),C
126C 23                    INC HL
126D 70                    LD (HL),B
126E C9                    RET
126F
126F
126F            ;PDPSUBR - USED BY POKE, DPOKE, PEEK, DPEEK, CALL
126F            ;ENTRY: ADDR ON FPCS.
126F            ;EXIT: IF ADDR IS 0-64K, THEN PAGING=ROM0 (OR BASE-1), BASE PAGE, BASE+1, BASE+2
126F            ;IF ADDR>64K, IT IS REDUCED TO 8000-BFFF RANGE AND PAGED IN. E.G. 10000H WOULD
126F            ;SWITCH PAGES BASE+3/BASE+4 IN AT 8000-FFFF, HL WOULD BE 8000H
126F            ;HL=ADDR, A=ORIG URPAGE. (LRPAGE UNCHANGED)
126F
126F C5         PDPSUBR:   PUSH BC             ;PRESERVE BC THROUGHOUT
1270 DBFB                  IN A,(251)
1272 F5                    PUSH AF
1273 CD8C3F                CALL UNSTLEN
1276 CBFC                  SET 7,H             ;AHL=ADDR IN PAGE, 8000-BFFF FORM
1278 11                    DB 11H              ;"JR+2"
1279
1279            ;PDPSR2. USED BY LOAD CODE (EXEC)
1279            ;ENTRY: AHL=EXEC ADDR
1279
1279 C5         PDPSR2:    PUSH BC
127A F5                    PUSH AF             ;KEEP STACK HAPPY
127B
127B FE04       PDPC:      CP 4
127D 300F                  JR NC,PDPSUBR4      ;JR IF NOT 0000-FFFF
127F
127F 0E02                  LD C,2              ;PAGING WILL BE ROM0,BASE,BASE+1,BASE+2
1281 B9                    CP C
1282 2809                  JR Z,PDPSUBR3       ;ADDR IS OK IF PAGE IS 2
1284
1284 3005                  JR NC,PDPSUBR2      ;JR IF PAGE 3 - ADD 4000H TO ADDR
1286
1286 CBBC                  RES 7,H             ;ADDR NOW 0000-3FFF
1288 A7                    AND A
1289 2802                  JR Z,PDPSUBR3       ;JR IF PAGE 0 - ADDR OK
128B                                           ;ELSE ADD 4000H FOR PAGE 1 ADDR
128B
128B CBF4       PDPSUBR2:  SET 6,H             ;ADD 4000H TO ADDR
128D
128D 79         PDPSUBR3:  LD A,C
128E
128E 3D         PDPSUBR4:  DEC A
```

```
128F CDDF3F                CALL TSURPG
1292 F1                    POP AF              ;ORIG URPORT
1293 C1                    POP BC
1294 C9                    RET
1295
1295
1295             ;CHECK MODE 2 OR MODE 3
1295
1295 3A405A      CHKMD23:      LD A,(MODE)
1298 FE02                      CP 2
129A D0                        RET NC
129B
129B CF          INVMERR:      RST 08H
129C 22                        DB 34           ;"Invalid screen mode"
129D
129D
129D             ;READ.  E.G. READ A, READ A$, READ LINE A$
129D
129D FE8C        READ:         CP LINETOK
129F F5                        PUSH AF                 ;Z IF LINE
12A0 200B                      JR NZ,READ2
12A2
12A2 CDF62C                    CALL SSYNTAX1           ;SKIP 'LINE', ASSESS VAR FOR ASSIGNMENT
12A5 213B5C                    LD HL,FLAGS
12A8 CB76                      BIT  6,(HL)
12AA C2290D                    JP NZ,NONSENSE          ;READ LINE NOT ALLOWED WITH NUMERICS
12AD                           ;SKIP NEXT CALL
12AD C4F72C      READ2:        CALL NZ,SYNTAX1
12B0 CDC63A                    CALL RUNFLG
12B3 D23C13                    JP NC,RJUNKFLG          ;JR IF SYNTAX TIME
12B6
12B6 DF                        RST  18H
12B7 22A95A                    LD (PRPTR),HL           ;SAVE CHAD IN AUTO-ADJUST VAR SO IF IT PTS TO
12BA 3A965A                    LD A,(CHADP)            ;E-LINE THE ASSIGNMENT WON'T BOLIX IT
12BD 32A85A                    LD (PRPTRP),A
12C0 CD381F                    CALL ADDRDATA           ;ADDRESS DATADD - SWITCH IN ITS PAGE, LD HL
12C3 32965A                    LD (CHADP),A            ;WITH ADDR PART
12C6 7E                        LD   A,(HL)
12C7 FE20                      CP   20H
12C9 281E                      JR   Z,READ3
12CB
12CB FE2C                      CP ","
12CD 281A                      JR   Z,READ3            ;SPACES AND COMMAS ARE OK TO READ FROM.
12CF                           ;OTHERWISE, NEED TO LOOK FOR NEXT DATA STAT.
12CF
12CF 22975A                    LD (CHAD),HL            ;NEEDED BY 'SRCHPROG'
12D2 1EB9                      LD E,0B9H               ;DATATOK
12D4 2AAF5A                    LD HL,(CLA)
12D7 E5                        PUSH HL
12D8 CD9C1D                    CALL SRCHPROG           ;LOOK FOR 'DATA' FROM CHAD ONWARDS
12DB D1                        POP DE
12DC ED53AF5A                  LD (CLA),DE
12E0 DBFB                      IN A,(251)              ;CHAD PTS TO JUST AFTER 'DATA'
12E2 32965A                    LD (CHADP),A            ;ASSUME FOUND...
12E5                       ;   LD (DATADDP),A          ;** BUG FIX
12E5 3806                      JR C,READ4              ;JR IF FOUND OK. PAGE MAY BE SWITCHED
12E7
12E7 CF                        RST 08H
12E8 03                        DB 3                    ;'DATA has all been read'
12E9
12E9 23          READ3:        INC HL
12EA 22975A                    LD (CHAD),HL
12ED
12ED F1          READ4:        POP  AF
12EE 2805                      JR   Z,READLN           ;JR IF 'READ LINE'
12F0
12F0 CDFA2A                    CALL VALFET1            ;ASSIGNMENT FOR NON-LINE READ
12F3 1831                      JR   READ7
12F5
12F5 01FFFF      READLN:       LD   BC,0FFFFH
12F8 E5                        PUSH HL
12F9
12F9 7E          READ5:        LD A,(HL)
12FA FE22                      CP 22H
12FC 2005                      JR NZ,READ6
12FE
12FE 23          RDSTRL:       INC HL
12FF 03                        INC BC
1300 BE                        CP (HL)
1301 20FB                      JR NZ,RDSTRL
1303
1303 CDA200      READ6:        CALL NUMBER
1306 22975A                    LD   (CHAD),HL          ;SKIP FP FORMS
1309 23                        INC  HL
130A 03                        INC  BC                 ;INC COUNT OF NON-INVISIBLE CHARS
130B CD753A                    CALL COMCRCO            ;CHECK IF COMMA, CR OR COLON
130E 20E9                      JR   NZ,READ5           ;LOOP UNTIL ONE IS FOUND
1310
1310 D1                        POP  DE
1311 C5                        PUSH BC                 ;LEN WITH NO FP FORMS
1312 ED52                      SBC  HL,DE              ;FIND DISTANCE CHAD MOVED
1314 44                        LD   B,H
1315 4D                        LD   C,L                ;LEN WITH FP FORMS,+1
1316 CD4D3F                    CALL SCOPYWK
1319 EB                        EX DE,HL                ;HL=ROOM START
131A E5                        PUSH HL
131B CD6910                    CALL REMOVEFP           ;FROM (HL) TO 0DH
131E D1                        POP  DE                 ;FIRST CHAR IN WKSPACE
131F C1                        POP  BC                 ;NON-INVISIBLE CHAR COUNT
1320 CDE71C                    CALL STKSTOREP          ;STORE REGS FOR A STRING
1323 CD0F2B                    CALL ASSIGN
1326
1326 DF          READ7:        RST  18H                ;GET CHAD
```

```
1327 228B5A                  LD (DATADD),HL      ;UPDATE DATA PTR
132A DBFB                    IN A,(251)
132C 328A5A                  LD (DATADDP),A
132F 2AA95A                  LD HL,(PRPTR)       ;GET REAL CHAD
1332 22975A                  LD (CHAD),HL
1335 3AA85A                  LD A,(PRPTRP)
1338 CDCE3F                  CALL SETCHADP
133B FE                      DB 0FEH             ;'JR +1'
133C
133C F1          RJUNKFLG:   POP AF              ;JUNK F
133D DF                      RST 18H
133E FE2C                    CP "," 
1340 C0                      RET NZ
1341
1341 E7                      RST 20H             ;SKIP COMMA
1342 C39D12                  JP READ
1345
1345
1345             ;CITEM.SAM - COLOUR ITEMS.
1345             ;CALLED BY SYNTAX 9
1345
1345 CDC63A      SYNT9SR:    CALL RUNFLG
1348 3011                    JR NC,SYN9SR1      ;JR IF SYNTAX CHECK
134A
134A AF                      XOR A
134B 32735A                  LD (DEVICE),A      ;UPPER SCREEN
134E CDD211                  CALL GRATEMPS
1351 214F5A                  LD HL,MASKT
1354 7E                      LD A,(HL)
1355 F6F8                    OR 0F8H
1357 77                      LD (HL),A
1358 23                      INC HL
1359 CBB6                    RES 6,(HL)         ;RES 6,PFLAGT=NOT PAPER 9
135B
135B DF          SYN9SR1:    RST 18H
135C
135C CD6613      CITEM:         CALL CITEMSR
135F D8                         RET  C
1360
1360 DF                         RST  18H
1361 CD813A                     CALL INSISCSC          ;CHECK FOR ,/; THEN SKIP
1364 18F6                       JR CITEM
1366
1366             ;CITEMSR - CALLED BY CITEM AND BASIC"S PRINT/INPUT
1366             ;EXIT WITH CY IF COLOUR ITEM NOT FOUND
1366
1366 FEA1        CITEMSR:       CP 0A1H                ;  INKTOK
1368 D8                         RET  C                 ;RET IF BELOW "INK"
1369
1369 FEA7                       CP 0A7H                ;  OVERTOK+1
136B 3F                         CCF
136C D8                         RET  C                 ;RET IF ABOVE "OVER"
136D
136D 4F                         LD C,A
136E E7                         RST  20H               ;SKIP INK/PAPER ETC
136F 79                         LD A,C
1370
1370             ;CALLED BY PERMS
1370 D691        COTEMP4:       SUB 0A1H-16            ;INK. CHANGE TO CONTROL CODE RANGE (16-21)
1372 F5                         PUSH AF
1373 CDE43A                     CALL EXPT1NUM          ;PARAM
1376 C1                         POP  BC                ;CONTROL CODE
1377 CDC63A                     CALL RUNFLG
137A D0                         RET NC                 ;ABORT WITH NC IF SYNTAX TIME
137B                                                   ;"COLOUR ITEM DEALT WITH"
137B C5                         PUSH BC
137C CD331D                     CALL GETBYTE
137F 57                         LD   D,A               ;PARAM TO D
1380 F1                         POP  AF                ;CONTINUE INTO PRCOITEM (ZX USED RST 8)
1381
1381
1381             ;CALLED BY PRINT WITH A=CONTROL CODE, D=PARAM
1381             ;NOTE: TRANSLATOR ALTERS INK/PAPER 8 OR 9 TO INK/PAPER 17/18. THERE WILL BE
1381             ;SOME FAILURES - E.G. INK N.
1381             ;INK I; BRIGHT B SELECTS INK I+8*B IN MODE 3
1381             ;BRIGHT IS IGNORED IN MODE 2, ALTHOUGH M0/1 SYS VARS ALTER.
1381             ;INK I WITH I>7 SELECTS INK I-8; BRIGHT 1
1381             ;OVER 0/1 ALTERS PFLAG AND OVERT, OVER 0-3 ALSO ALTERS GOVERT TO GIVE GRAB CMD"S
1381             ;XOR/AND OPTIONS.
1381
1381 F7          PRCOITEM:   RST 30H
1382 2BF0                    DW PRCOITEM2
1384 CDAC11                  CALL COLEX             ;COLOUR EXPANSION TABLE IF NEEDED
1387 A7                      AND A                  ;NC SHOWS COLOUR ITEM DEALT WITH
1388 C9                      RET
1389
1389 CDC63A      PERMS:      CALL RUNFLG
138C 3007                    JR NC,PER2             ;JR IF NOT RUNNING
138E
138E AF                      XOR A
138F 32735A                  LD (DEVICE),A          ;UPPER SCREEN
1392 CD8E11                  CALL TEMPS
1395
1395 3A745B      PER2:       LD A,(CURCMD)          ;GET CMD VALUE
1398 CD7013                  CALL COTEMP4           ;ALTER TEMP VALUES ACCORDING TO COLOUR CMDS
139B CD153B                  CALL CHKEND
139E
139E             ;CALLED BY CLS#
139E
139E 214E5A      PER3:       LD HL,ATTRT
13A1 11455A                  LD DE,ATTRP
13A4
13A4             ;USED BY SCREEN SCROLL
```

```
13A4
13A4 010800     LDIR8:        LD BC,8
13A7 EDB0                     LDIR                    ;COPY TEMP VALUES TO PERMS
13A9 C9                       RET                     ;(ATTRT-GOVERT)
13AA
13AA
13AA                                                  ;CHLETCHK, TEMPS, DATA, RESTORE, CITEM, PERMS,
13AA                                                  ;POKE, DPOKE, PDPSR, CHKM23
13AA                    INCLUDE LOOKVAR.SAM           ;SETUP VARS
13AA            ;LOOKVAR.SAM
13AA            ;ENTRY: A VARIABLE (STARTS WITH A LETTER) IS EXPECTED AT (CHAD)
13AA            ;(HL)/A=FIRST LETTER
13AA            ;ACTION: LOOK IN VARIABLES AREA FOR IT
13AA            ;EXIT:  Z IF VAR NOT FOUND: C=TYPE/LEN BYTE; BITS 4-0=NAME LENGTH, EXCLUDING
13AA            ;       FIRST CHARACTER. BIT 5 IS SET FOR NUMERIC ARRAYS.
13AA            ;       STRINGS: HL PTS TO FF TERMINATOR OF STRING/ARRAY VARS
13AA            ;       NUMBERS: HL PTS TO MSB OF PTR (FF) THAT TERMINATES LIST FOR LETTER.
13AA            ;
13AA            ;       NZ IF FOUND. HL POINTS TO VALUE IN VARS. (FOR NUMS, FIRST OF 5 BYTES,
13AA            ;       FOR STRINGS, LEN IN PAGES, FOLLOWED BY LEN MOD 16K AND TEXT)
13AA            ;       C=TYPE/LEN BYTE FROM VARS. NUMS: BIT 6 SET=FOR-NEXT
13AA            ;                          STRINGS/ARRAYS: BIT 6 SET=STR ARRY, BIT 5 SET=NUM ARRY
13AA            ;
13AA            ;ALWAYS: BIT 6,(FLAGS) IS SET FOR NUMERICS, RES FOR STRINGS
13AA            ;        NAME IS IN BUFFER AT "FIRLET" (FIRST LETTER IS CODED TABLE OFFSET)
13AA
13AA DF         LOOKVARS:  RST 18H
13AB
13AB            ;ENTRY FROM EVAL
13AB
13AB CD8D14     LKVARS2:   CALL NAMTOBUF      ;MOVE PAST NAME, COPY TO BUFFER, GET C=TYPE/LEN
13AE                                          ;HL=FLAGS, A=NAME TERMINATOR ($/(/OTHER)
13AE ED53975A              LD (CHAD),DE       ;PT PAST NAME (AND ANY (/$) IN BASIC LINE
13B2 79                    LD A,C
13B3 E660                  AND 60H
13B5 C22014                JP NZ,STARYLK      ;JP IF AN ARRAY
13B8
13B8 7E                    LD A,(HL)          ;FLAGS
13B9 87                    ADD A,A
13BA F22014                JP P,STARYLK       ;JP IF STRING
13BD
13BD 79                    LD A,C
13BE 323F51                LD (TLBYTE),A
13C1 D0                    RET NC             ;RET IF SYNTAX TIME. NZ="FOUND" (BIT 7=1 FOR MINUS)
13C2
13C2            ;LOOK FOR A NUMERIC VARIABLE ROUTINE
13C2
13C2            ;NVARS PTS TO A TABLE OF WORD PTRS. IF THE MSB OF THE PTR IS 0FFH, THERE
13C2            ; ARE NO MORE VARS STARTING WITH THAT LETTER. (SO TO CLEAR NUM VARS, INIT
13C2            ; TABLE WITH FFs.) IF PTR MSB<>FF THEN ADD PTR TO GET ADDR. OF
13C2            ; TYPE/LENGTH BYTE FOR NEXT VAR. NOTE: THERE MAY BE NO SECOND LETTER.
13C2            ; TYPE | PTR LSB | PTR MSB | SECOND LET | . . | LAST LET | VALUE
13C2            ; TYPE/LEN BYTE HAS BIT 7 SET FOR HIDDEN, BIT 6="FOR" VAR
13C2            ; BITS 4-0=LEN-1. (0 FOR 1-LETTER VAR NAME, 31 FOR 32 LETTER MAX)
13C2
13C2            ;ENTRY:  NAME IS IN BUFFER STARTING AT "FIRLET", LETTERS UPPER CASE, NO SPACES
13C2            ;        C=DESIRED TYPE/LEN BYTE
13C2            ;EXIT:   Z IF VAR NOT FOUND, C=TYPE/LEN BYTE.
13C2            ;        HL PTS TO LSB OF PTR (FFFF) THAT TERMINATES LIST FOR LETTER.
13C2
13C2            ;        NZ IF FOUND. HL POINTS TO VALUE IN VARS, IX-1 PTS TO TYPE BYTE,
13C2            ;        C=TYPE/LEN FROM VARS, DE=PTR ADDED TO PREV VAR. PTR MSB ADDR TO
13C2            ;        PT TO THIS VAR"S T/L BYTE
13C2
13C2 3A4051     NUMLOOK:   LD A,(FIRLET)
13C5 D661                  SUB 61H
13C7 87                    ADD A,A
13C8 5F                    LD E,A             ;LETTER TRANSFORMED TO WORD OFFSET (A=0, B=2..)
13C9 1600                  LD D,0
13CB CD1F1F                CALL ADDRNV        ;PT. HL AT NUMERIC VARS, PAGED IN
13CE 19                    ADD HL,DE          ;INDEX INTO TABLE OF WORD PTRS.
13CF FE                    DB 0FEH            ;"JR+1"
13D0
13D0 E1         NVMOLP:    POP HL
13D1
13D1 79         NVMLP:     LD A,C             ;DESIRED TYPE/LEN
13D2 5E                    LD E,(HL)
13D3 23                    INC HL             ;PTR=FFFFH IF NO MORE VARS START WITH REQUIRED
13D4 56                    LD D,(HL)          ; LETTER. CAUSES CARRY AND CHECK FOR FF IN NVSPOV
13D5 19                    ADD HL,DE          ;ELSE DE IS A PTR TO NEXT VAR STARTING
13D6                                          ; WITH REQUIRED LETTER.
13D6 3829                  JR C,NVSPOV        ;JR IF SEVERE PAGE OVERFLOW
13D8
13D8 CB74                  BIT 6,H
13DA 202B                  JR NZ,NVSINCP      ;KEEP IN 8000-BFFF REGION
13DC
13DC AE         NVSIEN:    XOR (HL)
13DD E6BF                  AND 0BFH           ;IS IT DESIRED TYPE/NAME LENGTH? IGNORE BIT 6
13DF                                          ;MISMATCH ("FOR" FLAG)
13DF 23                    INC HL             ;PT TO PTR LSB
13E0 20EF                  JR NZ,NVMLP        ;LOOP TIME=84Ts
13E2
13E2 79                    LD A,C
13E3 E61F                  AND 1FH            ;ISOLATE NAME LENGTH
13E5 E5                    PUSH HL            ;SAVE PTR TO PTR LSB IN CASE MATCH FAILS
13E6 23                    INC HL             ;
13E7 23                    INC HL             ;SKIP PTR BYTES.
13E8                                          ;HL=PTR TO NAME 2ND. LETTER IN VARS AREA.
13E8 2810                  JR Z,NVSFND        ;JR IF SINGLE LETTER VAR - ALREADY MATCHED.
13EA
13EA 47                    LD B,A
13EB DD214151              LD IX,FIRLET+1     ;PT TO SECOND LETTER
13EF
```

```
13EF DD7E00     NVMTCHLP:  LD A,(IX+0)
13F2 BE                    CP (HL)
13F3 20DB                  JR NZ,NVMOLP      ;EXIT IF MATCH FAILS
13F5
13F5 DD23                  INC IX
13F7 23                    INC HL
13F8 10F5                  DJNZ NVMTCHLP     ;LOOP TILL WHOLE NAME MATCHES. HL PTS TO VALUE.
13FA
13FA DDE1       NVSFND:    POP IX            ;PTR TO PTR LSB
13FC DD4EFF                LD C,(IX-1)       ;C=TYPE/LEN FROM VARS
13FF
13FF 3C         NZST:      INC A             ;NZ STATUS
1400 C9                    RET               ;PREV. VAR PTR MSB IS AT IX-DE-1
1401
1401
1401            ;PAGE OVERFLOW - MIGHT BE DUE TO LOTS OF VARS, OR PTR MSB=FF (TERMINATOR)
1401
1401 14         NVSPOV:    INC D
1402 C8                    RET Z             ;NOT FOUND - Z. HL HAS MOVED BACK BY 1 TO PTR LSB
1403
1403 CD9D1F                CALL PGOVERF
1406 BF                    CP A              ;Z
1407
1407 C4F23F     NVSINCP:   CALL NZ,INCURPAGE
140A
140A 79                    LD A,C
140B 18CF                  JR NVSIEN
140D
140D            ;FROM PARPRO
140D            ;ENTRY: HL POINTS TO NAME ON BSTK, A=T/L BYTE.
140D
140D 114051     LKBSV:     LD DE,FIRLET
1410 47                    LD B,A
1411 E60F                  AND 0FH
1413 4F                    LD C,A
1414 78                    LD A,B
1415 0600                  LD B,0            ;BC=NAME LEN
1417 EDB0                  LDIR              ;COPY STORED NAME FROM BSTK TO FIRLET BUFFER
1419 22C45B                LD (BSTKEND),HL
141C E66F                  AND 6FH           ;MASK BIT THAT SHOWS IF A GLOBAL VERSION WAS HIDDEN
141E                                         ;(7) - LOOK FOR "VISIBLE" FORM USED LOCALLY
141E                                         ;ALSO FORCE BIT 4 LOW
141E 1811                  JR STARYLK2     ;A=DESIRED T/L BYTE
1420
1420            ;LOOK FOR A STRING OR ARRAY
1420
1420            ;ENTRY:  NAME IS IN BUFFER AT "FIRLET"
1420            ;        C=TYPE/LEN BYTE. BIT 5 SET IF NUMERIC ARRAY, BIT 6 SET IF STRING ARRAY
1420            ;        OR SLICED STRING. (IF BIT 6 AND BIT 5=0, SIMPLE STRING NAME)
1420            ;EXIT:   Z IF VAR NOT FOUND. C=ORIG TYPE/LEN BYTE, STRLOCN/HL" PT TO STOPPER.
1420            ;       NZ IF FOUND. HL POINTS TO LEN IN PAGES, C/(DE)=TYPE/LEN IN VARS,
1420            ;       STRLOCN=TYPE/LEN IN VARS
1420            ;VARS AREA HOLDS TYPE/LEN BYTE, FIRST LET,..LAST LET, PADDED TO 10 CHARS,
1420            ;LEN IN PAGES, LEN MOD 16K, TEXT. REPEATED UNTIL FF STOPPER
1420            ;TYPE/LEN BIT 7=HIDDEN, 6=STRING ARRAY, 5=NUMERIC ARRAY (6 AND 5 LOW=SIMPLE $)
1420            ;USES ALL REGS
1420
1420 0C         STARYLK:   INC C
1421 79                    LD A,C
1422 323F51                LD (TLBYTE),A
1425 E61F                  AND 1FH           ;ISOLATE NAME LEN
1427 FE0B                  CP 11
1429 D2B214                JP NC,INVVARNM    ;ERROR IF STRING/ARRAY NAME LONGER THAN 10 CHARS
142C                                         ;EXCLUDING $ OR (
142C 7E                    LD A,(HL)
142D 17                    RLA
142E 30CF                  JR NC,NZST        ;IF SYNTAX TIME, SET NZ ("FOUND") (A REG COULD NOT
1430                                         ;HAVE HELD FFH)
1430 79                    LD A,C
1431
1431            ;CALLED BY PROCESS PARAMS
1431
1431 D9         STARYLK2:  EXX
1432 4F                    LD C,A            ;C"=DESIRED TYPE/LEN (LEN NOW TRUE NAME LEN)
1433 061F                  LD B,1FH          ;B"="LEN" MASK
1435 1EBF                  LD E,0BFH         ;E"=MASK TO FORCE BIT 6 LOW
1437 D9                    EXX
1438
1438 CD271F                CALL ADDRSAV      ;PT HL AT STR/ARRAY VARS, SWITCHED IN.
143B DBFB                  IN A,(251)        ;START WITH CURRENT PORT VALUE
143D
143D 114051     LKSTRLP:   LD DE,FIRLET      ;PT TO START OF STORED NAME.
1440 D3FB                  OUT (251),A       ;NO EFFECT ON FIRST PASS, AND MANY OTHERS
1442 22BC5B                LD (STRLOCN),HL   ;SAVE START OF CURRENT STRING/ARRAY
1445 7E                    LD A,(HL)         ;GET TYPE (BITS 7-5) AND NAME LEN (BITS 4-0)
1446 D9                    EXX
1447
1447 67                    LD H,A            ;SAVE DATA FROM VARS BRIEFLY
1448 A9                    XOR C             ;XOR WITH DESIRED
1449 A3                    AND E             ;AND RESULT WITH BFH.
144A                                         ;(SEE IF TYPE/LEN FROM VARS=DESIRED T/L IN C".
144A                                         ;IGNORE BIT 6 MISMATCH (SIMPLE VS. ARRAY STRINGS)
144A 201E                  JR NZ,TLNOMTCH
144C
144C 7C                    LD A,H
144D A0                    AND B             ;B"=1FH. ISOLATE NAME LEN. (SAME FOR DESIRED
144E D9                    EXX               ;AND CANDIDATE NAME)
144F
144F 47                    LD B,A
1450 23                    INC HL
1451 1A                    LD A,(DE)         ;FIRLET
1452 BE                    CP (HL)           ;DO A QUICK CHECK ON FIRST LETTER.
1453 2018                  JR NZ,FLNOMTCH    ;JR IF FAIL
```

```
1455                                    ;ELSE CHECK ENTIRE NAME
1455 1806                  JR DCNMLN    ;B=1-10
1457
1457 23          STMTCHLP: INC HL
1458 13                    INC DE
1459 1A                    LD A,(DE)
145A BE                    CP (HL)
145B 2010                  JR NZ,FLNOMTCH
145D
145D 10F8        DCNMLN:   DJNZ STMTCHLP
145F
145F 2ABC5B                LD HL,(STRLOCN)
1462 4E                    LD C,(HL)    ;T/L BYTE FROM VARS
1463 EB                    EX DE,HL
1464 210C00                LD HL,12
1467 2D                    DEC L        ;NZ
1468 19                    ADD HL,DE    ;HL PTS TO LEN IN PAGES, DE PTS TO T/L BYTE
1469 C9                    RET
146A
146A 24          TLNOMTCH: INC H        ;TEST FOR FF STOPPER
146B C8                    RET Z        ;RET IF IT IS. Z SHOWS NOT FOUND, C=DESIRED T/L
146C
146C D9                    EXX
146D
146D 2ABC5B      FLNOMTCH: LD HL,(STRLOCN)
1470 010B00                LD BC,11
1473 09                    ADD HL,BC    ;SKIP NAME, PT TO LEN IN PAGES
1474 DBFB                  IN A,(251)
1476 86                    ADD A,(HL)   ;ADD PAGES OF STRING/ARRAY LENGTH
1477 23                    INC HL
1478 4E                    LD C,(HL)
1479 23                    INC HL
147A 46                    LD B,(HL)
147B 23                    INC HL       ;PTS TO TEXT
147C 09                    ADD HL,BC    ;BC (0-3FFF) ADDED TO PT HL TO NEXT T/L BYTE
147D 3809                  JR C,LKSTRPO ;DEAL WITH RARE CASE OF STRLOC NEAR PAGE C END, BC
147F                                    ;NEAR 16K
147F CB74                  BIT 6,H
1481 28BA                  JR Z,LKSTRLP ;JR IF HL STILL IN 8000-BFFF REGION
1483
1483 CBB4                  RES 6,H      ;=SUB 4000H
1485
1485 3C          LKSI:     INC A        ;INC PAGE TO COMPENSATE
1486 18B5                  JR LKSTRLP
1488
1488 3C          LKSTRPO:  INC A        ;OVERFLOWED INTO 0000-3FFF - CORRECT BY 2 PAGES
1489 CBFC                  SET 7,H      ;8000-BFFF AGAIN
148B 18F8                  JR LKSI
148D
148D             ;NAME TO BUFFER
148D             ;ALTERS HL, DE, BC, A
148D             ;ENTRY: HL PTS TO A CHAR, ALSO HELD IN A
148D             ;EXIT:  DE POINTS TO CHAR PAST NAME
148D             ;       NAME, COMPRISING ALPHANUMERICS OR UNDERLINE CHARS, COPIED TO BASE PAGE
148D             ;       BUFFER AT "FIRLET", WITH SPACES REMOVED AND LOWER CASE FORCED.
148D             ;       HL=FLAGS ADDR
148D             ;       C BITS 4-0=NAME LEN, EXCLUDING FIRST CHAR. (0-31)
148D             ;       BIT 6,C SET IF STRING ARRAY OR SLICED STRING
148D             ;       BIT 5,C SET IF NUMERIC ARRAY
148D             ;       BIT 6,(FLAGS) SET IF NUMERIC, RES IF STRING
148D
148D 0620        NAMTOBUF: LD B,32      ;MAX LEN+1 FOR A NAME (EXCLUDING 1ST CHAR)
148F 114051                LD DE,FIRLET ;PT TO NAME BUFFER IN COMMON MEMORY
1492 CDEC3A                CALL GETALPH
1495 F620                  OR 20H
1497 12                    LD (DE),A    ;FIRST LETTER STORED
1498
1498 23          NMTBL:    INC HL
1499 7E                    LD A,(HL)
149A FE20                  CP 20H
149C 28FA                  JR Z,NMTBL   ;SKIP SPACES
149E
149E CD2C3B                CALL ALPHANUM ;SEE IF LETTER, NUMBER OR UNDERLINE
14A1 3806                  JR C,NMTB2   ;JR IF ALPANUMERIC
14A3
14A3 FE5F                  CP "_"
14A5 200D                  JR NZ,NAMEND
14A7
14A7 1802                  JR NMTB3
14A9
14A9 F620        NMTB2:    OR 20H       ;FORCE LOWER CASE (NUMS UNAFFECTED)
14AB
14AB 13          NMTB3:    INC DE
14AC 12                    LD (DE),A
14AD 10E9                  DJNZ NMTBL   ;LOOP TILL LENGTH ILLEGAL
14AF
14AF 22975A                LD (CHAD),HL ;(FOR XPTR)
14B2
14B2 CF          INVVARNM: RST 08H
14B3 28                    DB 40        ;"Invalid variable name" - USUALLY SPOTTED IN
14B4                                    ;SYNTAX CHECK
14B4
14B4             ;NAME PROPER HAS NOW ENDED, BUT TYPE CHARACTER MAY FOLLOW
14B4
14B4 EB          NAMEND:   EX DE,HL
14B5 213B5C                LD HL,FLAGS
14B8 3E20                  LD A,32
14BA 90                    SUB B
14BB 4F                    LD C,A       ;C=NAME LENGTH-1, RANGE 0-31.
14BC 1A                    LD A,(DE)
14BD FE24                  CP "$"
14BF 200B                  JR NZ,NMEN2  ;JR IF NUMERIC VAR
14C1
```

```
14C1 CBB6                  RES 6,(HL)        ;"STRING"
14C3 13                    INC DE
14C4 1A                    LD A,(DE)
14C5 FE28                  CP "("
14C7 C0                    RET NZ
14C8
14C8 13                    INC DE
14C9 CBF1                  SET 6,C           ;"STRING ARRAY"
14CB C9                    RET
14CC
14CC CBF6        NMEN2:    SET 6,(HL)        ;"NUMERIC"
14CE FE28                  CP "("
14D0 C0                    RET NZ
14D1
14D1 13                    INC DE
14D2 CBE9                  SET 5,C           ;"NUMERIC ARRAY"
14D4 C9                    RET
14D5
14D5 3E20                  LD A,32
14D7 90                    SUB B
14D8 4F                    LD C,A            ;C=NAME LENGTH-1, RANGE 0-31.
14D9 C9                    RET
14DA
14DA CDAA13      LVFLAGS:  CALL LOOKVARS
14DD 08                    EX AF,AF'         ;Z IF NOT FOUND
14DE 3A3B5C                LD A,(FLAGS)
14E1 87                    ADD A,A
14E2 C9                    RET               ;M IF NUM, P IF $, CY IF RUNNING
14E3                       INCLUDE EVAL.SAM      ;ROM 0 FNS
14E3             ;EVAL.SAM - SAM EXPRESSION EVALUATOR (ROM0). EXITS WITH PAGING UNCHANGED.
14E3             ;ENTRY: CHAD PTS TO FIRST CHAR. EXIT: RESULT ON FPCS IF RUNNING, ELSE SYNTAX
14E3             ;CHECKED AND 5-BYTES INSERTED IF NEEDED. BIT 6,(FLAGS)=0 IF STRING RESULT,
14E3             ;ELSE NUMERIC. CHAD PTS TO CHAR THAT CANNOT BE PART OF EXPR. A=CHAR, HL=CHAD
14E3             ;*************************************************************************
14E3             ;
14E3 CD0102      SCANNING: CALL R1OFFCL
14E6 ED14                  DW SCANSR
14E8 4F                    LD C,A            ;CURRENT CHAR
14E9 3A3B5C                LD A,(FLAGS)
14EC C9                    RET
14ED
14ED 1600        SCANSR:   LD D,0            ;PRIORITY "STOPPER"
14EF DF                    RST 18H           ;GET FIRST CHAR. HL=CHAD
14F0 FE                    DB 0FEH           ;"JR+1"
14F1
14F1 E7          SCANPLP:  RST 20H           ;NEXT CHAR
14F2
14F2 D5                    PUSH DE           ;PRIORITY/CODE
14F3
14F3             ;FROM UNARY PLUS WITH HL=CHAD
14F3
14F3 5F          SCANLP:   LD E,A
14F4 E6DF                  AND 0DFH          ;LETTERS BECOME UPPER CASE
14F6
14F6 FE5B                  CP 5BH            ;"Z"+1
14F8 303D                  JR NC,ABOVLETS
14FA
14FA FE41                  CP "A"
14FC DA9915                JP C,BELOWLETS ;JR UNLESS WE HAVE A LETTER
14FF
14FF             ;EVALUATE A VARIABLE
14FF
14FF 3A555C      SLETTER:  LD A,(DEFADD+1)   ;PTS PAST DEF FN "(" IF FN BEING EVALED.
1502 A7                    AND A
1503 2026                  JR NZ,SLLKFV      ;CHECK DEF FN BRACKETS FOR VAR IF DEFADD-HI NZ
1505
1505 7E          SLET1:    LD A,(HL)
1506 CDAB13                CALL LKVARS2
1509 CA082D                JP Z,VNFERR    ;ERROR IF NOT FOUND
150C
150C 3A3B5C                LD A,(FLAGS)
150F 87                    ADD A,A           ;CY IF RUNNING
1510 F21715                JP P,SLET2     ;JP IF STRING
1513
1513 CB69                  BIT 5,C
1515 280A                  JR Z,SLET3     ;JR IF NOT (NUMERIC) ARRAY
1517
1517 CD542D      SLET2:    CALL STKVAR2   ;IF STRING, STACK START AND LEN, IF N ARRAY
151A                                      ;GET HL=START ADDR OF ELEMENT
151A 3A3B5C                LD A,(FLAGS)
151D 87                    ADD A,A
151E
151E F2F115                JP P,SCONT1    ;JR IF STRING - CHECK FOR SLICER, THEN OPERATOR
1521
1521 DC201D      SLET3:    CALL C,HLTOFPCS ;STACK NUM IF RUNNING
1524
1524 CDD73F      SLET4:    CALL SELCHADP
1527 DF                    RST 18H
1528 C3F915                JP OPERATOR    ;AN OPERATOR OR A TERMINATOR MUST FOLLOW
152B
152B CD6E33      SLLKFV:   CALL LKFNVAR
152E 30D5                  JR NC,SLET1    ;JR IF NOT FOUND OR SYNTAX TIME
1530
1530 FE24                  CP "$"
1532 CAF415                JP Z,SCONT2
1535
1535 18ED                  JR SLET4
1537
1537             ;*************************************************************************
1537
1537 1C          ABOVLETS: INC E
1538 2069                  JR NZ,EVNONSE  ;ERROR UNLESS FF FUNCT LEADER FOUND
153A
```

```
153A 23                     INC HL
153B 7E                     LD A,(HL)       ;GET FUNCTION CODE
153C D61A                   SUB 1AH         ;ADJUST 3B-83H TO 21H-69H
153E 5F                     LD E,A
153F 22975A                 LD (CHAD),HL    ;SKIP "FF"
1542
1542 2AF65A                 LD HL,(EVALUV)
1545 24                     INC H
1546 25                     DEC H
1547 C40500                 CALL NZ,HLJUMP ;IF VECTORED JUMP WITH A=FN CODE.
154A
154A FE39                   CP SIN
154C 3825                   JR C,IMMEDCODES
154E
154E 16CF                   LD D,0CFH       ;"PRIORITY 0F, N ARG, N RESULT"
1550 FE4F                   CP EOF+3
1552 389D                   JR C,SCANPLP    ;JR IF SIN-EOF/PTR/POS
1554
1554 FE5D                   CP NOT+1
1556 304B                   JR NC,EVNONSE   ;RANGE NOW UDG-NOT
1558
1558 1600       SCANUMEN:   LD D,0
155A 214716                 LD HL,FNPRIORT-UDGA
155D 19                     ADD HL,DE
155E 56                     LD D,(HL)       ;FETCH PRIORITY AND INPUT/OUTPUT TYPE. (BIT 7=1/0
155F                                        ;FOR N/$ RESULT, BIT 6=DITTO FOR ARGUMENT)
155F 1890                   JR SCANPLP      ;STACK DE, GET NEXT CHAR, LOOP
1561
1561            ;*********************************************************************************
1561            ;INKEY$ - THIS IS MAIN (NON-FPC) ROUTINE
1561
1561 E7         IMINKEYS:   RST 20H         ;SKIP "INKEY"
1562 FE23                   CP "#"
1564 1E5B                   LD E,INKEY      ;FPC INKEY$ CODE
1566 C1                     POP BC          ;RET ADDR (STRCONT)
1567 28EF                   JR Z,SCANUMEN   ;STREAM VERSION HANDLED BY FPC. "#" WILL BE SKIPPED
1569
1569 C5                     PUSH BC
156A CD153B                 CALL ABORTER    ;(WE CANNOT USE THE NORMAL INPUT STREAM FOR INKEY$
156D                                        ;ON ITS OWN, SINCE THAT WOULD NOT CAUSE A KEYSCAN,
156D                                        ;BUT WOULD JUST FETCH LAST-K
156D
156D CDB11C                 CALL READKEY    ;RETURNS KEY CODE IN A AND CY IF OK
1570 F7                     RST 30H
1571 CE4F                   DW FPINKEN-8000H  ;MAKE 1-CHAR STRING AND STACK PARAMS
1573
1573            ;*********************************************************************************
1573            ;SOME FUNCTIONS HAVE TO BE HANDLED AT ONCE BECAUSE THEY HAVE NO ARGUMENTS
1573            ;(LIKE PI) OR BECAUSE THEY HAVE SEVERAL ARGUMENTS IN BRACKETS (LIKE POINT).
1573
1573            ;NUM. RESULT: PI, RND, POINT, FREE, LENGTH, ITEM, ATTR, FN, BIN, HIMEM, XMOUSE,
1573            ;YMOUSE, XPEN, YPEN, INARRAY, INSTR.
1573
1573            ;STR. RESULT: INKEY$, SCREEN$, MEMORY$, CHAR$, PATH$, STRING$, USING$, SHIFT$
1573
1573
1573 D621       IMMEDCODES: SUB PI
1575 382C                   JR C,EVNONSE    ;RANGE NOW PI-SHIFT$
1577
1577 87                     ADD A,A
1578 5F                     LD E,A
1579 1600                   LD D,0
157B 21A516                 LD HL,IMFNATAB
157E 19                     ADD HL,DE
157F 4E                     LD C,(HL)
1580 23                     INC HL
1581 46                     LD B,(HL)
1582 21E915                 LD HL,NUMCONT
1585 FE1F                   CP 0+(INSTR-PI)*2+1
1587 3803                   JR C,IMMEDNUM
1589
1589 219415                 LD HL,STRCONT
158C
158C E5         IMMEDNUM:   PUSH HL
158D CB78                   BIT 7,B
158F C2EA01                 JP NZ,R1ONCLBC
1592
1592 C5                     PUSH BC
1593 C9                     RET             ;JP TO BC
1594
1594 CD3E2E     STRCONT:    CALL SLLPEX     ;SIGNAL STRING RESULT
1597 185B                   JR SCONT2
1599
1599            ;*********************************************************************************
1599
1599 7B         BELOWLETS:  LD A,E
159A FE30                   CP "0"
159C DAD516                 JP C,BELOWNUM
159F
159F FE3A                   CP 3AH
15A1 3804                   JR C,SDECIMAL
15A3
15A3 CF         EVNONSE:    RST 08H
15A4 1D                     DB 29           ;"NONSENSE"
15A5
15A5            ;*********************************************************************************
15A5            ;HANDLE LITERAL NUMBER (0-9, DECIMAL PT., BIN OR AMPERSAND
15A5
15A5 F1         IMBIN:      POP AF          ;JUNK RET TO IMMEDNUM+3
15A6 DF                     RST 18H         ;GET HL=CHAD
15A7
15A7 3A3B5C     SDECIMAL:   LD A,(FLAGS)
15AA 17                     RLA
```

```
15AB 3019                  JR NC,INSERT5B ;JR AND INSERT THE INVISIBLE FORM IF SYNTAX TIME
15AD
15AD 23          LK0ELP:   INC HL
15AE 7E                    LD A,(HL)
15AF FE0E                  CP 0EH
15B1 20FA                  JR NZ,LK0ELP ;LOOP TILL NUMBER MARKER FND.
15B3
15B3 23                    INC HL
15B4 010500                LD BC,5
15B7 ED5B655C              LD DE,(STKEND)
15BB EDB0                  LDIR           ;COPY NUMBER TO FPCS
15BD ED53655C              LD (STKEND),DE
15C1
15C1 22975A      SCHADNUM: LD (CHAD),HL
15C4 1823                  JR NUMCONT
15C6
15C6 CD5517      INSERT5B: CALL CALC5BY   ;FIND 5-BYTE FORM OF DEC, HEX OR BIN NUMBER
15C9 2A975A                LD HL,(CHAD)
15CC CD6433                CALL MAKESIX   ;MAKE 6 SPACES IN ELINE, PLACE 0EH, INC HL
15CF EB                    EX DE,HL
15D0 CD121D                CALL FDELETE
15D3 010500                LD BC,5        ;HL PTS TO DELETED NUMBER, DE TO ROOM IN ELINE
15D6 EDB0                  LDIR           ;COPY 5 BYTES TO ELINE
15D8 EB                    EX DE,HL
15D9 18E6                  JR SCHADNUM    ;SET NUMERIC, SET CHAD
15DB
15DB             ;*************************************************************************
15DB
15DB 213B5C      SSLICER:  LD HL,FLAGS
15DE CB76                  BIT 6,(HL)
15E0 204D                  JR NZ,SLOOP    ;NUMBERS CANNOT HAVE A SLICER - TERMINATE
15E2
15E2 E7                    RST 20H        ;SKIP "("
15E3 CD692E                CALL SLICING
15E6 E7                    RST 20H
15E7 180C                  JR SLSTRLP
15E9
15E9             ;*************************************************************************
15E9
15E9 213B5C      NUMCONT:  LD HL,FLAGS
15EC CBF6                  SET 6,(HL)     ;SIGNAL "NUMERIC"
15EE DF                    RST 18H
15EF 1808                  JR OPERATOR
15F1
15F1 CDD73F      SCONT1:   CALL SELCHADP
15F4
15F4 DF          SCONT2:   RST 18H        ;STRING EXPRESSIONS CAN BE FOLLOWED BY A SLICER
15F5                                      ;EG (STR$ 123)(2) SO CHECK FOR THIS BEFORE LOOKING
15F5                                      ;FOR AN OPERATOR OR TERMINATOR
15F5 FE28        SLSTRLP:  CP "("
15F7 28E2                  JR Z,SSLICER
15F9
15F9             ;DEAL WITH BINARY OPERATORS: +,-,*,/,^,=,>,<,<=,>=,<>,OR,AND,MOD,IDIV,BOR, ETC.
15F9
15F9 1600        OPERATOR: LD D,0          ;PRIORITY=0
15FB 3C                    INC A
15FC 2011                  JR NZ,OPERAT2  ;JR IF NOT FF FN CODE LEADER
15FE
15FE 23                    INC HL
15FF 7E                    LD A,(HL)      ;FETCH OPERATOR CODE
1600 D67A                  SUB MODTOK
1602 382B                  JR C,SLOOP
1604
1604 FE0A                  CP 0AH
1606 3027                  JR NC,SLOOP    ;RANGE NOW 00-09 FOR MOD TO ">="
1608
1608 22975A                LD (CHAD),HL   ;SKIP FF
160B
160B C608                  ADD A,8        ;MOD TO ">=" BECOME 08-11H
160D 181A                  JR OPERAT3
160F
160F D62B        OPERAT2:  SUB "*"+1      ;MULT IS 0, DIVN IS 5, "<" IS 12H, ">" IS 14H
1611 381C                  JR C,SLOOP     ;JR IF BELOW BINARY OPERATOR RANGE
1613
1613 FE04                  CP 4
1615 2818                  JR Z,SLOOP     ;EXCLUDE "."
1617
1617 FE06                  CP 6
1619 380E                  JR C,OPERAT3   ;JR IF WITHIN RANGE FOR * TO /
161B
161B FE12                  CP 12H
161D 3810                  JR C,SLOOP
161F
161F FE15                  CP 15H
1621 3806                  JR C,OPERAT3   ;JR IF "<", "=" OR ">" (NOW 12H-14H)
1623
1623 FE34                  CP 34H         ;POWER-OF
1625 3E04                  LD A,4
1627 2006                  JR NZ,SLOOP
1629
1629 5F          OPERAT3:  LD E,A         ;E=BIN OPERATOR CODE 00-14H
162A 218116                LD HL,OPPRIORT
162D 19                    ADD HL,DE
162E 56                    LD D,(HL)      ;FETCH OPERATOR PRIORITY, OR 0 IF NOT RECOGNISED
162F
162F C1          SLOOP:    POP BC         ;PREV PRIORITY (B) AND OPERATION CODE (C)
1630 78                    LD A,B
1631 92                    SUB D
1632 E610                  AND 10H        ;IF PRIORITY NIBBLE IS HIGHER IN D, BIT 4 WILL BE 1
1634 2028                  JR NZ,PRIGRTR  ;JR IF CURRENT PRIORITY HIGHER - WAIT
1636
1636 B0                    OR B           ;ELSE B PRIORITY IS >=D
1637 CA1800                JP Z,0018H     ;EXIT IF BOTH PRIORITIES ARE ZERO
```

```
163A
163A D5                     PUSH DE         ;CURRENT PRIORITY/CODE
163B 213B5C                 LD HL,FLAGS
163E 7E                     LD A,(HL)
163F 17                     RLA             ;CY IF RUNNING
1640 3808                   JR C,EXECOP     ;PERFORM OPERATION IF RUNNING
1642
1642 78                     LD A,B          ;PRIORITY CODE OF FN TO BE CHECKED/EXECUTED
1643 AE                     XOR (HL)        ;CHECK THAT FNS THAT WORK ON STRINGS HAVE STRING
1644                                        ;"LAST VALUE"S, AND DITTO FOR NUMERICS. BIT 6 OF
1644                                        ;PRIORITY CODE IS 0 IF FN WORKS ON STRINGS, ELSE 1
1644 87                     ADD A,A
1645 FAA315                 JP M,EVNONSE    ;ERROR IF BIT 6,(FLAGS)<>BIT 6 OF FN CODE
1648
1648 1809                   JR CHKEXECC
164A
164A C5          EXECOP:    PUSH BC
164B 41                     LD B,C          ;FN CODE TO EXECUTE
164C
164C EF                     DB CALC
164D 24                     DB USEB
164E 33                     DB EXIT
164F
164F C1                     POP BC
1650 213B5C                 LD HL,FLAGS
1653
1653 78          CHKEXECC:  LD A,B          ;PRIORITY CODE OF FN JUST CHECKED/EXECUTED
1654 D1                     POP DE          ;NEXT PRIORITY/CODE
1655 CBF6                   SET 6,(HL)      ;"LAST VALUE IS NUMERIC"
1657 17                     RLA             ;BIT 7 OF THE P. CODE IS SET IF ITS RESULT IS A NUM
1658 38D5                   JR C,SLOOP      ;JR IF WE SET FLAGS CORRECTLY
165A
165A CBB6                   RES 6,(HL)      ;"LAST VALUE IS STRING"
165C 18D1                   JR SLOOP
165E
165E             ;*********************************************************************************
165E
165E C5          PRIGRTR:   PUSH BC         ;PREV PRIOR/CODE
165F 3A3B5C                 LD A,(FLAGS)
1662 87                     ADD A,A
1663 FAF114                 JP M,SCANPLP    ;LOOP IF NUMERIC - I.E. LEAVE TYPE CODE BITS
1666                                        ;ALONE IF A NUMERIC IS FOLLOWED BY A BINARY
1666                                        ;OPERATOR - THEY ARE SET FOR NUMERIC I/P AND O/P
1666
1666 7B                     LD A,E          ;CURRENT CODE
1667                                        ;ALL LEGAL $ BINARY OPS EXCEPT "+" WILL GET
1667                                        ;CODES 7 ABOVE THEIR NUMERIC EQUIVALENTS
1667
1667 CBBA                   RES 7,D         ;RESULT TYPE WILL BE STRING FOR "+" AND "AND"
1669 FE0E                   CP 0EH          ;CP "AND"
166B 280E                   JR Z,SCANPLPH   ;IF "AND" LEAVE "INPUT" BIT AS NUMERIC FOR
166D                                        ;"$ AND N"
166D CBB2                   RES 6,D         ;ELSE "INPUT" BIT IS STRING E.G. "$+$", "$<$"
166F 1C                     INC E           ;IF "+", E=2
1670 FE01                   CP 1            ;CP "+"
1672 280A                   JR Z,SCANPH2    ;JP WITH STRING INPUT AND OUTPUT FOR "+"
1674
1674 CBFA                   SET 7,D         ;NUMERIC OUTPUT FOR E.G. "$>$"
1676 FE0E                   CP 0EH          ;CP "AND"
1678 DA290D                 JP C,NONSENSE   ;E.G. $ MOD $ IS AN ERROR
167B
167B C607        SCANPLPH:  ADD A,7
167D 5F                     LD E,A
167E
167E C3F114      SCANPH2:   JP SCANPLP      ;JP TO STACK DE AND GET NEXT CHAR
1681
1681             ;*********************************************************************************
1681             ;BINARY OPERATOR PRIORITY TABLE
1681
1681 C8          OPPRIORT:  DB 0C8H  ;0  2A *
1682 C6                     DB 0C6H  ;1  2B +
1683 00                     DB 0     ;2  2C
1684 C6                     DB 0C6H  ;3  2D -
1685 CF                     DB 0CFH  ;4  5E TO-POWER-OF
1686 C8                     DB 0C8H  ;5  2F /
1687 00                     DB 0     ;6
1688 00                     DB 0     ;7
1689 CE                     DB 0CEH  ;8     MOD
168A CE                     DB 0CEH  ;9     IDIV
168B C2                     DB 0C2H  ;A     BOR
168C C2                     DB 0C2H  ;B     BXOR
168D C3                     DB 0C3H  ;C     BAND
168E C2                     DB 0C2H  ;D  .. OR
168F
168F C3                     DB 0C3H  ;E  .. AND  (ADD 7 FOR $ EQUIV. OF "AND" TO ">")
1690 C5                     DB 0C5H  ;F  .. <>
1691 C5                     DB 0C5H  ;10 .. <=
1692 C5                     DB 0C5H  ;11 .. >=
1693 C5                     DB 0C5H  ;12 3C <
1694 C5                     DB 0C5H  ;13 3D =
1695 C5                     DB 0C5H  ;14 3E >
1696
1696             ;*********************************************************************************
1696             ;UNARY FUNCTION PRIORITY TABLE (FOR MINORITY OF FNS THAT AREN"T PRI. 16,N,N)
1696             ;BIT 7=1 IF NUM RESULT. BIT 6=1 IF NUM ARG. BITS 4-0=PRIORITY
1696
1696 8F          FNPRIORT:  DB  8FH         ;UDG
1697 8F                     DB  8FH         ;NUMBER
1698 8F                     DB  8FH         ;LEN
1699 8F                     DB  8FH         ;CODE
169A 0F                     DB  0FH         ;VAL$
169B 8F                     DB  8FH         ;VAL
169C 0F                     DB  0FH         ;TRUNC$
```

```
169D 4F                     DB  4FH         ;CHR$
169E 4F                     DB  4FH         ;STR$
169F 4F                     DB  4FH         ;BIN$
16A0 4F                     DB  4FH         ;HEX$
16A1 4F                     DB  4FH         ;USR$
16A2 4F                     DB  4FH         ;INKEY$
16A3
16A3 C4                     DB 0C4H         ;NOT
16A4 C9                     DB 0C9H         ;NEGATE
16A5
16A5            ;*********************************************************************************
16A5            ;IMMEDIATE FN ADDRESS TABLE. ALL ARE EVALUATED AT ONCE BECAUSE THEY HAVE NO
16A5            ;ARGS, OR BRACKETED ARGS, OR #ARG.
16A5
16A5 A718       IMFNATAB:   DW IMPI         ;NUMERIC RESULT
16A7 F6CE                   DW IMRND
16A9 5DCF                   DW IMPOINT
16AB 3618                   DW IMMEM
16AD 16E9                   DW IMLENGTH
16AF AF18                   DW IMITEM
16B1 2ECF                   DW IMATTR
16B3 2B31                   DW IMFN
16B5 A515                   DW IMBIN
16B7 5018                   DW IMMOUSEX
16B9 5918                   DW IMMOUSEY
16BB 6118                   DW IMPENX
16BD 6F18                   DW IMPENY
16BF 2518                   DW IMHIMEM
16C1 290D                   DW NONSENSE
16C3 95E8                   DW IMINSTR
16C5
16C5 6115                   DW IMINKEYS     ;STRING RESULT
16C7 14F2                   DW IMSCREENS
16C9 1318                   DW IMMEMRYS
16CB 290D                   DW NONSENSE
16CD EFCE                   DW IMPATHS
16CF 8EE9                   DW IMSTRINGS
16D1 290D                   DW NONSENSE
16D3 290D                   DW NONSENSE
16D5
16D5            ;*********************************************************************************
16D5            ;BELOW NUMBERS ARE QUOTE, "&", OPEN BRACKET, UNARY PLUS & MINUS, DECIMAL PT.
16D5            ;(22H,26H,28H,2BH,2DH,2EH)
16D5
16D5 FE22       BELOWNUM:   CP 22H
16D7 282A                   JR Z,SQUOTE
16D9
16D9 FE26                   CP "&"
16DB 280A                   JR Z,SDECIMALH
16DD
16DD FE28                   CP "("
16DF 2817                   JR Z,SBRACKET
16E1
16E1 FE2D                   CP "-"
16E3 280E                   JR Z,UNARMIN
16E5
16E5 FE2E                   CP "."
16E7
16E7 CAA715     SDECIMALH:  JP Z,SDECIMAL
16EA
16EA FE2B                   CP "+"
16EC C2290D                 JP NZ,NONSENSE
16EF
16EF E7         UNARPLU:    RST  20H        ;JUST SKIP A UNARY PLUS
16F0 C3F314                 JP SCANLP
16F3
16F3 1E5D       UNARMIN:    LD E,NEGATE     ;UNARY MINUS CODE
16F5 C35815                 JP SCANUMEN
16F8
16F8 E7         SBRACKET:   RST  20H        ;SKIP "("
16F9 CDE314                 CALL SCANNING
16FC 79                     LD A,C
16FD CD953A                 CALL INSISCBRK ;INSIST ON ")"
1700 C3F415                 JP SCONT2
1703
1703            ;*********************************************************************************
1703            ;PASS PARAMS OF STRING LITERAL TO FPCS IN RUN TIME
1703
1703 23         SQUOTE:     INC HL          ;SKIP QUOTE
1704 E5                     PUSH HL         ;STRING TEXT START
1705 3A3B5C                 LD A,(FLAGS)
1708 17                     RLA
1709 08                     EX AF,AF'       ;CY IN F" IF RUNNING
170A 01FFFF                 LD BC,0FFFFH    ;INITIALISE LENGTH
170D
170D 7E         QUTSRLP:    LD A,(HL)
170E 23                     INC HL
170F 03                     INC BC          ;INC LEN COUNT
1710 FE0D                   CP 0DH
1712 CA290D                 JP Z,NONSENSE
1715
1715 FE22                   CP 22H
1717 20F4                   JR NZ,QUTSRLP  ;LOOP UNTIL QUOTE FOUND
1719
1719 D1                     POP DE          ;START
171A 7E                     LD A,(HL)
171B FE22                   CP 22H          ;CHECK IF DOUBLE QUOTE
171D
171D 280A                   JR Z,SQUOTE2    ;JR IF EMBEDDED QUOTES USED. ELSE STRING IS
171F                                        ;SIMPLE AND CAN STAY IN BASIC LINE.
171F 22975A                 LD (CHAD),HL    ;PT PAST CLOSING QUOTE
1722 08                     EX AF,AF
1723 DCE71C                 CALL C,STKSTOREP ;STACK PARAMS OF STR IN BASIC LINE, IF RUNNING
```

```
1726
1726 C39415     STRCONTH:     JP STRCONT
1729
1729            ;EMBEDDED QUOTES - HAVE TO COPY STRING TO BUFFER, OMITTING SOME ALT. QUOTES
1729
1729 21004F     SQUOTE2:      LD HL,INSTBUF  ;ALLOWS 256 BYTES
172C 0E00                     LD C,0
172E E5                       PUSH HL        ;BUFFER START
172F
172F 1A         SQUCOPY:      LD A,(DE)      ;CHAR FROM BASIC LINE
1730 13                       INC DE
1731 FE22                     CP 22H
1733 280C                     JR Z,SQUCO3
1735
1735 47         SQUCO1:       LD B,A
1736 08                       EX AF,AF'
1737 3002                     JR NC,SQUCO2
1739
1739 70                       LD (HL),B
173A 23                       INC HL
173B
173B 08         SQUCO2:       EX AF,AF'
173C 0C                       INC C
173D 20F0                     JR NZ,SQUCOPY  ;LOOP, COPYING CHARS FROM BASIC LINE TO BUFFER
173F                                         ;(MAX OF 255)
173F CF                       RST 08H
1740 2A                       DB 42          ;"String too long"
1741
1741 1A         SQUCO3:       LD A,(DE)
1742 13                       INC DE
1743 FE22                     CP 22H
1745 28EE                     JR Z,SQUCO1    ;COPY IN NEXT CHAR IF IT IS A SECOND QUOTE MARK
1747
1747 0600                     LD B,0
1749 1B                       DEC DE         ;PT TO JUST PAST FINAL QUOTE
174A ED53975A                 LD (CHAD),DE
174E E1                       POP HL         ;BUFFER START
174F 08                       EX AF,AF'
1750 DC673C                   CALL C,CWKSTK  ;IF RUNNING, COPY TO WKSPACE, STACK PARAMS.
1753 18D1                     JR STRCONTH
1755
1755            ;CALC5BY.SAM
1755            ;*****************************************************************************
1755            ;CALCULATE A 5-BYTE FORM FOR A DECIMAL, HEX OR BINARY NUMBER
1755            ;ENTRY WITH HL AND CHAD PTING TO FIRST CHAR OF NUMBER (&,.,0-9,BIN)
1755            ;EXIT WITH VALUE ON FPCS
1755
1755 7E         CALC5BY:      LD A,(HL)
1756 FE26                     CP "&"
1758 2003                     JR NZ,NAMP
175A
175A F7                       RST 30H
175B C550                     DW AMPERSAND-8000H
175D
175D FE43       NAMP:         CP BINTOK
175F C27817                   JP NZ,DECIMAL
1762
1762 010000                   LD BC,0        ;INITIALISE RESULT
1765
1765 E7         NXBINDIG:     RST 20H        ;SKIP BIN
1766 FE30                     CP "0"
1768 2806                     JR Z,BINDIG
176A
176A FE31                     CP "1"
176C 37                       SCF
176D C2DD1C                   JP NZ,STACKBC  ;STACK RESULT AS SOON AS NON-1, NON-0 FOUND
1770
1770 CB11       BINDIG:       RL C
1772 CB10                     RL B
1774 30EF                     JR NC,NXBINDIG
1776
1776 CF                       RST 08H
1777 1C                       DB 28          ;"Number too large"
1778
1778            ;*****************************************************************************
1778            ;HANDLE EG 0.123, .123, 1.234, 1E4, 1.23E+4, 7.89E-32, 1.E5
1778
1778 FE2E       DECIMAL:      CP "."
177A 200C                     JR NZ,DECINT
177C
177C E7                       RST 20H        ;SKIP "."
177D CD303B                   CALL NUMERIC
1780 D2290D                   JP NC,NONSENSE ;INSIST ON E.G. .1 OR .8
1783
1783 EF                       DB CALC
1784 E1                       DB STKZERO     ;INTEGER PART OF A FRACTION IS ZERO
1785 33                       DB EXIT
1786
1786 180D                     JR CONVFRAC
1788
1788 CDDE17     DECINT:       CALL INTTOFP
178B FE2E                     CP "."
178D 201F                     JR NZ,EFORMAT
178F
178F E7                       RST 20H
1790 CD303B                   CALL NUMERIC
1793 3019                     JR NC,EFORMAT  ;JR IF NOT A DIGIT
1795
1795 EF         CONVFRAC:     DB CALC
1796 E6                       DB STKFONE
1797 C8                       DB STOD0       ;MULTIPLIER (M) STARTS AT 1
1798 33                       DB EXIT
```

```
1799
1799 DF                         RST 18H
179A 180D                       JR CONVFRAC2
179C
179C D630       CONVFRALP:      SUB 30H
179E 47                         LD B,A
179F
179F EF                         DB CALC
17A0 23                         DB STKBREG
17A1 D8                         DB RCL0
17A2 EC                         DB STKTEN
17A3 05                         DB DIVN
17A4 D0                         DB STO0
17A5 00                         DB MULT
17A6 01                         DB ADDN
17A7 33                         DB EXIT
17A8
17A8 E7                         RST 20H
17A9
17A9 CD303B     CONVFRAC2:      CALL NUMERIC
17AC 38EE                       JR C,CONVFRALP ;JR IF A DIGIT
17AE
17AE E6DF       EFORMAT:        AND 0DFH
17B0 FE45                       CP "E"
17B2 C0                         RET NZ
17B3
17B3 E7                         RST 20H        ;SKIP "E"
17B4 0E2B                       LD C,"+"
17B6 B9                         CP C
17B7 2805                       JR Z,GEXSGN1
17B9
17B9 FE2D                       CP "-"
17BB 2002                       JR NZ,GEXSGN2
17BD
17BD 4F                         LD C,A
17BE
17BE E7         GEXSGN1:        RST 20H        ;SKIP +/-
17BF
17BF CD303B     GEXSGN2:        CALL NUMERIC
17C2 D2290D                     JP NC,NONSENSE ;INSIST ON NUMERIC NOW
17C5
17C5 C5                         PUSH BC        ;C=+/-
17C6 CDDE17                     CALL INTTOFP
17C9 CD651D                     CALL FPTOA
17CC 3803                       JR C,NTLERR    ;JR IF >255
17CE
17CE 07                         RLCA
17CF 3002                       JR NC,GEXSGN3   ;JR IF <=127
17D1
17D1 CF         NTLERR:         RST 08H
17D2 1C                         DB 28          ;"Number too large"
17D3
17D3 0F         GEXSGN3:        RRCA
17D4 C1                         POP BC
17D5 CB49                       BIT 1,C
17D7 2002                       JR NZ,POFTENH  ;JR IF 0010 1011 (+)
17D9
17D9 ED44                       NEG
17DB
17DB F7         POFTENH:        RST 30H
17DC D45B                       DW POFTEN-8000H ;MULT FPC LAST VALUE BY E+/-A REGISTER
17DE
17DE            ;*********************************************************************
17DE            ;GET VALUE OF ASCII INTEGER IN A AND (CHAD+1...) TO FPCS. ZERO IF NO DIGITS.
17DE            ;EXIT WITH NC AND A=NON-NUMERIC CHAR
17DE
17DE 47         INTTOFP:        LD B,A
17DF
17DF EF                         DB CALC
17E0 E1                         DB STKZERO     ;TOTAL=0
17E1 33                         DB EXIT
17E2
17E2 78                         LD A,B
17E3 180C                       JR INTTOFP3
17E5
17E5 D630       INTTOFPLP:      SUB 30H
17E7 47                         LD B,A
17E8
17E8 EF                         DB CALC        ;B GOES TO BREG
17E9 EC                         DB STKTEN
17EA 00                         DB MULT        ;TOTAL=TOTAL*10
17EB 23                         DB STKBREG
17EC 01                         DB ADDN
17ED 33                         DB EXIT
17EE
17EE CDDD00                     CALL NXCHAR    ;0074 EQU - NEXT CHAR, DON"T SKIP ANYTHING
17F1
17F1 CD303B     INTTOFP3:       CALL NUMERIC
17F4 38EF                       JR C,INTTOFPLP ;LOOP WHILE NUMERIC ASCII FOUND
17F6
17F6 C9                         RET
17F7            ;*********************************************************************
17F7
17F7            ;USR$. E.G. LET A$=USR$ 12345 OR &18000
17F7
17F7 21EB1C     R0USRS:     LD HL,STKSTOS      ;STACKS DE,BC,A
17FA FD                     DB 0FDH            ;"JR+3"
17FB
17FB            ;USR. E.G. LET X=USR 123456
17FB
17FB 21DD1C     R0USR:      LD HL,STACKBC
17FE
17FE E5         USRCOM:     PUSH HL
```

```
17FF
17FF            ;FROM "CALL"
17FF
17FF DDE5       CALLX:      PUSH IX
1801 CD6F12                 CALL PDPSUBR      ;SWITCH ADDR IN. HL=ADDR, A=ORIG URPORT
1804 44                     LD B,H
1805 4D                     LD C,L
1806 F5                     PUSH AF           ;STACKED IN SECTION B
1807 3AD05A                 LD A,(TEMPB3)     ;JUNK OR NO. OF PARAMS IF CALL
180A CD0500                 CALL HLJUMP
180D F1                     POP AF
180E D3FB                   OUT (251),A       ;ORIG URPORT
1810 DDE1                   POP IX
1812 C9                     RET               ;TO STACKBC OR STKSTOS OR NEXTSTAT (IF CALL)
1813
1813            ;*********************************************************************************
1813            ;MEMORY$  E.G. MEM$(N1 TO N2) HANDLE READING ROM? HOW?
1813
1813 CD8B3A     IMMEMRYS:   CALL SINSISOBRK    ;CHK "("
1816 CDE43A                 CALL EXPT1NUM     ;N1
1819 FE8E                   CP TOTOK          ;"TO"
181B C2290D                 JP NZ,NONSENSE
181E
181E CDC03A                 CALL SEX1NUMCB    ;SKIP, EXPT "N)", CY IF RUNNING
1821 D0                     RET NC            ;RET IF NOT RUNNING
1822
1822 F7                     RST 30H
1823 6F50                   DW MEMRYSP2-8000H
1825
1825            ;*********************************************************************************
1825            ;HIMEM - "RAMTOP"
1825
1825 CD143B     IMHIMEM:    CALL SABORTER
1828
1828 2AB25C                 LD HL,(RAMTOP)    ;MAINTAINED IN 8000-BFFF FORM,(UNLIKE OLDRT)
182B 3AB15C                 LD A,(RAMTOPP)
182E 47                     LD B,A
182F
182F            ;USED BY TPEEK - ADJUST/STACK BHL
182F
182F DBFA       ASBHL:      IN A,(250)
1831 4F                     LD C,A
1832 78                     LD A,B
1833 91                     SUB C             ;ADJUST TO RELATIVE PAGE
1834 1808                   JR STKPGFORM
1836
1836            ;*********************************************************************************
1836            ;MEM - FREE MEMORY.
1836
1836 CD143B     IMMEM:      CALL SABORTER
1839
1839 CD8C1F                 CALL GETROOM      ;AHL=19BIT NUMBER
183C 1803                   JR STK19BIT
183E
183E            ;STACK PAGE FORM IN AHL ON FPCS
183E
183E CD2120     STKPGFORM:  CALL AHLNORM      ;TURN TO 19-BIT NUMBER
1841
1841 F5         STK19BIT:   PUSH AF
1842 CDD61C                 CALL STACKHL
1845 C1                     POP BC
1846
1846 EF                     DB CALC           ;LSW
1847 23                     DB STKBREG        ;LSW,MSB
1848 E2                     DB STK16K
1849 00                     DB MULT           ;LSW,MSB*16K
184A E0                     DB STKHALF
184B 05                     DB DIVN           ;LSW,MSB*32K
184C E0                     DB STKHALF
184D 05                     DB DIVN           ;LSW,MSB*64K
184E 01                     DB ADDN           ;MSB*64K+LSW
184F 34                     DB EXIT2
1850
1850 CD143B     IMMOUSEX:   CALL SABORTER
1853
1853 2A965B                 LD HL,(MXCRD)
1856 C3D61C                 JP STACKHL
1859
1859 CD143B     IMMOUSEY:   CALL SABORTER
185C
185C 3A985B                 LD A,(MYCRD)
185F 181B                   JR STACKAH
1861
1861 CD143B     IMPENX:     CALL SABORTER
1864
1864 01F800                 LD BC,CLUTPORT    ;A8 IS LOW
1867 ED78                   IN A,(C)
1869 1F                     RRA
186A 1F                     RRA
186B E63F                   AND 3FH
186D 180D                   JR STACKAH
186F
186F CD143B     IMPENY:     CALL SABORTER
1872
1872 01F801                 LD BC,0100H+CLUTPORT ;A8 HIGH GIVES PEN Y, NOT PEN X
1875 1803                   JR FPIN2
1877
1877            ;IN (N1) - IN A,(BC)
1877
1877 CD2E1D     FPIN:       CALL GETINT       ;TO BC
187A
187A ED78       FPIN2:      IN A,(C)
187C
```

```
187C C3DA1C     STACKAH:   JP STACKA
187F
187F            ;**************************************************************************
187F            ;SWOPS. CAN BE CALLED, OR USED AS FPC FUNCTIONS
187F            ;EXIT: DE=STKEND
187F            ;DIRECT CALLS FROM FOR-NEXT, OPEN, ...
187F
187F            ;SWOP TOP AND THIRD ENTRIES
187F
187F 0EF6       FPSWOP13:  LD C,-10
1881 1809                  JR SWOPCOM1
1883
1883            ;SWOP SECOND AND THIRD ENTRIES
1883
1883 0EFB       FPSWOP23:  LD C,-5
1885 11F6FF                LD DE,-10
1888 1805                  JR SWOPCOM2
188A
188A            ;SWOP TOP AND SECOND ENTRIES
188A
188A 0EFB       SWOP12:    LD C,-5
188C
188C 11FBFF     SWOPCOM1:  LD DE,-5
188F
188F 42         SWOPCOM2:  LD B,D             ;B=FF
1890 2A655C                LD HL,(STKEND)
1893 19                    ADD HL,DE
1894 54                    LD D,H
1895 5D                    LD E,L             ;DE=STKEND-5 (SWOP12 AND SWOP13) OR -10 (SWOP23)
1896 09                    ADD HL,BC          ;HL=STKEND-10 (SWOP12) OR -15 (SWOP13 AND SWOP23)
1897
1897            ;FP "SWOP" ENTERS HERE -
1897            ;BINARY OP, SO HL AND DE PT TO STKEND-10 AND STKEND-5
1897
1897 0605       FPSWOP:    LD B,5             ;SWOP 5 BYTES
1899
1899 1A         FPSWOPLP:  LD A,(DE)
189A 4E                    LD C,(HL)
189B 77                    LD (HL),A
189C 79                    LD A,C
189D 12                    LD (DE),A
189E 23                    INC HL
189F 13                    INC DE
18A0 10F7                  DJNZ FPSWOPLP
18A2
18A2 ED5B655C              LD DE,(STKEND)
18A6 C9                    RET
18A7
18A7            ;PI (3.1415 ETC)
18A7
18A7 CD143B     IMPI:         CALL SABORTER  ;SKIP "PI", ABORT IF NOT RUNNING
18AA
18AA EF                       DB CALC
18AB F0                       DB STKHALFPI
18AC 33                       DB EXIT
18AD
18AD 34                       INC (HL)       ;DOUBLE IT
18AE C9                       RET
18AF
18AF            ;**************************************************************************
18AF            ;ITEM - RETURN DATA LIST STATUS.
18AF            ;0=NO DATA LEFT TO READ IN CURRENT DATA STATEMENT
18AF            ;1=NEXT ITEM IS STRING
18AF            ;2=NEXT ITEM IS NUMERIC
18AF
18AF CD143B     IMITEM:    CALL SABORTER     ;SKIP "ITEM", ABORT IF NOT RUNNING
18B2
18B2 DBFB                  IN A,(URPORT)
18B4 F5                    PUSH AF
18B5 CD381F                CALL ADDRDATA     ;USE DATADD AND DATAPG TO LOOK AT DATA PTR
18B8 010000                LD BC,0
18BB 7E                    LD A,(HL)
18BC FE20                  CP " "
18BE 2804                  JR Z,IMITEM2      ;THERE IS MORE DATA IF DATADD PTS TO A SPACE
18C0                                         ;(AFTER "DATA")
18C0 FE2C                  CP ","
18C2 2018                  JR NZ,IMITEM3     ;IF NO COMMA, DATA HAS ALL BEEN READ
18C4
18C4 0C         IMITEM2:   INC C             ;BC=1 ("STRING")
18C5 CDBC33                CALL FORESP       ;SKIP ANYTHING BELOW "!"
18C8 FE22                  CP 22H
18CA 2810                  JR Z,IMITEM3      ;END IF QUOTE - STRING
18CC
18CC 7E         IMITEMLP:  LD A,(HL)
18CD 23                    INC HL
18CE CD2C3B                CALL ALPHANUM
18D1 38F9                  JR C,IMITEMLP     ;JR IF LETTER OF NUMBER
18D3
18D3 FE20                  CP " "
18D5 28F5                  JR Z,IMITEMLP     ;SPACE IS ALSO POSSIBLE IN VAR NAMES (_?)
18D7
18D7 FE24                  CP "$"
18D9 2801                  JR Z,IMITEM3      ;STRING IF NAME ENDS IN $
18DB
18DB 0C                    INC C             ;ELSE NUMERIC
18DC
18DC F1         IMITEM3:   POP AF
18DD D3FB                  OUT (URPORT),A
18DF C3DD1C                JP STACKBC
18E2
18E2                       INCLUDE DO.SAM         ;DO, LOOP, LOOP IF, EXIT IF, ON, GOSUB,
18E2            ;DO.SAM
18E2
```

```
18E2 CDFD19     DO:         CALL WHUNT          ;WHILE/UNTIL SR - ONLY RET HERE IF RUNNING
18E5 3808                   JR C,DO3            ;JR IF WANT TO EXECUTE LOOP LINES
18E7                                            ;ELSE SKIP TO "LOOP"
18E7
18E7            ;ENTRY FOR EXIT IF
18E7
18E7 D1         DO2:        POP DE              ;NEXT STAT RET ADDR
18E8 11D5D4                 LD DE,0D4D5H        ;DOTOK/LOOPTOK
18EB CD470D                 CALL SEARCH
18EE 09                     DB 9                ;"Missing LOOP"
18EF
18EF
18EF            ;STACK A RETURN ADDR AND EXECUTE LINES IN THE DO-LOOP
18EF
18EF 0680       DO3:        LD B,80H            ;MASK TO SET BIT 7 OF PAGE - SHOW "DO" DATA
18F1
18F1            ;MAKE BASIC STACK ENTRY (TYPE/PAGE, ADDR (OF LINE START), STAT)
18F1            ;ENTRY WITH B=TYPE BYTE, FROM GOSUB OR PROC, OR DO (SEE ABOVE)
18F1            ;BITS 7-5: 100=DO, 010=PROC, 000=GOSUB
18F1            ;EXIT: HL PTS TO STACKED SUBPPC (GOSUB/PROC INCS IT)
18F1
18F1 2AC45B     BSTKE:      LD HL,(BSTKEND)
18F4 11FCFF                 LD DE,-4            ;SPACE NEEDED
18F7 19                     ADD HL,DE           ;FIND NEW BSTKEND. CY
18F8 ED5BC85B               LD DE,(HEAPEND)     ;LOWER IN MEM THAN STACK - END OF USER CODE
18FC ED52                   SBC HL,DE           ;END-(LIM+1) - BSTK CANNOT COME AS FAR DOWN AS HEAP
18FE 3002                   JR NC,BSTKOK
1900
1900 CF         BSFERR:     RST 08H
1901 29                     DB 41               ;"BASIC stack full"
1902
1902 19         BSTKOK:     ADD HL,DE
1903 23                     INC HL
1904 3AAE5A                 LD A,(CLAPG)
1907 E61F                   AND 1FH
1909 B0                     OR B                ;MARK FOR TYPE
190A 77                     LD (HL),A           ;TYPE/PAGE
190B 3A475C                 LD A,(SUBPPC)
190E ED5BAF5A               LD DE,(CLA)
1912
1912 22C45B     SEDA:       LD (BSTKEND),HL     ;BSTKEND IS NOW 4 BYTES LOWER
1915 23                     INC HL
1916 73                     LD (HL),E
1917 23                     INC HL
1918 72                     LD (HL),D
1919 23                     INC HL
191A 77                     LD (HL),A
191B C9                     RET
191C
191C CD583A     LOOPIF:     CALL SYNTAX6
191F
191F CD231B                 CALL TRUETST        ;DISCARD AND TEST TRUE/FALSE.
1922 C8                     RET Z               ;NEXT STAT IF FALSE
1923
1923 37                     SCF
1924 180F                   JR LOOP1            ;UNSTACK "DO" RET ADDR, LOOP
1926
1926 CD583A     EXITIF:     CALL SYNTAX6
1929
1929 CD231B                 CALL TRUETST        ;DISCARD AND TEST TRUE/FALSE. NC
192C C8                     RET Z               ;NEXT STAT IF FALSE
192D
192D CD3519                 CALL LOOP1          ;UNSTACK "DO" RET ADDR, DON"T LOOP
1930 18B5                   JR DO2              ;SKIP TO LOOP
1932
1932
1932 CDFD19     LOOP:       CALL WHUNT          ;ASSESS WHILE/UNTIL
1935
1935 08         LOOP1:      EX AF,AF'           ;SAVE LOOP/NO LOOP AS C/NC (ALWAYS NC IF "EXIT IF",
1936                                            ;ALWAYS CY IF LOOP IF)
1936 0680                   LD B,80H            ;"DO" TYPE
1938 CD201A                 CALL RETLOOP        ;GET C=STAT, HL=ADDR, NZ IF ERROR, A=PAGE
193B 2802                   JR Z,LOOP2
193D
193D CF                     RST 08H
193E 0A                     DB 10               ;"LOOP without DO"
193F
193F 08         LOOP2:      EX AF,AF'
1940 D0                     RET NC              ;RET TO "NEXT STAT" IF "NO LOOP"
1941                                            ;RET IS TO "EXIT IF" IF CALLED FROM THERE.
1941
1941 08                     EX AF,AF'           ;PAGE
1942
1942
1942            ;USED BY RETURN, END PROC, LOOP, NEXT. ESSENTIALLY A "GOTO" STAT IN LINE AT AHL.
1942            ;ENTRY: A=PAGE, HL=ADDR OR 00XX IF ELINE, C=STAT.
1942
1942 D1         RLEPCOM:    POP DE              ;JUNK NEXT STAT RET ADDR
1943 24                     INC H
1944 25                     DEC H
1945 CA2B0D                 JP Z,LOOPEL         ;JP IF LOOPING BACK TO ELINE (ADDR MSB=0)
1948                                            ;(NSPPC WILL BE SET BY FROM C,
1948                                            ;PAGE WILL BE SET TO ELINEP, CHAD BY SKIP STATS)
1948 E61F                   AND 1FH
194A
194A            ;FROM PROCS:
194A
194A 41         RLEPC2:     LD B,C
194B C5                     PUSH BC             ;B=STAT
194C CDDF3F                 CALL SELURPG
194F C32E0E                 JP RLEPI            ;USE HL AS LINE START, A AS CHAD/CLA/NXTLN PAGE,
1952                                            ;(SP) AS STAT
1952
```

```
1952            ;*********************************************************************************
1952
1952            ;ON VALUE: STAT1: STAT2: STAT3
1952
1952 CD583A     ON:         CALL SYNTAX6
1955
1955 CD331D                 CALL GETBYTE
1958 57                     LD D,A
1959 21475C                 LD HL,SUBPPC
195C 86                     ADD A,(HL)
195D 77                     LD (HL),A           ;ADJ SUBPPC BY VALUE
195E DF                     RST 18H             ;HL=CHAD
195F CDF910                 CALL SKIPS0         ;SKIP D STATS
1962 D8                     RET C               ;RET IF HIT END OF LINE - NEXT STAT ->NXT LINE
1963
1963 DF                     RST 18H             ;PT TO ":"
1964 E5                     PUSH HL
1965 E7                     RST 20H             ;A=FIRST SIGNIF CHAR IN STAT
1966 E1                     POP HL              ;":" PTR
1967 CD1E3B                 CALL ALPHA
196A 3804                   JR C,ON2            ;JR IF LETTER - IT"S A PROC
196C
196C FEB5                   CP 0B5H             ;GOSUBTOK
196E 2013                   JR NZ,ON3           ;JR UNLESS "GOSUB"
1970
1970            ;PROCS AND GOSUBS MUST RETURN TO NEXT LINE AFTER EXECUTING
1970
1970 22975A     ON2:        LD (CHAD),HL        ;PT TO ":"
1973 2A9D5A                 LD HL,(NXTLINE)
1976 22AF5A                 LD (CLA),HL         ;MAKE IT LOOK AS THOUGH WE ARE AT NEXT LINE
1979 21475C                 LD HL,SUBPPC
197C 7E                     LD A,(HL)
197D 32465B                 LD (ONSTORE),A      ;SAVE SUBPPC, WHICH WILL BE ZERO AFTER INCR
1980                                            ;ANY ERRORS WILL GIVE SENSIBLE STAT NO. BECAUSE
1980                                            ;ERROR HANDLER USES (ONSTORE) IF STAT NO=00. THIS
1980 36FF                   LD (HL),255         ;WILL BE INCED TO ZERO. GOSUB/PROCS WILL THINK
1982                                            ;WE ARE AT A STAT ZERO, NEXT LINE, AND WILL RETURN
1982                                            ;TO STAT 1, NEXT LINE
1982 C9                     RET                 ;TO NEXT STAT
1983
1983 D1         ON3:        POP DE              ;NEXT STAT
1984 FEB4                   CP 0B4H             ;GOTOTOK
1986 2803                   JR Z,ON4            ;GOTO KEEPS DE AS NEXT STAT ADDR
1988
1988 11170E                 LD DE,OLNEND        ;ELSE USE LINEEND SO ONLY 1 STAT EXECUTED
198B
198B FE3A       ON4:        CP ":"              ;SEE IF NULL STATEMENT
198D C29E0D                 JP NZ,ON4ENT        ;PUSHES DE (NEXT STAT OR LINE END),EXECUTES STAT.
1990                                            ;RETURN GOES TO LINE END UNLESS GOTO WAS USED
1990
1990 EB                     EX DE,HL
1991 E9                     JP (HL)             ;NULL STAT (:) JPS TO LINE END
1992
1992 CD2E1D     GOTO2:      CALL GETINT
1995 7C                     LD A,H
1996 3C                     INC A
1997
1997 CA391D     GTERRHP:    JP Z,IOORERR       ;RANGE 0-65279 (0000-FEFFH)
199A
199A AF         GOTO3:      XOR A               ;STAT NO. ZERO
199B
199B 32445C     GOTO4:      LD (NSPPC),A
199E 22425C                 LD (NEWPPC),HL
19A1 C9                     RET
19A2
19A2 CD153B     CONTINUE:   CALL CHKEND
19A5
19A5 3A705C     CONTINUE2:  LD A,(OSPPC)
19A8 2A6E5C                 LD HL,(OLDPPC)
19AB 18EE                   JR GOTO4
19AD
19AD            ;CALBAS - CALL BASIC SUBROUTINE FROM MACHINE CODE
19AD            ;ENTRY: HL=LINE TO CALL.
19AD            ;EXIT: Z IF OK, ELSE A=ERROR NUMBER
19AD
19AD CD9A19     CALBAS:     CALL GOTO3          ;LD (NEWPPC),HL: ZERO (NSPPC)
19B0 47                     LD B,A              ;TYPE/PAGE=GOSUB/PAGE 0
19B1 3D                     DEC A
19B2 32475C                 LD (SUBPPC),A       ;"STATMENT" FF SHOWS M/C
19B5 DBFB                   IN A,(251)
19B7 F5                     PUSH AF
19B8 CDF118                 CALL BSTKE          ;STACK RETURN ADDR, STAT FFH. INSTEAD OF DOING A
19BB                                            ;BASIC RETURN, "RETURN" CAUSES RET TO ERR HANDLER.
19BB CD2E20                 CALL SETESP         ;SET ERRSP SO ERRORS RETURN TO THIS ROUTINE
19BE CDD40D                 CALL NEXTSTAT       ;RUN LINE. NORMAL BASIC STACK AT "NEXTSTAT" HOLDS
19C1                                            ;MAINER ADDR, WITH ERRSP POINTING TO IT. NOW ERRSP
19C1                                            ;PTS TO THIS ROUTINE
19C1 E1                     POP HL
19C2 223D5C                 LD (ERRSP),HL       ;RESTORE ORIG.
19C5 F1                     POP AF
19C6 D3FB                   OUT (251),A         ;ORIG URPAGE
19C8 3A3A5C                 LD A,(ERRNR)
19CB A7                     AND A
19CC C9                     RET
19CD
19CD
19CD CD153B     RETURN:     CALL CHKEND
19D0
19D0 0600                   LD B,0              ;"GOSUB" TYPE
19D2 CD201A                 CALL RETLOOP        ;GET RET ADDR
19D5 2005                   JR NZ,RWGERR        ;HL=LINE ADDR, A=TYPE/PAGE, C=STAT, Z IF TYPE OK
19D7
19D7 0C                     INC C               ;RETURN TO *NEXT* STAT
```

```
19D8 2013                   JR NZ,ENDP1       ;STAT IS ONLY 0FFH IF CALL CAME FROM M/C
19DA
19DA C1                     POP BC
19DB C9                     RET               ;TO ERROR HANDLER (M/C)
19DC
19DC CF           RWGERR:   RST 08H
19DD 08                     DB 8              ;"RETURN without GOSUB"
19DE
19DE CD153B       ENDPROC:  CALL CHKEND
19E1
19E1 CD2C33                 CALL DPRA         ;GET RET ADDR. C=STAT, HL=ADDR, A=PAGE
19E4 E5                     PUSH HL
19E5 C5                     PUSH BC
19E6 F5                     PUSH AF
19E7 CD0A37                 CALL DELOCAL
19EA F1                     POP AF
19EB C1                     POP BC
19EC E1                     POP HL
19ED
19ED              ;FROM RETURN, IF TYPE OK
19ED
19ED 47           ENDP1:    LD B,A
19EE 3A455B                 LD A,(ONERRFLG)   ;T BIT,000000,P BIT
19F1 1F                     RRA
19F2 3005                   JR NC,ENDP2       ;JR IF "ON ERROR" PERM OFF
19F4
19F4 3E81                   LD A,81H          ;END PROC/RETURN RESET TEMP ERROR BIT TO PERM
19F6                                          ;STATUS SO ERROR PROC OR SR EASY.
19F6 32455B                 LD (ONERRFLG),A   ;TEMP AND PERM NOW ON
19F9
19F9 78           ENDP2:    LD A,B
19FA C34219                 JP RLEPCOM        ;USE HL AS LINE ADDR, A AS PAGE, C AS STAT
19FD
19FD
19FD              ;WHILE/UNTIL SR OF DO AND LOOP COMMANDS
19FD
19FD FE8A         WHUNT:    CP WHILETOK
19FF 280B                   JR Z,WHUNT2
1A01
1A01 FE8B                   CP UNTILTOK
1A03 37                     SCF
1A04 2806                   JR Z,WHUNT2
1A06
1A06 E1                     POP HL            ;RET ADDR IN DO OR LOOP ROUTINE
1A07 CDC63A                 CALL RUNFLG
1A0A D0                     RET NC            ;NEXT STAT IF SYNTAX TIME - CHECK "DO" OR "LOOP"
1A0B
1A0B E9                     JP (HL)           ;RET WITH C IF NO QUALIFIERS (NO WHILE OR UNTIL)
1A0C
1A0C F5           WHUNT2:   PUSH AF           ;WHILE/UNTIL FLAG
1A0D CDE33A                 CALL SEXPT1NUM    ;SKIP WHILE/UNTIL, GET EXPR
1A10 CDC63A                 CALL RUNFLG
1A13 3008                   JR NC,WHUNT3      ;GOTO NEXT STAT NOW IF SYNTAX TIME
1A15
1A15 CD231B                 CALL TRUETST      ;DROP AND TEST EXPR.
1A18 2804                   JR Z,WHUNT4       ;JR IF FALSE
1A1A
1A1A F1                     POP AF            ;IF TRUE AND UNTIL, NC
1A1B 3F                     CCF               ;IF TRUE AND WHILE, C
1A1C C9                     RET
1A1D
1A1D F1           WHUNT3:   POP AF            ;ENTRY TO JUNK FLAG, RET ADDR, RET TO NEXT STAT
1A1E
1A1E F1           WHUNT4:   POP AF            ;IF FALSE AND UNTIL, C
1A1F C9                     RET               ;IF FALSE AND WHILE, NC
1A20
1A20              ;IF CALLED FROM LOOP, C MEANS EXECUTE THE LOOP, ELSE CONTINUE
1A20              ;IF CALLED FROM  DO , C MEANS EXECUTE DO, ELSE FIND LOOP AND JP THERE
1A20
1A20              ;RETURN/LOOP SR. ENTRY: B=DESIRED TYPE OF DATA TO UNSTACK FROM BASIC STACK.
1A20              ;80H=DO, 40H=PROC, 00=GOSUB
1A20              ;EXIT: A=STAT NR, HL=ADDR OF LINE, C=TYPE/PAGE
1A20              ;IF ENTRY WAS AT RETLOOP, NZ=WRONG TYPE/EMPTY STACK, Z=OK
1A20
1A20 2AC45B       RETLOOP:  LD HL,(BSTKEND)
1A23 7E                     LD A,(HL)         ;TYPE/PAGE
1A24 E6E0                   AND 0E0H          ;ISOLATE TYPE BITS
1A26 B8                     CP B
1A27 C0                     RET NZ            ;RET IF WRONG TYPE OR STACK MT (FF STOPPER)
1A28
1A28 7E           RETLOOP2: LD A,(HL)         ;TYPE/PAGE
1A29 23                     INC HL
1A2A 5E                     LD E,(HL)
1A2B 23                     INC HL
1A2C 56                     LD D,(HL)         ;ADDR
1A2D 23                     INC HL
1A2E 4E                     LD C,(HL)         ;STAT
1A2F 23                     INC HL
1A30 22C45B                 LD (BSTKEND),HL
1A33 EB                     EX DE,HL
1A34 C9                     RET
1A35
1A35
1A35              ;FIND LINE NO. "HL" (OR "BC", WITH LATER ENTRY)
1A35              ;STARTS SEARCH AT PROG, OR IF RUNNING AND TARGET IS AT OR PAST EPPC, SEARCHES
1A35              ;FROM CURRENT LINE (PPC) ADDRESS. USES HL,DE,BC,AF, *TEMPW1*
1A35              ;ENTRY:HL=LINE NO.
1A35              ;EXIT: HL PTS. TO LINE NO. MSB IN PROGRAM, DE PTS TO PREVIOUS LINE. DE IS IN
1A35              ;8000-BFFF AREA, HL MIGHT HAVE CROSSED IN TO C000 BY A LINE LEN OR SO.
1A35              ;IF NO PROGRAM, DE=HL
1A35              ; Z=LINE FOUND
1A35              ;NZ=FOUND A LATER LINE, OR FF STOPPER
1A35
```

```
1A35 44          FNDLNHL:   LD B,H
1A36 4D                     LD C,L
1A37
1A37 CDC63A      FNDLNBC:   CALL RUNFLG
1A3A
1A3A 3013                   JR NC,FNDLP       ;JR IF NOT RUNNING (E.G. EDITING)
1A3C
1A3C 2A455C                 LD HL,(PPC)
1A3F 2B                     DEC HL            ;SO CY IF PPC=TARGET
1A40 A7                     AND A
1A41 ED42                   SBC HL,BC
1A43 300A                   JR NC,FNDLP       ;JR IF DESIRED LINE IS BEFORE CURRENT LINE
1A45                                          ;(ALWAYS, IF PPC=FFFF (ELINE))
1A45
1A45 2AAF5A                 LD HL,(CLA)       ;START LOOKING FROM PPC LINE START
1A48 3AAE5A                 LD A,(CLAPG)      ;NEEDED? OR USE JR+2
1A4B 1808                   JR FNDL0
1A4D
1A4D             ;ENTRY HERE IF WANT TO INSIST ON STARTING AT PROG (RENUM)
1A4D
1A4D 44          FNDLINE:   LD B,H
1A4E 4D                     LD C,L
1A4F
1A4F 2AA05A      FNDLP:     LD HL,(PROG)
1A52 3A9F5A                 LD A,(PROGP)
1A55
1A55 CDDF3F      FNDL0:     CALL TSURPG       ;SWITCH IN A PROGRAM BLOCK
1A58 22C85A                 LD (TEMPW1),HL    ;KEEP PTR TO LINO MSB
1A5B 180F                   JR FNDL2
1A5D
1A5D CB74        FNDL1:     BIT 6,H
1A5F C4F23F                 CALL NZ,INCURPAGE
1A62 22C85A                 LD (TEMPW1),HL    ;KEEP PTR TO LINO MSB
1A65 23                     INC HL
1A66 23                     INC HL
1A67 5E                     LD E,(HL)
1A68 23                     INC HL
1A69 56                     LD D,(HL)
1A6A 23                     INC HL
1A6B 19                     ADD HL,DE         ;ADD LINE LEN TO PT TO NEXT LINE NO.
1A6C
1A6C 7E          FNDL2:     LD A,(HL)         ;GET MSB OF LINE NO.
1A6D B8                     CP B              ;CP MSB OF TARGET
1A6E DA5D1A                 JP C,FNDL1        ;JP IF NOT AT OR PAST TARGET LINE YET
1A71
1A71 2007                   JR NZ,FNDL3       ;JR IF MSB SHOWS WE ARE PAST TARGET LINE NO.
1A73
1A73 23                     INC HL
1A74 7E                     LD A,(HL)         ;LSB
1A75 2B                     DEC HL
1A76 B9                     CP C
1A77 DA5D1A                 JP C,FNDL1        ;JP IF LSB SHOWS WE ARE NOT AT TARGET YET
1A7A
1A7A ED5BC85A    FNDL3:     LD DE,(TEMPW1)
1A7E C9                     RET
1A7F
1A7F             ;IF SAM - 2.3.89
1A7F             ;*************************************************************************
1A7F             ;E.G. IF x=1 THEN PRINT
1A7F             ;     IF X=1 THEN PRINT "Y": ELSE PRINT "N"
1A7F             ;     IF X=1: PRINT: PRINT: END IF
1A7F
1A7F             ;BOTH LONG AND SHORT IFS COME HERE
1A7F
1A7F             LIF:
1A7F 2A975A      SIF:       LD HL,(CHAD)
1A82
1A82 2B          SIFLP:     DEC HL
1A83 7E                     LD A,(HL)
1A84 FE21                   CP 21H
1A86 38FA                   JR C,SIFLP           ;PT HL TO CMD CODE
1A88
1A88 E5                     PUSH HL
1A89 CDE43A                 CALL EXPT1NUM
1A8C E1                     POP HL
1A8D 32755A                 LD (IFTYPE),A        ;RECORD LAST IF TYPE AS SHORT/LONG USING
1A90                                             ; "THEN"/NON "THEN"
1A90                                             ;(IF TYPE IS SET TO "LONG" AT START OF LINE
1A90                                             ;SYNTAX CHECK)
1A90 FE8D                   CP THENTOK
1A92 57                     LD D,A
1A93 2002                   JR NZ,IFL1           ;JR IF NOT "THEN" (SHOULD BE ":")
1A95
1A95 36D8                   LD (HL),0D8H         ;SIFTOK
1A97                                             ;AT LINE ENTRY, I-F IS TOKENISED AS THE "LIF"
1A97                                             ;TOKEN BECAUSE IT OCCURS FIRST IN TOKEN LIST
1A97                                             ;(LIF TOKEN LISTS AS I-F, LIKE THE IF TOKEN)
1A97                                             ;NOW FORCE TOKEN TO "SIF" BECAUSE "THEN" USED.
1A97
1A97 CD153B      IFL1:      CALL CHKEND          ;IF SYNTAX CHECK, CHECK FOR OD/:/THEN, EXIT
1A9A
1A9A CD231B                 CALL TRUETST         ;DROPS EXPR AND TESTS IT
1A9D C0                     RET NZ               ;RET IF TRUE, DO NEXT STATMENT (FPCS IS CLEARED)
1A9E
1A9E C1                     POP BC               ;NEXT STAT
1A9F 7A                     LD A,D
1AA0 FE8D                   CP THENTOK
1AA2 282B                   JR Z,SHORTIF         ;IF SHORT AND NOT TRUE, LOOK FOR ELSE OR CR
1AA4
1AA4 D9          EIFLP:     EXX
1AA5 01D9FF                 LD BC,0FF00H+0D9H    ;LELSETOK
1AA8 D9                     EXX                  ;WHOLE PROG (OR ELINE), TARGET 2 RELOAD=LELSE
1AA9
```

```
1AA9 01D901               LD BC,0100H+0D9H     ;LELSETOK. COUNT=1, TARGET2=LELSE
1AAC 11DBD7               LD DE,0D7DBH         ;LIFTOK/ENDIFTOK
1AAF                                           ;LOOK FOR LELSE OR ENDIF WITH LIF INTERVENING
1AAF CDA61D               CALL SRCHALL3
1AB2 3019                 JR NC,MEIERR         ;ERROR IF NEITHER WAS FOUND
1AB4
1AB4 32475C               LD (SUBPPC),A        ;STAT
1AB7 08                   EX AF,AF'            ;FINAL TARGET
1AB8 BB                   CP E                 ;WAS IT "END IF"?
1AB9 CACA1A               JP Z,XCHDH           ;IF IT WAS, CONTINUE AFTER "END IF"
1ABC
1ABC DF                   RST 18H
1ABD FED8                 CP 0D8H              ;SIFTOK
1ABF C2CA1A               JP NZ,XCHDH          ;JP IF "LELSE", NOT "LELSE SIF"; CONTINUE AFTER
1AC2                                           ;"LELSE"
1AC2
1AC2 CDE33A               CALL SEXPT1NUM       ;SKIP "SIF", EVAL CONDITION
1AC5 CD231B               CALL TRUETST
1AC8 28DA                 JR Z,EIFLP           ;IF FALSE, KEEP LOOKING FOR ENDIF/LELSE
1ACA
1ACA C3510D     XCHDH:    JP EXCHAD2           ;IF TRUE, CONTINUE AFTER "ELSE IF cond"
1ACD
1ACD CF         MEIERR:   RST 08H
1ACE 27                   DB 39                ;"Missing END IF"
1ACF
1ACF D9         SHORTIF:  EXX
1AD0 018D00               LD BC,THENTOK        ;ONE LINE SRCH, NULL TARGET2 RELOAD
1AD3 D9                   EXX
1AD4
1AD4 11DAD8               LD DE,0D8DAH         ;SIFTOK/ELSETOK
1AD7                                           ;LOOK FOR "ELSE" WITH SIF "INTERVENING"
1AD7 CDA31D               CALL SRCHALL2
1ADA D2120E               JP NC,LINEEND        ;JP TO LINE END IF NO "ELSE" FOUND
1ADD
1ADD 32475C               LD (SUBPPC),A
1AE0 C3840D               JP STMTLP2           ;JP TO NEXT STATEMENT, JUST AFTER "ELSE"
1AE3                                           ;(CHAD, CLA, NXTLINE STILL OK - SAME LINE)
1AE3
1AE3            ;E-L-S-E IS TOKENISED AS LELSE TOKEN BECAUSE IT APPEARS FIRST IN LIST
1AE3
1AE3 4F         LELSE:    LD C,A               ;CHAR AFTER "ELSE"
1AE4 CDC63A               CALL RUNFLG
1AE7 3828                 JR C,RLELSE          ;JR IF RUNNING A LONG ELSE
1AE9
1AE9 2A975A               LD HL,(CHAD)
1AEC
1AEC 2B         FELSLP:   DEC HL
1AED 7E                   LD A,(HL)
1AEE FE21                 CP 21H
1AF0 38FA                 JR C,FELSLP          ;PT TO CMD CODE
1AF2
1AF2 3A755A               LD A,(IFTYPE)
1AF5 FE8D                 CP THENTOK
1AF7 2814                 JR Z,NLELS           ;JR IF "SHORT" STATUS
1AF9
1AF9 79                   LD A,C
1AFA D6D8                 SUB 0D8H             ;SIFTOK
1AFC CE00                 ADC A,0              ;SIF/LIF BOTH BECOME 0
1AFE 201F                 JR NZ,ELSE2          ;CHECK SYNTAX AFTER "LELSE"
1B00
1B00 2A975A               LD HL,(CHAD)
1B03 36D8                 LD (HL),0D8H         ;"LELSE LIF" BECOMES "LELSE SIF"
1B05 CDE33A               CALL SEXPT1NUM       ;EVAL condition
1B08 FE8D                 CP THENTOK
1B0A C0                   RET NZ               ;TO CHECK SYNTAX OF NEXT STATEMENT
1B0B
1B0B CF         DNS:      RST 08H              ;"LELSE SIF cond THEN" NOT ALLOWED AS IT
1B0C 1D                   DB 29                ;WILL WORK IN A CONFUSING WAY
1B0D
1B0D 36DA       NLELS:    LD (HL),0DAH         ;ELSETOK
1B0F                                           ;FORCE "SHORT ELSE" IF "SHORT IF" PRECEDED IT.
1B0F                                           ;(THERE WAS A PRECEDING "SHORT IF" ON THIS LINE)
1B0F 180E                 JR ELSE2             ;CHECK SYNTAX FROM CHAD ON
1B11
1B11            ;LONG ELSE REQUIRES A SEARCH FOR ENDIF, WITH LIF INTERVENING
1B11
1B11 D1         RLELSE:   POP DE               ;JUNK NEXT STAT ** BUG FIX
1B12 11DBD7               LD DE,0D7DBH         ;LIFTOK/ENDIFTOK
1B15 CD470D               CALL SEARCH          ;LOOK FOR "ENDIF" WITH LIF "INTERVENING"
1B18 00                   DB 0                 ;"OK" IF NOT FOUND
1B19
1B19            ;SHORT ELSE - SHOULD ALWAYS BE RUNNING!
1B19
1B19 CDC63A     ELSE:     CALL RUNFLG
1B1C DA110E               JP C,REMARK          ;JP, SKIP REST OF LINE IF RUNNING A SHORT ELSE
1B1F
1B1F C1         ELSE2:    POP BC               ;NEXT STAT
1B20 C3840D               JP STMTLP2           ;CHECK SYNTAX FROM CHAD ONWARDS WITHOUT
1B23                                           ;REQUIRING CR/COLON
1B23
1B23            ;DROP EXPRESSION FROM FPCS, TEST FOR TRUE/FALSE
1B23
1B23 2A655C     TRUETST:  LD HL,(STKEND)    ;PT TO END OF RESULT OF THE EXPRESSION
1B26 2B                   DEC HL
1B27 2B                   DEC HL
1B28 7E                   LD A,(HL)         ;MSB IF INTEGER
1B29 2B                   DEC HL
1B2A B6                   OR (HL)           ;LSB
1B2B 2B                   DEC HL
1B2C B6                   OR (HL)           ;SGN - PROB NOT NEEDED IF MINUS ZERO EXCLUDED!
1B2D 2B                   DEC HL
1B2E B6                   OR (HL)           ;EXP.
1B2F 22655C               LD (STKEND),HL    ;"DISCARD"
```

```
1B32                                        ;NZ IF TRUE, DO NEXT STATMENT (FPCS IS CLEARED)
1B32
1B32 C9         ENDIF:     RET                 ;DOES NOTHING - ACTS AS A MARKER ONLY
1B33
1B33
1B33            ;FOR.SAM 26.5.89
1B33            ;*************************************************************************
1B33            '
1B33 CDD32C     FOR:           CALL SYNTAX4     ;ASSESS FOR-VARIABLE
1B36
1B36 DF                        RST 18H
1B37 FE3D                      CP "="
1B39 20D0                      JR NZ,DNS        ;NONSENSE
1B3B
1B3B CDE33A                    CALL SEXPT1NUM   ;START VALUE
1B3E FE8E                      CP TOTOK
1B40 C20B1B                    JP NZ,DNS        ;NONSENSE
1B43
1B43 CDE33A                    CALL SEXPT1NUM   ;LIMIT
1B46 FE8F                      CP STEPTOK       ;"STEP"
1B48 2807                      JR Z,FORSTEP
1B4A
1B4A CD153B                    CALL CHKEND
1B4D
1B4D EF                        DB CALC
1B4E E9                        DB STKONE        ;DEFAULT STEP OF 1
1B4F 33                        DB EXIT
1B50
1B50 14                        INC D            ;NZ
1B51
1B51 CC573A     FORSTEP:       CALL Z,SSYNTAX6  ;GET THE STEP VALUE
1B54
1B54                                            ;VALUE/LIMIT/STEP ON FPCS
1B54 DF         FOR2:          RST 18H
1B55 F5                        PUSH AF          ;CR OR COLON
1B56 CD8A18                    CALL SWOP12      ;V/S/L
1B59 CD7F18                    CALL FPSWOP13    ;L/S/V
1B5C
1B5C 216051                    LD HL,TLBYTE+33
1B5F CBF6                      SET 6,(HL)       ;"FOR-NEXT" TYPE MARKED ON T/L BYTE
1B61 CD372B                    CALL ASSISR      ;IF NORMAL VAR EXISTS, DEST PTS TO PREV PTR, AND
1B64                                            ;FLAGX BIT 0 SHOWS "NEW", SO VAR IS "LINKED OUT"
1B64                                            ;ASSIGNS V TO FIRST 5 LOCNS. DE PTS AFTER
1B64                                            ;THESE ON EXIT (14 EXTRA LOCATIONS AVAILABLE IF
1B64                                            ;"OLD" FOR-NEXT, ELSE NUMEND IS PAST VAR, NEEDS
1B64                                            ;MOVING 14 ON)
1B64 2A655C                    LD HL,(STKEND)
1B67 010A00                    LD BC,10
1B6A A7                        AND A
1B6B ED42                      SBC HL,BC
1B6D 22655C                    LD (STKEND),HL   ;DELETE L,S
1B70 EDB0                      LDIR             ;COPY TO VARIABLE BUFFER TO GIVE V,L,S
1B72 F1                        POP AF           ;STAT END CHAR
1B73 D5                        PUSH DE
1B74 1B                        DEC DE
1B75 EB                        EX DE,HL         ;SRC=END OF S
1B76 112F51                    LD DE,MEMVAL+14
1B79 0E0F                      LD C,15
1B7B EDB8                      LDDR             ;COPY V,L,S TO MEM 0,1,2 FOR NEXTTEST TO USE
1B7D FE0D                      CP 0DH
1B7F 2811                      JR Z,FOR22       ;JR IF LOOPING ADDRESS IS ON NEXT LINE
1B81
1B81 3A475C                    LD A,(SUBPPC)
1B84 3C                        INC A
1B85 4F                        LD C,A
1B86 3A465C                    LD A,(PPC+1)
1B89 3C                        INC A
1B8A 67                        LD H,A
1B8B 280A                      JR Z,FOR25       ;JR IF ELINE
1B8D
1B8D 2AAF5A                    LD HL,(CLA)      ;ELSE THIS LINE, NEXT STAT
1B90 1805                      JR FOR25
1B92
1B92 2A9D5A     FOR22:         LD HL,(NXTLINE)
1B95 0E01                      LD C,1           ;FIRST STATEMENT
1B97
1B97 EB         FOR25:         EX DE,HL
1B98 E1                        POP HL           ;VARS PTR
1B99 3A9C5A                    LD A,(NXTLINEP)  ;PAGE OF CURRENT LINE (SAME AS CLAPG)
1B9C 77                        LD (HL),A        ;ORDER CHANGE VS. ROM 1.0**
1B9D 23                        INC HL
1B9E 73                        LD (HL),E
1B9F 23                        INC HL
1BA0 72                        LD (HL),D        ;LINE ADDRESS
1BA1 23                        INC HL
1BA2 71                        LD (HL),C        ;STAT
1BA3 23                        INC HL
1BA4 3A715C                    LD A,(FLAGX)
1BA7 1F                        RRA
1BA8 EB                        EX DE,HL
1BA9 DCBB2B                    CALL C,NELOAD    ;SET NUMEND IF "NEW" VARIABLE
1BAC
1BAC CD1C1C                    CALL NEXTTEST
1BAF C0                        RET NZ           ;RET IF A LOOP IS POSSIBLE
1BB0
1BB0 CDD73F                    CALL SELCHADP
1BB3
1BB3 1EC1       FORMLP:        LD E,0C1H        ;NEXTTOK
1BB5 CD9C1D                    CALL SRCHPROG    ;LOOK FOR "NEXT" FROM CHAD ONWARDS
1BB8 3802                      JR C,FOR3        ;JR IF ONE FOUND
1BBA
1BBA CF                        RST 08H
1BBB 06                        DB 6             ;"FOR without NEXT"
```

```
1BBC
1BBC 32475C     FOR3:         LD (SUBPPC),A    ;STAT
1BBF 116051                   LD DE,TLBYTE+33
1BC2 CD0A31                   CALL MATCHFN     ;CHECK (HL) VS (TLBYTE+33) OVER T/L+1 BYTES
1BC5 38EC                     JR C,FORMLP      ;LOOP IF THE WRONG NEXT VARIABLE
1BC7
1BC7 ED53975A                 LD (CHAD),DE     ;SKIP var - DE PTS TO PAST VAR NAME
1BCB D1                       POP DE           ;JUNK NEXT STAT
1BCC C3510D                   JP EXCHAD2       ;CONTINUE EXECUTION AFTER "NEXT var"
1BCF
1BCF CF         NWFERR:       RST 08H
1BD0 05                       DB 5             ;"NEXT without FOR"
1BD1
1BD1            ;*********************************************************************
1BD1
1BD1 CDD32C     NEXT:         CALL SYNTAX4     ;ASSESS "FOR" VARIABLE
1BD4 CD153B                   CALL CHKEND
1BD7
1BD7 CD630E                   CALL BRKSTOP   ;TEST FOR BREAK (RLEPCOM BELOW AVOIDS NORMAL
1BDA                                         ;BETWEEN-STATEMENT TEST)
1BDA
1BDA 3A725C                   LD A,(STRLEN)    ;TYPE BYTE (FROM NVARS, IF VAR FOUND)
1BDD E640                     AND 40H          ;BIT 6 SET=FOR-NEXT TYPE, FOUND
1BDF 28EE                     JR Z,NWFERR
1BE1                                           ;5 BYTE VALUE, 5 BYTE LIMIT, 5 BYTE STEP
1BE1                                           ;2 BYTE ADDRESS, 1 BYTE PAGE, 1 BYTE STAT. NO.
1BE1 CD1B1F                   CALL ADDRDEST    ;PT TO VALUE
1BE4 CD2D1C                   CALL NEXTSR
1BE7 281F                     JR Z,NEXT1       ;JR IF INTEGER MATHS ALREADY DONE
1BE9
1BE9 2AA65A                   LD HL,(DEST)
1BEC E5                       PUSH HL
1BED ED5B685C                 LD DE,(MEM)
1BF1 010F00                   LD BC,15
1BF4 EDB0                     LDIR             ;COPY VAR TO CALC MEMS 0,1,2
1BF6
1BF6 EF                       DB CALC
1BF7 D8                       DB RCL0          ;V
1BF8 DA                       DB RCL2          ;V,S
1BF9 01                       DB ADDN          ;V+S
1BFA C8                       DB STOD0
1BFB 33                       DB EXIT
1BFC
1BFC EB                       EX DE,HL         ;HL PTS TO DROPPED NEW V
1BFD D1                       POP DE
1BFE 010500                   LD BC,5
1C01 EDB0                     LDIR             ;COPY NEW V BACK TO VARS
1C03
1C03 CD1C1C                   CALL NEXTTEST
1C06 C8                       RET Z            ;RET IF "NO LOOP"
1C07
1C07 21                       DB 21H           ;="JR +2"
1C08
1C08 A7         NEXT1:        AND A
1C09 C8                       RET Z            ;RET IF INTEGER MATHS SHOWS "NO LOOP"
1C0A
1C0A 110F00                   LD DE,15
1C0D 2AA65A                   LD HL,(DEST)
1C10 19                       ADD HL,DE
1C11 7E                       LD A,(HL)        ;A=PAGE OF LOOPING LINE **
1C12 23                       INC HL
1C13 5E                       LD E,(HL)
1C14 23                       INC HL
1C15 56                       LD D,(HL)        ;DE=ADDR OF LOOPING LINE (8000-BFFF) OR
1C16                                           ;00?? IF ELINE IS THE LOOPING LINE
1C16 23                       INC HL
1C17 4E                       LD C,(HL)        ;LOOPING STATMENT NO.
1C18 EB                       EX DE,HL         ;HL=ADDR, A=PAGE, C=STAT
1C19 C34219                   JP RLEPCOM       ;GOTO STAT C IN LINE AT AHL
1C1C
1C1C
1C1C            ;CHECK TO SEE IF "LIMIT" HAS BEEN EXCEEDED BY "VALUE"
1C1C
1C1C EF         NEXTTEST:     DB CALC
1C1D D8                       DB RCL0
1C1E D9                       DB RCL1
1C1F DA                       DB RCL2          ;V,L,S
1C20 2E                       DB GRTR0         ;V,L,TRUE/FALSE
1C21 1E                       DB JPTRUE
1C22 02                       DB 02H           ;TO NEXTTST1
1C23
1C23 06                       DB SWOP          ;SWOP IF STEP IS NEGATIVE
1C24
1C24 03         NEXTTST1:     DB SUBN          ;V-L IF +VE STEP
1C25 42                       DB SGN
1C26 07                       DB DROP
1C27 33                       DB EXIT
1C28
1C28 13                       INC DE           ;DE PTS TO DROPPED SGN(V-L) (OR L-V) SGN BYTE
1C29 13                       INC DE
1C2A 1A                       LD A,(DE)
1C2B 3D                       DEC A            ;SGN IS -1/0/1 SO A=1 IF SGN 1, ELSE 00 OR FF
1C2C C9                       RET              ;NZ IF LOOP POSSIBLE, Z IF NOT (SGN=+VE)
1C2D
1C2D            ;*********************************************************************
1C2D            ;NEXT SUBROUTINE
1C2D            ;ENTRY: HL AND TEMPW1 PTS TO FIRST BYTE OF FOR-NEXT VARIABLE
1C2D            ;EXIT: NZ="USE FLOATING POINT". Z=INTEGER MATH DONE. A=00 IF NO LOOP, OR FF
1C2D
1C2D AF         NEXTSR:       XOR  A
1C2E BE                       CP   (HL)
1C2F C0                       RET  NZ          ;RET IF VALUE=FP
1C30
```

```
1C30 23                         INC  HL
1C31 46                         LD   B,(HL)
1C32 23                         INC  HL
1C33 5E                         LD   E,(HL)
1C34 23                         INC  HL
1C35 56                         LD   D,(HL)      ;DE=VALUE, B=SGN OF VALUE
1C36 23                         INC  HL
1C37 23                         INC  HL
1C38 BE                         CP   (HL)
1C39 C0                         RET  NZ          ;RET IF LIMIT=F.P.
1C3A
1C3A 23                         INC HL
1C3B 23                         INC HL
1C3C 23                         INC HL
1C3D 23                         INC HL
1C3E 23                         INC HL
1C3F BE                         CP (HL)
1C40 C0                         RET  NZ          ;RET IF STEP=F.P.
1C41
1C41 23                         INC  HL
1C42 7E                         LD   A,(HL)
1C43 08                         EX   AF,AF'      ;A AND A"=SGN OF STEP
1C44 7E                         LD   A,(HL)
1C45 23                         INC  HL
1C46 4E                         LD   C,(HL)
1C47 23                         INC  HL
1C48 66                         LD   H,(HL)
1C49 69                         LD   L,C         ;HL=STEP
1C4A 19                         ADD  HL,DE       ;ADD STEP, VALUE
1C4B 88                         ADC  A,B         ;ADD SGN STEP,SGN VALUE,CARRY FLAG
1C4C 0F                         RRCA
1C4D CE00                       ADC  A,0
1C4F C0                         RET  NZ          ;RET IF OVERFLOW OF INTEGER MATHS
1C50
1C50 9F                         SBC  A,A         ;A=SGN OF NEW VALUE
1C51 EB                         EX   DE,HL       ;DE=NEW VALUE
1C52 2AA65A                     LD   HL,(DEST)
1C55 23                         INC  HL          ;SKIP 00
1C56 77                         LD   (HL),A      ;PLACE SGN
1C57 47                         LD   B,A         ;B=SGN OF NEW VALUE
1C58 23                         INC  HL
1C59 73                         LD   (HL),E
1C5A 23                         INC  HL
1C5B 72                         LD   (HL),D      ;PLACE NEW VALUE
1C5C 23                         INC HL
1C5D 23                         INC HL
1C5E 23                         INC HL
1C5F 7E                         LD A,(HL)        ;A=SGN OF LIMIT
1C60 23                         INC  HL
1C61 4E                         LD   C,(HL)
1C62 23                         INC  HL
1C63 66                         LD   H,(HL)
1C64 69                         LD   L,C         ;HL=LIMIT
1C65 A8                         XOR  B           ;XOR SGN OF LIMIT,SGN OF VALUE
1C66 2007                       JR   NZ,NEXTSR1  ;JR IF THEY DO NOT MATCH
1C68
1C68 3D                         DEC  A           ;A=FF ("LOOP")
1C69 ED52                       SBC  HL,DE
1C6B C8                         RET  Z           ;RET IF LIMIT=VALUE - LOOP
1C6C
1C6C 9F                         SBC  A,A
1C6D 2F                         CPL              ;A=00 IF C, FF IF NC
1C6E 47                         LD   B,A         ;B=LOOP/NO LOOP
1C6F
1C6F 08         NEXTSR1:        EX   AF,AF'      ;A=SGN OF STEP
1C70 A8                         XOR  B           ;REVERSE LOOP/NO LOOP DECISION IF SGN NEWVAL=-VE
1C71 BF                         CP   A           ;SET Z ("INTEGER MATHS DONE")
1C72 C9                         RET              ;RET WITH LOOP/NO LOOP (FF/00)
1C73
1C73 E1         ONERROR:    POP HL              ;NEXT STAT
1C74 CDC63A                 CALL RUNFLG
1C77 D27D0D                 JP NC,STMTLP1       ;CHECK SYNTAX FROM CHAD ONWARDS
1C7A
1C7A E5                     PUSH HL
1C7B DF                     RST 18H
1C7C FEB1                   CP 0B1H             ;STOPTOK
1C7E 2004                   JR NZ,ONERR2
1C80
1C80 E7                     RST 20H             ;SKIP "STOP"
1C81 AF                     XOR A               ;"OFF"
1C82 1816                   JR ONERR3
1C84
1C84 2A455C     ONERR2:     LD HL,(PPC)
1C87 22435B                 LD (ERRLN),HL
1C8A 3A475C                 LD A,(SUBPPC)
1C8D 32425B                 LD (ERRSTAT),A
1C90 E5                     PUSH HL
1C91 CDF210                 CALL SKIPCSTAT
1C94 F1                     POP AF
1C95 3C                     INC A
1C96 2802                   JR Z,ONERR3         ;OFF IF "ON ERROR" USED IN ELINE
1C98
1C98 3E81                   LD A,81H            ;TEMP/PERM BITS ARE "ON" (BITS 7/0)
1C9A                                            ;IF AN ERROR OCCURS NOW, ERRSTAT/LN ARE USED TO
1C9A                                            ;FIND "ON ERROR" AND DO WHAT IT SAYS
1C9A 32455B     ONERR3:     LD (ONERRFLG),A
1C9D C9                     RET
1C9E
1C9E                                            ;RETURN, END PROC, FNDLNHL, IF, ELSE
1C9E                                            ;CALBAS, FOR, NEXT
1C9E
1C9E                    INCLUDE TADJM.SAM       ;KEYSCAN, FPSTACK, SETMIN, SEARCH
1C9E            ;TADJM.SAM
```

```
1C9E
1C9E
1C9E 31004F     NMISTOP:    LD SP,ISPVAL
1CA1 11ED0E                 LD DE,MAINER
1CA4 D5                     PUSH DE
1CA5 ED733D5C               LD (ERRSP),SP
1CA9 CF                     RST 08H
1CAA 0F                     DB 15           ;"BREAK into program"
1CAB
1CAB
1CAB            ;GET KEY FROM BUFFER (ACTUALLY, LASTK). NZ=GOT KEY IN A, ELSE A=0
1CAB
1CAB CDC91C     GETKEY:     CALL KEYRD
1CAE C8                     RET Z           ;RET IF NO KEY - Z
1CAF
1CAF 1812                   JR KBF2
1CB1
1CB1            ;USED BY INKEY$
1CB1
1CB1 F7         READKEY:    RST 30H
1CB2 B8D5                   DW TWOKSC
1CB4 2802                   JR Z,RKY2       ;JR IF GOT CODES IN DE
1CB6
1CB6 AF                     XOR A
1CB7 C9                     RET             ;RET IF NO KEY - Z,NC
1CB8
1CB8 F7         RKY2:       RST 30H
1CB9 9AD5                   DW KYVL         ;USE DE TO GET KEYMAP CODE IN A
1CBB A7                     AND A           ;NZ
1CBC 37                     SCF             ;"GOT KEY"
1CBD
1CBD 210000     KBFLUSH:    LD HL,0
1CC0 22955C                 LD (KBQP),HL    ;EMPTY QUEUE
1CC3
1CC3 213B5C     KBF2:       LD HL,FLAGS
1CC6 CBAE                   RES 5,(HL)      ;"NO KEY"
1CC8 C9                     RET
1CC9
1CC9 F7         KEYRD:      RST 30H
1CCA 10D5                   DW KEYRD2
1CCC 3A3B5C                 LD A,(FLAGS)
1CCF E620                   AND 20H         ;Z IF NO KEY
1CD1 3A085C                 LD A,(LASTK)
1CD4 18ED                   JR KBF2         ;SHOW "NO KEY" IN CASE WE GOT ONE
1CD6
1CD6            ;FPSTACK.SAM - FPCS SUBROUTINES
1CD6
1CD6            ;STACKA - STACK "A" REGISTER ON FPCS. EXIT WITH DE=STKEND
1CD6            ;STACKBC - DITTO WITH BC
1CD6            ;STACKHL - DITTO WITH HL
1CD6
1CD6 55         STACKHL:    LD D,L
1CD7 4C                     LD C,H
1CD8 1805                   JR STACKCM
1CDA
1CDA 0600       STACKA:     LD B,0
1CDC 4F                     LD C,A
1CDD
1CDD 51         STACKBC:    LD D,C
1CDE 48                     LD C,B
1CDF
1CDF AF         STACKCM:    XOR A           ;NC
1CE0 47                     LD B,A
1CE1 5F                     LD E,A
1CE2
1CE2            ;ENTRY FROM TRUNC$
1CE2
1CE2 CDF01C     STKSTOREX:  CALL STKSTORE
1CE5 EB                     EX DE,HL        ;DE=STKEND
1CE6 C9                     RET
1CE7
1CE7            ;STORE A STRING. DE=START, BC=LEN. PAGE IS ASSUMED TO BE SWITCHED IN
1CE7
1CE7 DBFB       STKSTOREP:  IN A,(251)
1CE9
1CE9            ;STACK-STORE
1CE9
1CE9 E67F       STKST0:     AND 7FH         ;BIT 7=0 IF ARRAY OR "SLICED". BITS 4-0=START PAGE
1CEB
1CEB 213B5C     STKSTOS:    LD HL,FLAGS
1CEE CBB6                   RES 6,(HL)      ;STRING RESULT
1CF0
1CF0 2A655C     STKSTORE:   LD HL,(STKEND)
1CF3 77                     LD (HL),A       ;PAGE AND FLAG IF STRING, EXPONENT IF FP NUMBER
1CF4 23                     INC HL
1CF5 73                     LD (HL),E       ;E=SIGN IF SMALL INTEGER
1CF6 23                     INC HL
1CF7 72                     LD (HL),D       ;START IF STRING
1CF8 23                     INC HL
1CF9 71                     LD (HL),C       ;CD=INTEGER IF SMALL INTEGER
1CFA 23                     INC HL
1CFB 70                     LD (HL),B       ;LEN IF STRING
1CFC 23                     INC HL
1CFD 22655C                 LD (STKEND),HL
1D00 C9                     RET
1D01
1D01 2A655C     STKFETCH:   LD HL,(STKEND)
1D04 2B                     DEC HL
1D05 46                     LD B,(HL)
1D06 2B                     DEC HL
1D07 4E                     LD C,(HL)
1D08 2B                     DEC HL
1D09 56                     LD D,(HL)
```

```
1D0A 2B                   DEC HL
1D0B 5E                   LD E,(HL)
1D0C 2B                   DEC HL
1D0D 7E                   LD A,(HL)
1D0E
1D0E 22655C     STSTKE:   LD (STKEND),HL
1D11 C9                   RET
1D12
1D12            ;FASTER DELETE-TOP-OF-FPCS. TO BE CALLED! THIS IS NOT A FPCS FUNCTION!
1D12            ;EXIT: HL PTS TO DELETED NUMBER.
1D12
1D12 2A655C     FDELETE:  LD HL,(STKEND)
1D15 7D                   LD A,L
1D16 D605                 SUB 5
1D18 6F                   LD L,A
1D19 32655C               LD (STKEND),A
1D1C D0                   RET NC            ;ALWAYS RETS ON STANDARD SAM...
1D1D
1D1D 25                   DEC H
1D1E 18EE                 JR STSTKE
1D20
1D20 010500     HLTOFPCS: LD BC,5
1D23 ED5B655C             LD DE,(STKEND)
1D27 EDB0                 LDIR
1D29 ED53655C             LD (STKEND),DE
1D2D C9                   RET
1D2E
1D2E            ;GET FPCS INTEGER TO BC AND HL. A=C. IOOR IF TOO BIG OR -VE
1D2E
1D2E CD3B1D     GETINT:   CALL FPTOBC
1D31 1803                 JR GETIBC
1D33
1D33            ;GET FPCS BYTE TO A AND C. IOOR IF TOO BIG OR -VE
1D33
1D33 CD651D     GETBYTE:  CALL FPTOA
1D36
1D36 3801       GETIBC:   JR C,IOORERR
1D38
1D38 C8                   RET Z             ;RET IF +VE
1D39
1D39 CF         IOORERR:  RST 08H
1D3A 1E                   DB 30             ;"Integer out of range"
1D3B
1D3B            ;COMPRESS TOP OF FPCS TO BC AND HL. CY IF TOO BIG, NZ IF -VE. A=C
1D3B
1D3B 2A655C     FPTOBC:   LD HL,(STKEND)
1D3E 01FBFF               LD BC,-5
1D41 09                   ADD HL,BC
1D42 7E                   LD A,(HL)
1D43 A7                   AND A             ;NC
1D44 2805                 JR Z,FPBCINT
1D46
1D46 EF                   DB CALC
1D47 E0                   DB STKHALF
1D48 01                   DB ADDN
1D49 44                   DB INT            ;CHANGES FORM TO INTEGER IF POSSIBLE
1D4A 33                   DB EXIT
1D4B
1D4B 22655C     FPBCINT:  LD (STKEND),HL    ;"DELETE"
1D4E AF                   XOR A
1D4F 96                   SUB (HL)          ;NC IF SMALL INTEGER FORM (ELSE IOOR)
1D50 23                   INC HL
1D51 CB7E                 BIT 7,(HL)        ;SET NZ IF -VE
1D53 23                   INC HL
1D54 4E                   LD C,(HL)
1D55 23                   INC HL
1D56 46                   LD B,(HL)
1D57 79                   LD A,C
1D58 60                   LD H,B
1D59 69                   LD L,C
1D5A C8                   RET Z             ;RET IF +VE
1D5B
1D5B D8                   RET C             ;RET IF OUT OF RANGE
1D5C
1D5C ED62                 SBC HL,HL         ;HL=0
1D5E ED42                 SBC HL,BC         ;NEGATE BC. RESULT NZ, CY
1D60 3F                   CCF
1D61 44                   LD B,H
1D62 4D                   LD C,L
1D63 79                   LD A,C
1D64 C9                   RET               ;NEGATED RESULT IN HL/BC. C=A. NZ, NC
1D65
1D65            ;COMPRESS TOP OF FPCS TO A (AND C). CY IF TOO BIG, NZ IF -VE.
1D65
1D65 CD3B1D     FPTOA:    CALL FPTOBC
1D68 D8                   RET C
1D69
1D69 08                   EX AF,AF'
1D6A 04                   INC B             ;INC B WITHOUT ALTERING THE FLAGS
1D6B 08                   EX AF,AF'
1D6C 1001                 DJNZ FPTOA2       ;JR IF B<>0 - SIGNAL OUT OF RANGE (CY)
1D6E
1D6E C9                   RET
1D6F
1D6F 37         FPTOA2:   SCF
1D70 C9                   RET
1D71
1D71            ;CALLED BY MAINER, INIT
1D71
1D71 DBFB       SETMIN:   IN A,(URPORT)
1D73 F5                   PUSH AF
1D74 CD351F               CALL ADDRELN
1D77 CDB904               CALL SETKC2       ;SET KCUR
```

```
1D7A 360D                LD (HL),0DH
1D7C 23                  INC HL
1D7D 36FF                LD (HL),0FFH
1D7F 23                  INC HL
1D80 22915A              LD (WORKSP),HL   ;CLEAR ELINE
1D83 32905A              LD (WORKSPP),A
1D86 F1                  POP AF
1D87 D3FB                OUT (URPORT),A
1D89
1D89 2A915A     SETWORK: LD HL,(WORKSP)
1D8C 3A905A              LD A,(WORKSPP)
1D8F 228E5A              LD (WKEND),HL
1D92 328D5A              LD (WKENDP),A    ;CLEAR WORKSPACE
1D95
1D95 2ACC5B     SETSTK:  LD HL,(FPSBOT)
1D98 22655C              LD (STKEND),HL   ;CLEAR FLOATING POINT CALC STACK
1D9B C9                  RET
1D9C
1D9C
1D9C
1D9C            ;SEARCH PROGRAM. CALLED TO FIND DEF FN AND DATA.
1D9C            ;ENTRY: E=TARGET, CHAD PTS TO START
1D9C
1D9C 168D       SRCHPROG: LD D,THENTOK         ;NULL INTERVENING
1D9E
1D9E D9         SEARCHALL: EXX
1D9F 018DFF              LD BC,0FF00H+THENTOK ;ALL PROGRAM/NULL TARGET2 RELOAD
1DA2 D9                  EXX
1DA3
1DA3 018D01     SRCHALL2: LD BC,0100H+THENTOK  ;NO INTERVENING/NULL TARGET2
1DA6
1DA6 DF         SRCHALL3: RST 18H              ;START AT CHAD
1DA7 3A475C              LD A,(SUBPPC)         ;STAT NO FOR A"
1DAA 1802                JR FINDERS
1DAC
1DAC            ;ENTRY AT "FINDER" OR "FINDERS" (WITH A=CURRENT STAT NO)
1DAC            ;B=1 FOR NO INTERVENING TOKENS.
1DAC            ;D=INTERVENING TOKS OR "THEN" FOR NULL, E=TARGET, C=TARGET2 (LESLE OR NULL)
1DAC            ;HL PTS TO START
1DAC            ;C"=TARGET 2 RELOAD, B"=ONE LINE/ALL PROG
1DAC            ;EXIT: CY=FOUND JUST BEFORE HL/(CHAD) AT STAT. A
1DAC
1DAC 08         FINCSTAT: EX AF,AF'            ;INC STATEMENT NO. IN A"
1DAD 3C                  INC A
1DAE
1DAE 08         FINDERS: EX AF,AF'
1DAF 1805                JR FINDER
1DB1
1DB1 23         FSKIP5:  INC HL
1DB2
1DB2 23         FSKIP4:  INC HL
1DB3 23                  INC HL
1DB4 23                  INC HL
1DB5 23                  INC HL
1DB6
1DB6 7E         FINDER:  LD A,(HL)
1DB7 23                  INC HL
1DB8 FE0E                CP 0EH
1DBA 28F5                JR Z,FSKIP5           ;SKIP FP FORMS
1DBC
1DBC FE0D                CP 0DH
1DBE 2832                JR Z,FINDER5          ;JR IF LINE END
1DC0
1DC0 FEB7                CP 0B7H             ;REMTOK
1DC2 2827                JR Z,FREMARK          ;AVOID PROBLEMS FROM EG SINGE QUOTE IN REMS
1DC4
1DC4 FE22                CP 22H                ;QUOTE
1DC6 283D                JR Z,FQUOTE
1DC8
1DC8 FE3A                CP ":"
1DCA 28E0                JR Z,FINCSTAT         ;INC STAT NO
1DCC
1DCC FE8D                CP THENTOK            ;IF COLON OR "THEN"
1DCE 28DC                JR Z,FINCSTAT
1DD0
1DD0 BA                  CP D
1DD1 2839                JR Z,FINTERV          ;INC "INTERVENING" COUNT IN B IF FOUND
1DD3                                           ; E.G. "DO" OR "IF" (USE "THEN" AS NULL)
1DD3                                           ;THEN: JR FINDER4
1DD3
1DD3                                           ;TARGET2 IS NULL UNLESS LELSE/LIF
1DD3                                           ;SEARCH, AND INTERVENING COUNT SHOWS
1DD3                                           ;NO NESTED LIF-ENDIF STRUCTURES (B=1)
1DD3 B9                  CP C
1DD4 2806                JR Z,FOUNDY           ;JR IF TARGET 2 FOUND
1DD6
1DD6 BB         FINDER4: CP E
1DD7 C2B61D              JP NZ,FINDER          ;JR IF NOT TARGET
1DDA
1DDA 1006                DJNZ FOUNDX           ;JR IF B<>0 - DON"T ACCEPT FIND
1DDC
1DDC 22975A     FOUNDY:  LD (CHAD),HL          ;PT CHAD TO TARGET LOCN+1
1DDF 08                  EX AF,AF'             ;A=STAT NO.
1DE0 37                  SCF                   ;"FOUND"
1DE1 C9                  RET
1DE2
1DE2 1004       FOUNDX:  DJNZ FINDN1           ;JR IF COUNT WASN"T 1
1DE4
1DE4 D9                  EXX                   ;OR, IF IT WAS, PRIME TARGET2 TO BE NON-NULL
1DE5 79                  LD A,C                ;FETCH TARGET 2 RELOAD FROM C"
1DE6 D9                  EXX
1DE7 4F                  LD C,A
1DE8
```

```
1DE8 04          FINDN1:    INC B                ;CORRECT INTERVENING COUNTER
1DE9 18CB                   JR FINDER
1DEB
1DEB 3E0D        FREMARK:   LD A,0DH
1DED
1DED BE          FREMLP:    CP (HL)
1DEE 23                     INC HL
1DEF C2ED1D                 JP NZ,FREMLP         ;LOOP PAST REMS TO LINE END
1DF2
1DF2 7E          FINDER5:   LD A,(HL)            ;MSB OF NEXT LINE NUMBER, OR FF TERMINATOR
1DF3 D9                     EXX
1DF4 B8                     CP B                 ;B"=00 FOR 1 LINE SEARCH OR FF FOR WHOLE PROG
1DF5 D9                     EXX                  ;E-LINE NEEDS AN FF TERMINATOR
1DF6 D0                     RET NC               ;RET IF FINISHED
1DF7
1DF7 CB74                   BIT 6,H
1DF9 C4F23F                 CALL NZ,INCURPAGE    ;INC UPPER RAM PAGE, ADJ HL, IF HL>BFFF
1DFC 22AF5A                 LD (CLA),HL          ;CURRENT LINE ADDR
1DFF 3E01                   LD A,1               ;STATEMENT 1
1E01 08                     EX AF,AF'            ;IN A"
1E02 C3B21D                 JP FSKIP4            ;DO NEXT LINE
1E05
1E05 BE          FQUOTE:    CP (HL)
1E06 23                     INC HL
1E07 C2051E                 JP NZ,FQUOTE         ;LOOP PAST LITERAL STRINGS
1E0A
1E0A 18AA                   JR FINDER            ;HL IS PAST END QUOTE
1E0C
1E0C 04          FINTERV:   INC B                ;INC COUNT OF TARGETS TO SKIP
1E0D 0E8D                   LD C,THENTOK         ;TARGET2=NULL
1E0F 18C5                   JR FINDER4
1E11
1E11             ;MAKE ROOM, ALLOWING NO OVERHEAD
1E11
1E11 AF          MKRMCH:    XOR A
1E12 E5                     PUSH HL
1E13 CD143F                 CALL TSTRMBIG
1E16 1810                   JR MKRM2
1E18
1E18             ;OPEN 1 BYTE AT HL
1E18
1E18 010100      MKRM1:     LD BC,1
1E1B
1E1B             ;OPEN BC BYTES AT HL. BC MUST BE <4000H
1E1B
1E1B AF          MAKEROOM:  XOR A
1E1C
1E1C             ;OPEN ABC BYTES AT HL. A=16K PAGES, BC=MOD 16K
1E1C
1E1C E5          MKRBIG:    PUSH HL              ;LOCN
1E1D CD143F                 CALL TSTRMBIG
1E20 219600                 LD HL,150
1E23 ED52                   SBC HL,DE
1E25 D24B3F                 JP NC,OOMERR         ;INSIST ON A 150-BYTE OVERHEAD
1E28
1E28 50          MKRM2:     LD D,B
1E29 59                     LD E,C
1E2A 4F                     LD C,A
1E2B 7A                     LD A,D
1E2C E63F                   AND 3FH
1E2E 57                     LD D,A               ;CDE=ROOM. NC (SIGNALS "MAKEROOM")
1E2F E1                     POP HL               ;LOCN
1E30 D5                     PUSH DE              ;MOD 16K
1E31 E5                     PUSH HL              ;LOCN
1E32 F7                     RST 30H
1E33 92E7                   DW XOINTERS          ;EXIT WITH AHL=OLD WORKEND (SRC), MOD/PAGCOUNT SET
1E35 ED5B8E5A               LD DE,(WKEND)
1E39 ED4B8D5A               LD BC,(WKENDP)       ;CDE=DEST (NEW WKEND)
1E3D CD4E2A                 CALL FARLDDR
1E40 D1                     POP DE               ;LOCN
1E41 E1                     POP HL               ;MOD 16K
1E42 19                     ADD HL,DE
1E43 EB                     EX DE,HL             ;HL=LOCN
1E44 1B                     DEC DE               ;DE=END (IF <16K MADE)
1E45 C9                     RET
1E46
1E46 CD4D1A      FNORECL:   CALL FNDLINE
1E49 C0                     RET NZ
1E4A
1E4A CD4E1F      NORECL:    CALL NEXTONE
1E4D 1803                   JR RECLAIM2      ;DELETE LINE FROM PROGRAM
1E4F
1E4F             ;ENTRY WITH DE=LOCN, HL=END, DE PAGED IN
1E4F
1E4F CD571F      RECLAIM1:  CALL DIFFER          ;GET BC, SWOP LOCN TO HL
1E52
1E52             ;ENTRY WITH BC=BYTES TO RECLAIM AT HL (<16K)
1E52
1E52 AF          RECLAIM2:  XOR A
1E53
1E53             ;RECLAIM ABC BYTES AT HL. A=16K PAGES, BC=MOD 16K
1E53
1E53 CBB8        RECL2BIG:  RES 7,B
1E55 CBB0                   RES 6,B
1E57 57                     LD D,A
1E58 B0                     OR B
1E59 B1                     OR C
1E5A C8                     RET Z
1E5B
1E5B 7A                     LD A,D
1E5C 50                     LD D,B
1E5D 59                     LD E,C
1E5E 4F                     LD C,A               ;CDE=SPACE
```

```
1E5F C5                     PUSH BC             ;PAGES
1E60 D5                     PUSH DE             ;MOD 16K
1E61 E5                     PUSH HL             ;LOCN
1E62 37                     SCF                 ;"RECLAIMING"
1E63 F7                     RST 30H
1E64 92E7                   DW XOINTERS         ;SETS PAG/MODCOUNT TO MOVE
1E66 E1                     POP HL              ;LOCN
1E67 D1                     POP DE              ;MOD 16K
1E68 C1                     POP BC              ;PAGES
1E69 DBFB                   IN A,(251)
1E6B F5                     PUSH AF
1E6C E5                     PUSH HL             ;LOCN
1E6D CB74                   BIT 6,H
1E6F 2803                   JR Z,RECL5
1E71
1E71 CBB4                   RES 6,H
1E73 3C                     INC A
1E74
1E74 CDDE1F     RECL5:      CALL ADDAHLCDE      ;AHL=SRC
1E77 D1                     POP DE
1E78 C1                     POP BC
1E79 48                     LD C,B              ;CDE=DEST (LOCN)
1E7A D5                     PUSH DE
1E7B CD5E2A                 CALL FARLDIR
1E7E E1                     POP HL
1E7F C9                     RET
1E80
1E80
1E80            ;MAKE ROOM AT WKSPACE END, BC BYTES LONG. EXIT WITH BC=ROOM SIZE, DE=START,
1E80            ;HL=END (IF ROOM <16K), A=UNCHANGED. ROOM PAGE SWITCHED IN.
1E80
1E80 F5         WKROOM:         PUSH AF
1E81 CD243F                     CALL TESTROOM       ;CHECK BC BYTES OK, GET AHL=NEW WKEND
1E84 57                         LD D,A
1E85 3A8D5A                     LD A,(WKENDP)
1E88 CDDF3F                     CALL SELURPG        ;SWITCH IN OLD WKEND
1E8B 7A                         LD A,D
1E8C 328D5A                     LD (WKENDP),A       ;NEW WKENDP
1E8F ED5B8E5A                   LD DE,(WKEND)       ;START OF ROOM
1E93 228E5A                     LD (WKEND),HL
1E96 62                         LD H,D
1E97 6B                         LD L,E
1E98 09                         ADD HL,BC
1E99 2B                         DEC HL              ;END OF ROOM, IF ROOM<16K (HL MAY BE >C000)
1E9A F1                         POP AF
1E9B C9                         RET
1E9C
1E9C
1E9C            ;ADJUST SINGLE "SYS VAR" (ACTUALLY, FOR-NEXT OR DO/GOSUB/PROC ADDR)
1E9C            ;ENTRY: HL POINTS TO SVAR, CDE=LOCN, F'=CY IF RECLAIMING
1E9C
1E9C E5         ASSV:       PUSH HL
1E9D FDE1                   POP IY
1E9F 0601                   LD B,1
1EA1
1EA1 FD7E00     PNLP:       LD A,(IY+0)
1EA4 FD6E01                 LD L,(IY+1)
1EA7 FD6602                 LD H,(IY+2)         ;AHL=SVAR
1EAA 24                     INC H
1EAB 25                     DEC H
1EAC 2837                   JR Z,NPSV           ;no adj if e.g. eline addr v2.1
1EAE
1EAE            ;COMPARE AHL AND CDE
1EAE
1EAE CB74                   BIT 6,H
1EB0 2803                   JR Z,PNT2           ;JR IF IN SECTION C
1EB2
1EB2 3C                     INC A
1EB3 CBB4                   RES 6,H             ;ELSE ADJUST
1EB5
1EB5 E61F       PNT2:       AND 1FH
1EB7 B9                     CP C
1EB8 382B                   JR C,NPSV           ;JR IF LOCN PAGE IS HIGHER, SO NO ADJ
1EBA
1EBA 2007                   JR NZ,PADJ          ;JR IF LOCN PAGE IS LOWER - ADJUST
1EBC
1EBC EB                     EX DE,HL
1EBD ED52                   SBC HL,DE           ;ELSE COMPARE OFFSETS - SBC LOCN,SVAR
1EBF 19                     ADD HL,DE
1EC0 EB                     EX DE,HL
1EC1 3022                   JR NC,NPSV          ;DO NOT ADJUST SVAR IF IT IS <= LOCN
1EC3
1EC3 C5         PADJ:       PUSH BC             ;SAVE PTR COUNT AND LOCN PAGE
1EC4 D5                     PUSH DE
1EC5 ED4B4D5B               LD BC,(TEMPW4)
1EC9 ED5B4F5B               LD DE,(TEMPW5)      ;CDE=AMOUNT TO ADJ BY
1ECD 08                     EX AF,AF'
1ECE 3806                   JR C,PRECL          ;JR IF RECLAIMING
1ED0
1ED0 08                     EX AF,AF'
1ED1 CDDE1F                 CALL ADDAHLCDE
1ED4 1804                   JR PNT3
1ED6
1ED6 08         PRECL:      EX AF,AF'
1ED7 CDE71F                 CALL SUBAHLCDE
1EDA
1EDA D1         PNT3:       POP DE
1EDB C1                     POP BC
1EDC FD7700                 LD (IY+0),A
1EDF FD7501                 LD (IY+1),L
1EE2 FD7402                 LD (IY+2),H         ;PLACE ADJUSTED SYS VAR
1EE5
1EE5 FD23       NPSV:       INC IY
```

```
1EE7 FD23                  INC IY
1EE9 FD23                  INC IY
1EEB 10B4                  DJNZ PNLP
1EED
1EED C9                    RET
1EEE
1EEE             ;ADJUST 'FOR' LOOPS IF  LOCN C'D'E' IS BEFORE THEM
1EEE
1EEE 08          AFLPS:    EX AF,AF'
1EEF 4A                    LD C,D
1EF0 061A                  LD B,26           ;26 LETTER LISTS
1EF2
1EF2 E5          AFML:     PUSH HL           ;PT TO CURRENT LETTER LIST
1EF3
1EF3 5E          AFLL:     LD E,(HL)
1EF4 23                    INC HL
1EF5 56                    LD D,(HL)
1EF6 19                    ADD HL,DE
1EF7 381B                  JR C,AFLE         ;JR IF THIS LIST ENDED
1EF9
1EF9 CB76                  BIT 6,(HL)
1EFB 2814                  JR Z,AFNF         ;JR IF NOT FOR-VAR
1EFD
1EFD E5                    PUSH HL
1EFE C5                    PUSH BC
1EFF 7E                    LD A,(HL)
1F00 E61F                  AND 1FH           ;NAME LEN-1
1F02 C612                  ADD A,18          ;3 TO SKIP TO START OF VALUE, 15 FOR VLS
1F04 5F                    LD E,A
1F05 1600                  LD D,0
1F07 19                    ADD HL,DE         ;PT TO ADDR
1F08 ED5BCC5A              LD DE,(TEMPW3)    ;CDE=LOCN
1F0C CD9C1E                CALL ASSV         ;ADJUST SINGLE "SYSTEM VAR"
1F0F C1                    POP BC
1F10 E1                    POP HL
1F11
1F11 23          AFNF:     INC HL
1F12 18DF                  JR AFLL           ;NEXT VAR OF THIS LETTER
1F14
1F14 E1          AFLE:     POP HL            ;THIS LIST ENDED NOW
1F15 23                    INC HL
1F16 23                    INC HL            ;PT TO NEXT LIST
1F17 10D9                  DJNZ AFML
1F19
1F19 08                    EX AF,AF'
1F1A C9                    RET
1F1B
1F1B
1F1B 2EA5        ADDRDEST:     LD L,>(DEST-1)
1F1D 181F                      JR ADDRSV
1F1F
1F1F 2E87        ADDRNV:       LD L,>(NVARS-1)
1F21 181B                      JR ADDRSV
1F23
1F23 2E84        ADDRNE:       LD L,>(NUMEND-1)
1F25 1817                      JR ADDRSV
1F27
1F27 2E81        ADDRSAV:      LD L,>(SAVARS-1)
1F29 1813                      JR ADDRSV
1F2B
1F2B 2E90        ADDRWK:       LD L,>(WORKSP-1)
1F2D 180F                      JR ADDRSV
1F2F
1F2F 3E99        ADDRKC:       LD A,>(KCUR-1)
1F31 21                        DB 21H
1F32
1F32 3E9F        ADDRPROG:     LD A,>(PROG-1)
1F34 21                        DB 21H
1F35
1F35 3E93        ADDRELN:      LD A,>(ELINE-1)
1F37 21                        DB 21H          ;"JR+2"
1F38
1F38             ;USED BY READ AND ITEM
1F38
1F38 3E8A        ADDRDATA:     LD A,>(DATADD-1)
1F3A 21                        DB 21H          ;"JR+2"
1F3B
1F3B 3E96        ADDRCHAD:     LD A,>(CHAD-1)
1F3D
1F3D 6F                        LD L,A
1F3E
1F3E             ;LOOK AT SYS VAR LOCN. ENTRY: HL PTS TO SYS VAR (PAGE, OFFSET)
1F3E             ;EXIT: PAGE SELECTED, HL=OFFSET (IN SECTION C), A=PAGE
1F3E
1F3E 265A        ADDRSV:       LD H,VAR2/256
1F40
1F40             ;FROM FN, WITH HL=DEFADDRP
1F40
1F40 7E          ASV2:         LD A,(HL)
1F41 CDDF3F                    CALL SELURPG
1F44 23                        INC HL
1F45 7E                        LD A,(HL)
1F46 23                        INC HL
1F47 66                        LD H,(HL)
1F48 6F                        LD L,A
1F49 DBFB                      IN A,(251)
1F4B E61F                      AND 1FH
1F4D C9                        RET
1F4E
1F4E
1F4E E5          NEXTONE:  PUSH HL
1F4F 23                    INC HL
1F50 23                    INC HL
```

```
1F51 4E                     LD C,(HL)
1F52 23                     INC HL
1F53 46                     LD B,(HL)         ;LINE LEN
1F54 23                     INC HL
1F55 09                     ADD HL,BC
1F56 D1                     POP DE            ;DE=OLD START, HL=NEXT LINE START
1F57
1F57 A7           DIFFER:   AND A
1F58 ED52                   SBC HL,DE
1F5A 44                     LD B,H
1F5B 4D                     LD C,L
1F5C 19                     ADD HL,DE
1F5D EB                     EX DE,HL
1F5E C9                     RET
1F5F
1F5F
1F5F              ;ENTRY AS LIMBYTE, BUT FPCS VALUE DECED BEFORE CHECKING, RETURNED DECED. C IS
1F5F              ;NOT DECED
1F5F
1F5F 3EFF         LIMDB:    LD A,0FFH
1F61 FE                     DB 0FEH           ;"JR+1"
1F62
1F62              ;ENTRY: D=LIMIT (VALUE MUST BE BELOW IT), E=ERROR TO GIVE IF VALUE TOO HIGH.
1F62              ;VALUE IS ON FPCS
1F62              ;EXIT: A AND C=VALUE, DE=ORIG
1F62
1F62 AF           LIMBYTE:  XOR A
1F63 F5                     PUSH AF
1F64 D5                     PUSH DE
1F65 CD651D                 CALL FPTOA
1F68 D1                     POP DE            ;LIMIT/ERROR
1F69 3806                   JR C,ERRORE       ;JR IF >FF
1F6B
1F6B 2004                   JR NZ,ERRORE      ;JR IF -VE
1F6D
1F6D F1                     POP AF
1F6E 81                     ADD A,C
1F6F BA                     CP D
1F70 D8                     RET C
1F71
1F71 213A5C       ERRORE:   LD HL,ERRNR
1F74 73                     LD (HL),E
1F75 E5                     PUSH HL           ;WILL BE POPPED AND USED AS "RET ADDR"
1F76 C30800                 JP 0008H
1F79
1F79              ;SET PAGCOUNT/MODCOUNT FROM BC (0000-FFFF)
1F79
1F79 F5           SPLITBC:      PUSH AF
1F7A 78                         LD A,B
1F7B CBB8                       RES 7,B
1F7D CBB0                       RES 6,B
1F7F ED43845B                   LD (MODCOUNT),BC  ;LEN MOD 16K
1F83 07                         RLCA
1F84 07                         RLCA
1F85 E603                       AND 03H
1F87 32835B                     LD (PAGCOUNT),A   ;PAGES (0-3)
1F8A F1                         POP AF
1F8B C9                         RET
1F8C
1F8C              ;GET ROOM. RETURN FREE MEMORY (UPPER RAMS) IN AHL AS A 19 BIT NO. NZ IF >=64K
1F8C
1F8C C5           GETROOM:      PUSH BC
1F8D D5                         PUSH DE
1F8E CD1620                     CALL WENORMAL
1F91
1F91 4F           GRM2:         LD C,A
1F92 EB                         EX DE,HL          ;CDE=WKSPACE END
1F93 CD0E20                     CALL RTNORMAL     ;AHL=RAMTOP (19 BIT)
1F96 37                         SCF
1F97 ED52                       SBC HL,DE
1F99 99                         SBC A,C
1F9A D1                         POP DE
1F9B C1                         POP BC
1F9C C9                         RET               ;AHL=ROOM (19BIT). NZ IF ROOM >=64K
1F9D
1F9D
1F9D              ;PAGE OVERFLOW. USED BY INSTRING, S16OP
1F9D              ;CALLED IF ADD HL,RR GIVES CARRY. CORRECTS THE PAGE AND ADDRESS IN HL
1F9D              ;ENTRY: HL=0000-BFFE (BFFE IF HL WAS BFFF+FFFF)
1F9D
1F9D DBFB         PGOVERF:      IN A,(URPORT)
1F9F CDA41F                     CALL PGOA
1FA2 1825                       JR PGOE
1FA4
1FA4              ;CORRECT AHL AFTER PAGE OVERFLOW, BUT DON"T ALTER PAGING
1FA4
1FA4 C602         PGOA:         ADD A,2             ;IF THERE WAS A CARRY, ADDR IS AT LEAST 0000
1FA6                                                ;SO INC 2 PAGES SO ADDR CAN BE DROPPED 32K
1FA6 CB74                       BIT 6,H             ;IF BIT 6 IS HIGH WE CAN INC AGAIN
1FA8 2803                       JR Z,PGOA2
1FAA
1FAA CBB4                       RES 6,H
1FAC 3C                         INC A
1FAD
1FAD CB7C         PGOA2:        BIT 7,H
1FAF 2802                       JR Z,PGOA3          ;IF BIT 7 IS HIGH INC TWICE
1FB1
1FB1 C602                       ADD A,2
1FB3
1FB3 CBFC         PGOA3:        SET 7,H
1FB5 C9                         RET
1FB6
1FB6              ;ADDRESS ELINE AND DEC PTR TO PT TO SAVARS END
```

```
1FB6
1FB6 CD351F     ADDRELND:  CALL ADDRELN
1FB9
1FB9            ;ENTRY: HL HOLDS AN ADDRESS. IT IS DECREMENTED, AND ADJUSTED IF IT FALLS
1FB9            ;TOO LOW, SO IT PTS TO 8000-BFFF, PAGING AS NEEDED. IF CALLED FROM SYNTAX4,
1FB9            ;HL MAY BE BELOW 8000H ALREADY. COPES WITH UNDERFLOW OF UP TO 32K
1FB9
1FB9 2B         DECPTR:    DEC HL
1FBA
1FBA CB7C       CHKPTR:    BIT 7,H
1FBC C0                    RET NZ            ;RET IF STILL AT 8000H OR MORE
1FBD
1FBD DBFB                  IN A,(251)
1FBF 3D                    DEC A
1FC0 CBFC                  SET 7,H
1FC2 CB74                  BIT 6,H
1FC4 2001                  JR NZ,DECPT2      ;JR IF FALLEN INTO 4000-7FFF - CORRECT BY 1 PAGE
1FC6
1FC6 3D                    DEC A             ;IF IN 0000-3FFF, 2 PAGES
1FC7
1FC7 CBB4       DECPT2:    RES 6,H
1FC9
1FC9 D3FB       PGOE:      OUT (251),A
1FCB C9                    RET
1FCC
1FCC            ;ADD ADDR,BC. ADD BC TO AN ADDRESS IN AHL. A=PAGE, HL=8000-BFFF. BC UNCHNGED.
1FCC            ;BC CAN HAVE ANY VALUE
1FCC            ;SUB ADDR,BC. DITTO
1FCC            ;CARRY IF OVERFLOW
1FCC
1FCC CD2120     ADDAHLBC:     CALL AHLNORM     ;GET 19 BIT FORM IN AHL
1FCF 09                       ADD HL,BC
1FD0 CE00                     ADC A,0          ;ADD AHL,BC
1FD2 181E                     JR PAGEFORM
1FD4
1FD4 CD2120     SUBAHLBC:     CALL AHLNORM
1FD7 A7                       AND A
1FD8 ED42                     SBC HL,BC
1FDA DE00                     SBC A,0          ;SUB AHL,BC
1FDC 1814                     JR PAGEFORM
1FDE
1FDE            ;ADD ADDR IN AHL (PAGE FORM) TO THE ADDRESS IN CDE (PAGE FORM)
1FDE            ;RESULT IN AHL (PAGE FORM). CARRY IF OVERFLOW PAGES. CDE UNCHANGED.
1FDE
1FDE C5         ADDAHLCDE:    PUSH BC
1FDF D5                       PUSH DE
1FE0 CD0120                   CALL TWOCONV
1FE3 19                       ADD HL,DE
1FE4 89                       ADC A,C
1FE5 1809                     JR PPFCOM
1FE7
1FE7            ;SUBTRACT FROM ADDR IN AHL (PAGE FORM) THE ADDRESS IN CDE (PAGE FORM)
1FE7            ;RESULT IN AHL (PAGE FORM). CARRY IF OVERFLOW. CDE UNCHANGED.
1FE7
1FE7 C5         SUBAHLCDE:    PUSH BC
1FE8 D5                       PUSH DE
1FE9 CD0120                   CALL TWOCONV
1FEC A7                       AND A
1FED ED52                     SBC HL,DE
1FEF 99                       SBC A,C          ;SUB AHL,CDE
1FF0
1FF0 D1         PPFCOM:    POP DE            ;CONTINUE INTO PAGEFORM
1FF1 C1                    POP BC
1FF2
```

```
1FF2            ;TRANSFORM 19-BIT NUMBER IN AHL TO PAGE, ADDR (8000-BFFF)
1FF2
1FF2 CB14       PAGEFORM:  RL H
1FF4 17                    RLA
1FF5 CB14                  RL H
1FF7 17                    RLA               ;NC. PAGE NOW OK
1FF8 CB1C                  RR H
1FFA 37                    SCF
1FFB CB1C                  RR H              ;ADDR NOW OK IN 8000-BFFF FORM
1FFD FE20                  CP 20H
1FFF 3F                    CCF               ;SET CARRY IF OVERFLOW
2000 C9                    RET
2001
2001            ;CONVERT PAGE FORMS IN AHL AND CDE TO 19-BIT. USES B
2001
2001 CD2120     TWOCONV:   CALL AHLNORM
2004
2004            ;CONVERT PAGE FORM IN CDE TO 19-BIT IN CDE. CHANGES CDE ONLY
2004
2004 F5         CDENORM:   PUSH AF
2005 EB                    EX DE,HL
2006 79                    LD A,C
2007 CD2120                CALL AHLNORM
200A EB                    EX DE,HL
200B 4F                    LD C,A
200C F1                    POP AF
200D C9                    RET
200E
200E 3AB15C     RTNORMAL:  LD A,(RAMTOPP)
2011 2AB25C                LD HL,(RAMTOP)
2014 180B                  JR AHLNORM
2016
2016 3A8D5A     WENORMAL:  LD A,(WKENDP)
2019 2A8E5A                LD HL,(WKEND)
```

```
201C CB74                  BIT 6,H
201E 2801                  JR Z,AHLNORM
2020
2020 3C                    INC A
2021
2021               ;CONVERT PAGE FORM IN AHL TO 19-BIT IN AHL.
2021               ;TOP 3 BITS OF ORIG A AND TOP 2 BITS OF HL ARE IRREL.
2021
2021 CB04          AHLNORM:   RLC H
2023 CB04                     RLC H
2025 1F                       RRA
2026 CB1C                     RR H
2028 1F                       RRA
2029 CB1C                     RR H
202B E607                     AND 07H
202D C9                       RET
202E
202E
202E               ;SET ERROR STACK PTR. EXITS WITH OLD ERRSP ON STACK. AFTER A ROUTINE HAS CALLED
202E               ;HERE, IT CAN CALL OTHER SRS. AND RETURN WILL STILL OCCUR EVEN AFTER AN ERROR.
202E               ;USES HL ONLY
202E
202E 2A3D5C        SETESP:    LD HL,(ERRSP)
2031 E3                       EX (SP),HL
2032 E5                       PUSH HL
2033 ED733D5C                 LD (ERRSP),SP
2037 C9                       RET
2038
2038
2038               ;READ LEN FROM "DESIRED" HEADER (TAPE)
2038
2038 2E22          RDRLEN:    LD L,>HDR+HDN+3
203A 11                       DB 11H               ;"JR+2"
203B
203B               ;READ LEN FROM LOADED HEADER
203B
203B 2E72          RDLLEN:    LD L,>HDL+HDN+3
203D
203D 264B                     LD H,<HDR
203F
203F               ;READ THREE BYTES FROM HL INTO C AND DE
203F
203F 4E            RDTHREE:   LD C,(HL)
2040 23                       INC HL
2041 5E                       LD E,(HL)
2042 23                       INC HL
2043 56                       LD D,(HL)
2044 C9                       RET
2045
2045 0E00          EDGE2:     LD C,0               ;INIT LOOP COUNTER TO 0
2047
2047 CD4C20        EDGEC:     CALL EDGSENS         ;FIND AN EDGE
204A D0                       RET NC               ;RET IF BREAK OR TIMEOUT
204B
204B A7                       AND A                ;NC
204C
204C 3E08          EDGSENS:   LD A,8               ;DELAY TO AVOID DOUBLE COUNTS, RELATED TO SPEED
204E
204E 3D            EWL:       DEC A
204F 20FD                     JR NZ,EWL
2051
2051 DBFE          EDGLP:     IN A,(KEYPORT)
2053 0C                       INC C
2054 C8                       RET Z                ;TIMEOUT - NC, Z
2055
2055 A8                       XOR B                ;CP EAR BIT
2056 E640                     AND 40H              ;WITH LAST EDGE TYPE
2058 78                       LD A,B
2059 28F6                     JR Z,EDGLP           ;LOOP UNTIL SIGNAL LEVEL CHANGES. 47 Ts PER LOOP
205B                                               ;(LENGTHENED TO NEAR MULTIPLE OF 8)
205B EE67                     XOR 67H
205D 47                       LD B,A               ;FLIP "LAST EDGE TYPE" (BIT 6)
205E                                               ;FLIP          BORDER  (BITS 2-0)
205E                                               ;KEEP MIC (BIT 3) UNCHANGED (HI)
205E                                               ;KEEP BEEP (BIT 4) UNCHANGED
205E                                               ;KEEP SOF (BIT 7) UNCHANGED
205E
205E E61F                     AND 1FH              ;SOFF BIT (BIT 7) LOW
2060                                               ;THRO MIDI (BIT 6) LOW
2060                                               ;BORDER MSB (BIT 5) LOW
2060
2060 D3FE                     OUT (KEYPORT),A      ;ALTER BORDER
2062 3EF7                     LD A,0F7H
2064 DBF9                     IN A,(STATPORT)      ;BIT 5 IS LOW IF ESC PRESSED
2066 07                       RLCA
2067 07                       RLCA
2068 07                       RLCA
2069 79                       LD A,C               ;A AND C HOLD LENGTH OF PULSE
206A C9                       RET                  ;CY="OK", NC=ESC.
206B
206B
206B                          INCLUDE GRAPH0.SAM     ;CIRCLE, DRAW
206B
206B               ;GRAPH0.SAM
206B               ;**************************************************************************
206B               ;
206B
206B               ;ENTRY FROM JUMP TABLE WITH A=RADIUS, C,B=COORDS, HL=OFFSET IF THINPIX
206B
206B F5            JCIRCLE:   PUSH AF
206C 22C85A                   LD (TEMPW1),HL
206F 1812                     JR JCIRC3
2071
2071 CDF23A        CIRCLE:    CALL SYNTAX9         ;GET COLOURS, X,Y
```

```
2074 CD853A                CALL INSISCOMA    ;REQUIRE COMMA
2077 CD583A                CALL SYNTAX6      ;RADIUS (X,Y ALREADY ON FPCS)
207A
207A CD331D                CALL GETBYTE      ;COMPRESS RADIUS INTO A AND C REGS.
207D
207D             CIRCLEFD:                   ;RETURN TO CIRCEXT AFTER PLOT
207D C5                    PUSH BC           ;RAD IN C
207E F5                    PUSH AF           ;RAD
207F F7                    RST 30H
2080 C8E9                  DW CIFILSR        ;Y COORD IN B, X IN C, POSSIBLE OFFSET IN TEMPW1
2082
2082 F1                    POP AF
2083
2083 DD21FC20    JCIRC3:   LD IX,CIRCEXT     ;RET SEE ABOVE
2087 57                    LD D,A            ;RADIUS
2088 CB3F                  SRL A
208A 08                    EX AF,AF'         ;RAD/2 PROTECTED TO A"
208B 1E03                  LD E,3            ;BLITZ CODE FOR "CIRCLE"
208D 3A4D5A                LD A,(THFATT)
2090 A7                    AND A
2091 C46628                CALL NZ,GRAREC    ;CALL IF FAT
2094 CDB13F                CALL SPSS         ;STORE PAGE, SELECT SCREEN
2097 CDB423                CALL SETIY        ;SET IY TO PLOT ROUTINE
209A 3004                  JR NC,CIRCTK
209C
209C FD21FF20              LD IY,THCIRCSR
20A0
20A0 D9          CIRCTK:   EXX
20A1 57                    LD D,A            ;D"=M3 PLOT INK (OR JUNK IF NOT M3)
20A2 D9                    EXX
20A3
20A3 50                    LD D,B
20A4 59                    LD E,C            ;DE=Y/X
20A5 E1                    POP HL            ;L=RADIUS
20A6 2C                    INC L
20A7 2D                    DEC L
20A8 2007                  JR NZ,DOCIRCLE    ;JR IF RADIUS NON-ZERO
20AA
20AA EB                    EX DE,HL          ;HL=COORDS OF CENTRE
20AB DD21C63F              LD IX,TRCURP
20AF FDE9                  JP (IY)           ;PLOT POINT, THEN GOTO TRCURP
20B1
20B1
20B1 2600        DOCIRCLE: LD H,0
20B3
20B3 CDE820                CALL CIRCP5       ;FOLLOWING SECTION DEALS WITH FIRST 4. R & L..
20B6 7D                    LD A,L
20B7 6C                    LD L,H
20B8 67                    LD H,A
20B9 CDED20                CALL CIRC6        ;TOP
20BC E5                    PUSH HL
20BD AF                    XOR A
20BE 94                    SUB H
20BF 67                    LD H,A
20C0 CDED20                CALL CIRC6        ;BOTTOM
20C3 E1                    POP HL
20C4 2C                    INC L             ;START WITH DISP OF 1. OTHERWISE FIRST 4 POINTS ARE
20C5                                         ;DONE TWICE (0 ABOVE AND BELOW AXIS ARE THE SAME)
20C5                                         ;WHICH MAKES GAPS WITH OVER 1
20C5
20C5             ;MAIN PLOT A POINT ON EACH SEGMENT LOOP
20C5 CDDC20      CIRC0:    CALL CIRCEX
20C8
20C8 2C                    INC L
20C9 08                    EX AF,AF'
20CA 95                    SUB L
20CB 3002                  JR NC,CIRC2
20CD
20CD 84                    ADD A,H
20CE 25                    DEC H
20CF
20CF 08          CIRC2:    EX AF,AF'
20D0 7D                    LD A,L
20D1 94                    SUB H
20D2 DAC520                JP C,CIRC0
20D5
20D5 3D                    DEC A
20D6 C4E220                CALL NZ,CIRCP4    ;IF ARCS HAVE NOT MET YET, DO 4 POINTS.
20D9
20D9 C3C63F                JP TRCURP         ;CALL TEMPS, RESET PAGE
20DC
20DC             ;DO 8 POINTS
20DC CDDF20      CIRCEX:   CALL CIRC3        ;CALL DO 4 POINTS, THEN ENTER IT AGAIN...
20DF
20DF             ;DO 4 POINTS
20DF 7D          CIRC3:    LD A,L
20E0 6C                    LD L,H
20E1 67                    LD H,A            ;SWOP DISPS
20E2
20E2 CDE520      CIRCP4:   CALL CIRC4        ;CALL DO 2 POINTS, THEN ENTER IT AGAIN...
20E5
20E5             ;DO 2 POINTS
20E5 AF          CIRC4:    XOR A
20E6 94                    SUB H
20E7 67                    LD H,A            ;NEGATE H FOR BOTTOM HALF
20E8
20E8 7B          CIRCP5:   LD A,E            ;X COORD
20E9 85                    ADD A,L
20EA D4F020                CALL NC,CIRC5     ;DO RHS UNLESS WRAPPED ROUND (DO 1 POINT)
20ED
20ED 7B          CIRC6:    LD A,E
20EE 95                    SUB L
20EF D8                    RET C             ;RET IF WRAPPED, ELSE DO LHS
```

```
20F0
20F0                ;DO 1 POINT
20F0 44         CIRC5:      LD B,H
20F1 4D                     LD C,L              ;SAVE DISPS
20F2 6F                     LD L,A              ;X
20F3 7A                     LD A,D
20F4 84                     ADD A,H
20F5 FEC0                   CP 192
20F7 3003                   JR NC,CIRCEXT
20F9
20F9 67                     LD H,A
20FA
20FA FDE9       CIPLOT:     JP (IY)             ;PLOT HL, RETURN TO CIRCEXT
20FC
20FC 60         CIRCEXT:    LD H,B
20FD 69                     LD L,C              ;DISPS RESTORED
20FE C9                     RET
20FF
20FF
20FF                ;ENTRY: H=Y (RANGE OK), C=X
20FF
20FF DDE5       THCIRCSR:   PUSH IX
2101 D5                     PUSH DE
2102 C5                     PUSH BC
2103 44                     LD B,H              ;B=Y
2104 EB                     EX DE,HL            ;E=X
2105 1600                   LD D,0
2107 2AC85A                 LD HL,(TEMPW1)      ;OFFSET OF ZERO IF CIRCLE CENTRE X IS <80H
210A                                            ;OFFSET OF 0100 IF CIRCLE CENTRE X IS >017F
210A                                            ;OFFSET 0-FF OTHERWISE.
210A                                            ;E OF 0-FF+OFFSET 0-0100 IS ALWAYS IN RANGE
210A
210A 19                     ADD HL,DE           ;REAL X
210B CDE322                 CALL M2CTPLOT       ;PLOT HL,B WITHOUT ALTERING COORDS SYS VARS
210E C1                     POP BC
210F D1                     POP DE
2110 DDE1                   POP IX
2112 DDE9                   JP (IX)             ;TO CIRCEXT OR TEMPS
2114
2114                ;*************************************************************************
2114                ;DRAW.SAM. SAM DRAW COMMAND.
2114
2114                ;FROM JUMP TABLE WITH X,Y IN C,B OR HL,B
2114
2114 3A4D5A     JDRAWTO:        LD A,(THFATT)
2117 FE01                       CP 1
2119 08                         EX AF,AF'           ;CY IF THIN DRAW
211A 183B                       JR JDRTO3
211C
211C D68E       DRAW:           SUB TOTOK           ;VAR IS ZERO IF "DRAW TO" USED
211E 32D05A                     LD (TEMPB3),A
2121 2001                       JR NZ,PASTTO
2123
2123 E7                         RST 20H             ;SKIP "TO"
2124
2124 CDF23A     PASTTO:         CALL SYNTAX9
2127 FE2C                       CP ","
2129 2013                       JR NZ,DRNOCU
212B
212B CD573A                     CALL SSYNTAX6       ;GET CURVATURE
212E
212E 21D05A                     LD HL,TEMPB3
2131 7E                         LD A,(HL)
2132 A7                         AND A
2133 2803                       JR Z,DRNC2        ;JR IF NOT DRAW X,Y,Z
2135
2135 F7                         RST 30H
2136 5E6A                       DW DRCURVE-8000H
2138
2138 74         DRNC2:          LD (HL),H           ;TEMPB3=NZ, MEANS "CURVED"
2139
2139 EF                         DB CALC
213A C8                         DB STOD0
213B 33                         DB EXIT             ;DELETE CURVATURE SO X,Y CAN BE ACCESSED
213C
213C 1809                       JR DRAWTOFD
213E
213E CD153B     DRNOCU:         CALL CHKEND
2141
2141 3AD05A                     LD A,(TEMPB3)
2144 A7                         AND A
2145 204C                       JR NZ,DRAWFD        ;JR IF NOT DRAW TO.
2147                                                ;ELSE DO DRAW TO. TEMPB3=0, MEANS "STRAIGHT"
2147
2147                ;NOTE: COORDS SYSTEM WITH XOS, YOS ETC NORMAL IS 0 TO 255/512, AND -16 TO 175
2147                ;Y COORD STORED INVERTED AS 0 (TOP) TO 191 (BOTTOM)
2147
2147 CD9327     DRAWTOFD:       CALL GTFCOORDS      ;FIDDLE USING XOS/YOS/XRG/YRG IF "DRAW TO"
214A                                                ;UNSTACK COORDS.
214A                                                ;B=Y, WITH 0 AT TOP. IF THIN PIX, HL=X, CY
214A                                                ;ELSE C=X. RANGES CHECKED ALREADY.
214A 380A                       JR C,FDRNR
214C
214C 3AD05A                     LD A,(TEMPB3)
214F A7                         AND A
2150 1E02                       LD E,2              ;BLITZ CODE FOR "DRAW TO - FAT"
2152 CC6628                     CALL Z,GRAREC       ;CALL IF STRAIGHT (CURVES ARE RECURSIVE
2155                                                ;STRAIGHT LINE DRAWS)
2155 A7                         AND A
2156
2156 08         FDRNR:          EX AF,AF'           ;CY IF THIN
2157
2157                ;NOW CONVERT TO SIGNED DISPLACEMENT FROM CURRENT POSN TO NEW ONE
```

```
2157
2157 3A415A     JDRTO3:       LD A,(YCOORD)
215A 90                       SUB B                ;SUB Y COORD, Y DEST
215B 16FF                     LD D,0FFH            ;ASSUME -VE
215D 3004                     JR NC,ADJOK2         ;JR IF COORD NEEDS -VE INCREMENT TO REACH DEST
215F
215F ED44                     NEG
2161 1601                     LD D,01H
2163
2163 47         ADJOK2:       LD B,A               ;B=Y DIFF, D=Y DIFF SGN
2164 08                       EX AF,AF'            ;CY IF THIN PIX
2165 380F                     JR C,THINADJ
2167
2167 3A425A                   LD A,(XCOORD)
216A 91                       SUB C                ;SUB X COORD, X DEST
216B 1EFF                     LD E,0FFH            ;ASSUME -VE
216D 3004                     JR NC,ADJOK3         ;JR IF COORD NEEDS -VE INCREMENT TO REACH DEST
216F
216F ED44                     NEG
2171 1E01                     LD E,01H
2173
2173 4F         ADJOK3:       LD C,A               ;C=Y DIFF, E=Y DIFF SGN
2174 1814                     JR ADJOK5
2176
2176 D5         THINADJ:      PUSH DE
2177 ED5B425A                 LD DE,(XCOORD)
217B AF                       XOR A
217C ED52                     SBC HL,DE            ;SUB X DEST, X COORD
217E 3C                       INC A                ;ASSUME +VE
217F 3007                     JR NC,ADJOK4         ;JR IF COORD NEEDS +VE INCREMENT TO REACH DEST
2181
2181 EB                       EX DE,HL
2182 6F                       LD L,A               ;L=1
2183 3D                       DEC A
2184 67                       LD H,A               ;H=0. HL=1 COMPS FOR CARRY SET HERE
2185 ED52                     SBC HL,DE            ;NEGATE HL.
2187 3D                       DEC A                ;A=FF
2188
2188 D1         ADJOK4:       POP DE
2189 5F                       LD E,A
218A
218A 3AD05A     ADJOK5:       LD A,(TEMPB3)
218D A7                       AND A
218E 280E                     JR Z,JDRAW
2190
2190 F7                       RST 30H
2191 386A                     DW DRTCRV-8000H
2193                        ;  JP DRTCRV
2193
2193            ;REL. DRAW
2193
2193 CDF827     DRAWFD:       CALL DRCOORDFD       ;FIDDLE USING XRG/YRG ONLY
2196
2196            ;REG USE: HL=LARGER AND SMALLER DIFFS, DE=SGN Y AND X, THEN D=M3 INK
2196            ;         B=POINT COUNT, C=TRACKING ERROR
2196            ;         HL"=Y AND X COORDS, DE"=HOR OR VERT STEP, BC"=DIAG STEP
2196
2196 CDCC27     DRAWLINE:     CALL TWONUMS         ;B=Y DIFF, C=X DIFF, D=SGN Y, E=SGN X (01/FF)
2199                                               ; OR HL=X AND CY IF THINPIX
2199                        ;      JR C,JDRAW      ;JR IF THIN
2199
2199 1D                       DEC E                ;SGN X -> FE/00 (BLITZ CODE FOR FAT REL DRAW)
219A D46628                   CALL NC,GRAREC
219D 1C                       INC E                ;NORMAL AGAIN
219E
219E CDB13F     JDRAW:        CALL SPSS            ;SAVE PAGE, SELECT SCREEN
21A1 CDB423                   CALL SETIY           ;POINT IY TO APPROPRIATE PLOT ROUTINE.
21A4                                               ;A=INK IF M3, CY SET IF THIN PIX M2
21A4 DA4A22                   JP C,THINDRAW
21A7
21A7 08                       EX AF,AF'            ;A"=INK TO USE FOR MODE 3
21A8 DD212A22                 LD IX,PRLABEL        ;RETURN ADDR FROM PLOT ROUTINE
21AC 2A415A                   LD HL,(YCOORD)       ;H=X COORD, L=Y
21AF 7C                       LD A,H
21B0 65                       LD H,L
21B1 6F                       LD L,A               ;H=Y, L=X
21B2 E5                       PUSH HL
21B3
21B3 D9                       EXX
21B4 E1                       POP HL               ;HL"=COORDS
21B5 D9                       EXX
21B6
21B6 79                       LD A,C
21B7 A7                       AND A
21B8 2812                     JR Z,CHKYCO2         ;IF X DISP=0 X WON"T RUN OFF. AVOID +/- 0
21BA
21BA 7D                       LD A,L
21BB 1D                       DEC E                ;DEC X SGN
21BC 2806                     JR Z,DOXADD          ;JR AND ADD XCOORD AND DISTANCE IF +VE
21BE
21BE A7                       AND A
21BF 281B                     JR Z,OSERRHP         ;ERROR IF MOVE LEFT AND X IS ZERO
21C1
21C1 91                       SUB C                ;SUB X,X DISP
21C2 1805                     JR CHKYCO
21C4
21C4 FEFF       DOXADD:       CP 255
21C6 2814                     JR Z,OSERRHP         ;ERROR IF MOVE RIGHT AND X IS AT RHS
21C8
21C8 81                       ADD A,C              ;ADD X,X DISP
21C9
21C9 1C         CHKYCO:       INC E
21CA 381A                     JR C,RUNOFF          ;JR IF ADD OR SUB CARRIED
```

```
21CC
21CC 78          CHKYCO2:      LD A,B
21CD A7                        AND A
21CE 7C                        LD A,H
21CF 2811                      JR Z,FINCHK2      ;Y WON"T RUN OFF IF DISP=0
21D1
21D1 15                        DEC D             ;INC Y SGN
21D2 2806                      JR Z,DOYADD       ;JR IF +VE (MOVE DOWN)
21D4
21D4 A7                        AND A
21D5 2805                      JR Z,OSERRHP      ;ERROR IF MOVE UP AND Y IS ZERO (AT TOP)
21D7
21D7 90                        SUB B             ;SUB Y,Y DISP
21D8 1805                      JR FINCHK
21DA
21DA FEBF        DOYADD:       CP 191
21DC
21DC 286A        OSERRHP:      JR Z,OSERROR      ;ERROR IF MOVE DOWN AND Y IS AT BOTTOM
21DE
21DE 80                        ADD A,B           ;ADD Y,Y DISP
21DF
21DF 14          FINCHK:       INC D             ;(D=SGN AGAIN. CY NOT ALTERED)
21E0 3804                      JR C,RUNOFF
21E2
21E2 FEC0        FINCHK2:      CP 192
21E4 3804                      JR C,DRMSUBC      ;JR IF Y COORD IN RANGE
21E6
21E6 DD212F22    RUNOFF:       LD IX,PLOTCHK     ;CHECK AFTER PLOT IF EDGE HAS BEEN REACHED,
21EA                                             ;(IF LINE WILL RUN OFF-SCREEN) RATHER THAN
21EA                                             ;JUST RETURNING TO DRAW ROUTINE. IF OK, RETS
21EA                                             ;TO DRAW, ELSE EXITS WITH ERROR
21EA
21EA CDF221      DRMSUBC:      CALL DRMSUB
21ED D9                        EXX             ;GET HL=FINAL COORDS
21EE 37                        SCF             ;SIGNAL "OK"
21EF C33C22                    JP DRAWEND
21F2
21F2 D5          DRMSUB:       PUSH DE           ;U/D AND L/R FLAGS
21F3
21F3 D9                        EXX
21F4 C1                        POP BC
21F5 0C                        INC C
21F6 2001                      JR NZ,NDHDIAG
21F8
21F8 05                        DEC B           ;IF C WAS FF, THEN ADDING C (-VE X DISP) TO L
21F9                                           ; (X COORD) WILL ALWAYS CARRY TO H+B, SO DEC B.
21F9 0D          NDHDIAG:      DEC C           ;BC"=DIAG STEP
21FA D9                        EXX
21FB
21FB 79                        LD   A,C
21FC B8                        CP   B
21FD 3005                      JR   NC,XGRTR  ;JR IF X DIFF >=Y DIFF
21FF
21FF 69                        LD   L,C        ;L=ABS X DIFF (LESS THAN Y DIFF)
2200 1E00                      LD   E,0        ;WE WILL SOMETIMES NEED TO USE LR=STAY, SINCE X
2202                                           ; DIFF IS LESS THAN Y DIFF.
2202                                           ;B=GREATEST DIFFERENCE (Y)
2202 1806                      JR   DRPREL
2204
2204 B0          XGRTR:        OR   B
2205 C8                        RET  Z          ;RET IF BOTH DIFFS EQUAL - NO LINE!
2206
2206 68                        LD   L,B        ;L=ABS Y DIFF ( <= X DIFF)
2207 41                        LD   B,C        ;B=GREATEST DIFFERENCE (X)
2208 1600                      LD   D,0        ;SOMETIMES NEED UP/DOWN=STAY
220A
220A D5          DRPREL:       PUSH DE
220B
220B D9                        EXX
220C D1                        POP DE
220D 1C                        INC E
220E 2001                      JR NZ,NDHFLAT
2210
2210 15                        DEC D           ;IF E WAS FF, THEN ADDING E (-VE X DISP) TO L
2211                                           ; (X COORD) WILL ALWAYS CARRY TO H+D, SO DEC D.
2211 1D          NDHFLAT:      DEC E           ;DE"=HORIZ OR VERT STEP
2212 D9                        EXX
2213
2213 08                        EX AF,AF'
2214 57                        LD D,A          ;D=M3 INK
2215 60                        LD H,B          ;H=GREATEST COORD DIFF
2216 78                        LD A,B          ;INITIALISE TRACKING ERROR BYTE. (IT ACCUMULATES
2217 CB3F                      SRL A           ;ERRORS FROM NOT MOVING ALONG LESS-DIF AXIS)
2219
2219 85          DRLOOP:       ADD A,L         ;ADD LESSER COORD DIFF TO TRACKING ERROR
221A 3803                      JR C,TWOMOVE    ;JR C AND SUB GREATER DIFF, MOVE IN BOTH AXES
221C                                           ;(ERROR IS BOUND TO MOVE BACK INTO RANGE)
221C
221C BC                        CP H            ;CP GREATER COORD DIFF
221D 3806                      JR C,ONEMOVE    ;THE ERROR IS NOT OUT OF RANGE SO FAR. IF WE CANNOT
221F                                           ;YET MAKE A MOVE ON THE LESSER-CHANGING AXIS (SMALL
221F                                           ;ADDITIONS TO THE TRACKING ERROR DO NOT YET JUSTIFY
221F                                           ;IT) MOVE JUST ON THE GREATER-CHANGING AXIS (JR)
221F
221F 94          TWOMOVE:      SUB H           ;SUB GREATER
2220 4F                        LD C,A          ;SAVE TRACKING ERROR BYTE IN C
2221
2221 D9                        EXX
2222 09                        ADD HL,BC       ;ADD DIAGONAL STEP
2223 FDE9                      JP (IY)         ;JP TO CORRECT PLOT ROUTINE, THEN TO PRLABEL
2225
2225 4F          ONEMOVE:      LD   C,A        ;SAVE TRACKING ERROR
2226
```

```
2226 D9                         EXX
2227 19                         ADD HL,DE       ;ADD HORIZ OR VERT MOVE
2228 FDE9                       JP (IY)         ;JP TO CORRECT PLOT ROUTINE, THEN TO PRLABEL
222A
222A D9             PRLABEL:    EXX
222B 79                         LD   A,C        ;GET TRACKING ERROR BACK
222C 10EB                       DJNZ DRLOOP     ;LOOP FOR GREATER AXIS DIFF. OF POINTS
222E
222E C9                         RET             ;RETURN WITH HL"=COORDS, DE", BC"=DIR FLAGS
222F
222F                ;POST-CHECK ROUTINE USED AFTER PLOT IF LINE WILL RUN OFF-SCREEN
222F
222F 7D             PLOTCHK:    LD A,L
2230 3C                         INC A
2231 FE02                       CP 2
2233 3F                         CCF
2234 3006                       JR NC,DRAWEND   ;JR IF X COORD=255 OR 0
2236
2236 7C                         LD A,H
2237 3D                         DEC A           ;0->255,191->190
2238 FEBE                       CP 190
223A 38EE                       JR C,PRLABEL    ;STILL OK - SO BACK TO DRAW LOOP
223C
223C F5             DRAWEND:    PUSH AF         ;SAVE NC IF HIT EDGE
223D 7C                         LD A,H
223E 65                         LD H,L
223F 6F                         LD L,A
2240 22415A                     LD (YCOORD),HL
2243 CDC63F                     CALL TRCURP       ;TEMPS, RESET PAGE
2246 F1                         POP AF            ;NC IF HIT EDGE, C IF OK
2247 D8                         RET C
2248
2248 CF             OSERROR:    RST 08H
2249 20                         DB 32             ;"Off screen"
224A
224A                ;THIN PIXEL DRAW
224A                ;ENTRY: B=Y DIFF,HL=X DIFF, D=SGN Y, E=SGN X (01/FF)
224A
224A D9             THINDRAW:   EXX
224B 210000                     LD HL,0           ;"COORDS"=0
224E D9                         EXX
224F FD217922                   LD IY,THINDR2
2253
2253 7C             DUBT:       LD A,H
2254 A7                         AND A
2255 281B                       JR Z,EASYTHIN
2257
2257 1F                         RRA
2258 7D                         LD A,L            ;HALVE X DIFF
2259 1F                         RRA               ;A=HALF X DIFF
225A 4F                         LD C,A
225B F5                         PUSH AF           ;SAVE HALF X DIFF AND CY IF DIFF IS ODD
225C 78                         LD A,B
225D A7                         AND A
225E 1F                         RRA
225F 47                         LD B,A
2260 F5                         PUSH AF           ;SAVE HALF Y DIFF AND CY IF DIFF IS ODD
2261 D5                         PUSH DE
2262 CDF221                     CALL DRMSUB
2265 D1                         POP DE            ;SGN FLAGS
2266 F1                         POP AF            ;Y/2
2267 CE00                       ADC A,0           ;INC IF WAS ODD
2269 47                         LD B,A
226A F1                         POP AF            ;X/2
226B 2601                       LD H,1
226D CE00                       ADC A,0           ;INC IF WAS ODD - COULD NOW BE ZERO IF ORIG=1FF
226F 6F                         LD L,A
2270 38E1                       JR C,DUBT         ;JR WITH HL=0100 IF DOUBLE NEEDED AGAIN
2272
2272 4D             EASYTHIN:   LD C,L
2273 CDF221                     CALL DRMSUB
2276 C3C63F                     JP TRCURP
2279
2279                ;THIN DRAW KEEPS SETTING DRAW"S COORDS IN HL TO 0000, SO HL WILL EQUAL BC OR DE
2279
2279 C5             THINDR2:    PUSH BC
227A D5                         PUSH DE
227B EB                         EX DE,HL
227C 3A415A                     LD A,(YCOORD)
227F 82                         ADD A,D
2280 47                         LD B,A
2281 2A425A                     LD HL,(XCOORD)
2284 7B                         LD A,E
2285 A7                         AND A
2286 2811                       JR Z,TPNCHNG
2288
2288 F28E22                     JP P,TPINC
228B
228B 04                         INC B             ;COMPENSATE FOR FIDDLED DIR FLAGS
228C 2B                         DEC HL            ;DEC X
228D 2B                         DEC HL            ;COMP FOR NEXT INSTR
228E
228E 23             TPINC:      INC HL
228F
228F 7C                         LD A,H
2290 FE02                       CP 2
2292 30B4                       JR NC,OSERROR     ;ERROR IF X INCED TO 0200H OR DECED TO FFFFH
2294
2294 78                         LD A,B
2295 FEC0                       CP 192
2297 30AF                       JR NC,OSERROR
2299
2299 CDDC22         TPNCHNG:    CALL TDPLOT
```

```
229C 210000                 LD HL,0
229F D1                     POP DE
22A0 C1                     POP BC
22A1 C32A22                 JP PRLABEL
22A4
22A4                    INCLUDE GRAPH1.SAM     ;DRAW CURVE, PLOT
22A4            ;GRAPH1.SAM
22A4
22A4            ;PLOT.SAM
22A4
22A4 3A4D5A     JPLOT:      LD A,(THFATT)
22A7 A7                     AND A
22A8 2013                   JR NZ,JPLOT3       ;JR IF FAT
22AA
22AA 1827                   JR THINPLOT
22AC
22AC CDF23A     PLOT:       CALL SYNTAX9
22AF CD153B                 CALL CHKEND
22B2
22B2            ;ENTRY POINT FOR BLITZ
22B2
22B2 CD9327     PLOTFD:     CALL GTFCOORDS     ;CORRECT Y SO 0 AT TOP, 191 AT BOT. CHECK X AND Y
22B5 DAD322                 JP C,THINPLOT      ;JR IF THIN PLOT - HL=X, B=Y. ELSE C=X, B=Y
22B8
22B8 1E01                   LD E,1             ;BLITZ CODE FOR PLOT (FAT)
22BA CD6628                 CALL GRAREC
22BD
22BD CDB13F     JPLOT3:     CALL SPSS          ;SAVE PAGE, SELECT SCREEN
22C0 61                     LD H,C
22C1 68                     LD L,B
22C2 22415A                 LD (YCOORD),HL     ;UPDATE Y COORD AND X COORD LSB
22C5 60                     LD H,B             ;H=Y WITH 0 AT TOP
22C6 69                     LD L,C             ;L=X
22C7 CDB423                 CALL SETIY
22CA D9                     EXX
22CB 57                     LD D,A             ;INK TO D'
22CC D9                     EXX
22CD DD21C63F               LD IX,TRCURP
22D1 FDE9                   JP (IY)
22D3            ;*************************************************************************
22D3            ;THIN PIXEL PLOT. PLOT HL,B. HL CHECKED ALREADY, 00-01FF
22D3
22D3 CDB13F     THINPLOT:   CALL SPSS          ;USED BY PLOT
22D6 CDDC22                 CALL TDPLOT
22D9 C3C63F                 JP TRCURP
22DC
22DC 22425A     TDPLOT:     LD (XCOORD),HL     ;USED BY DRAW - AVOIDS PAGING SCREEN IN AND OUT
22DF 78                     LD A,B
22E0 32415A                 LD (YCOORD),A
22E3
22E3 DDE1       M2CTPLOT:   POP IX             ;USED BY CIRCLE - AVOIDS ALTERING COORD VARS.
22E5 E5                     PUSH HL
22E6 D5                     PUSH DE
22E7 C5                     PUSH BC
22E8 7D                     LD A,L             ;A=X
22E9 CB1C                   RR H
22EB CB1D                   RR L               ;L=X/2
22ED 60                     LD H,B             ;H=Y
22EE E603                   AND 03H            ;ISOLATE BIT OFFSET
22F0 3C                     INC A
22F1 47                     LD B,A             ;B COUNTS TIMES TO ROTATE MASK
22F2 37                     SCF
22F3 CB1C                   RR H
22F5 CB1D                   RR L               ;HL=Y/2+X/4+8000H=ADDRESS
22F7 3EFC                   LD A,0FCH          ;BIT MASK
22F9
22F9 0F         P80RLP:     RRCA
22FA 0F                     RRCA
22FB 10FC                   DJNZ P80RLP
22FD
22FD 4F                     LD C,A
22FE ED5B535A               LD DE,(OVERT)       ;E=00/01 FOR OVER 0/1, D=00/FF FOR INVERSE 0/1
2302 3A525A                 LD A,(M23INKT)
2305 14                     INC D
2306 2003                   JR NZ,M2TPIN0       ;JR IF INVERSE 0
2308
2308 3A515A                 LD A,(M23PAPT)
230B
230B 47         M2TPIN0:    LD B,A
230C 1D                     DEC E
230D 2806                   JR Z,M2TPOV1         ;JR IF OVER 1
230F
230F 7E                     LD A,(HL)
2310 A8                     XOR B
2311 A1                     AND C                ;FORCE PIXEL TO INK OR PAPER COLOUR
2312 A8                     XOR B
2313 1806                   JR M2TPC
2315
2315 15         M2TPOV1:    DEC D
2316 2030                   JR NZ,DRPLEND2       ;DO NOTHING IF INVERSE 1, OVER 1
2318
2318 79                     LD A,C
2319 2F                     CPL
231A AE                     XOR (HL)             ;REVERSE PIXEL
231B
231B 77         M2TPC:      LD (HL),A
231C 182A                   JR DRPLEND2
231E
231E            ;*************************************************************************
231E            ;MODE 0 PLOT - USED BY DRAW. PLOT L,H
231E
231E E5         M0DPLOT:    PUSH HL
```

```
231F D5                    PUSH DE
2320 C5                    PUSH BC
2321 44                    LD B,H
2322 4D                    LD C,L
2323 CD3B3E                CALL M0PIXAD
2326 47                    LD B,A
2327 1806                  JR M01DPCOM
2329
2329              ;*************************************************************************
2329              ;MODE 1 PLOT -  USED BY DRAW. PLOT L,H
2329
2329 E5           M1DPLOT:   PUSH HL
232A D5                    PUSH DE
232B C5                    PUSH BC
232C CD583E                CALL M1PIXAD      ;HL=Y/8+X/8+8000H=ADDRESS, B=BIT OFFSET (0-7)
232F
232F 3EFE         M01DPCOM:  LD A,0FEH
2331 04                    INC B
2332
2332 0F           DPM2FLP:   RRCA
2333 10FD                  DJNZ DPM2FLP
2335
2335              ;OVER 0,INVERSE 0=FORCE PIXEL HIGH
2335              ;OVER 1,INVERSE 0=REVERSE PIXEL
2335              ;OVER 0,INVERSE 1=FORCE PIXEL LOW
2335              ;OVER 1,INVERSE 1=NO EFFECT
2335
2335 4F                    LD C,A
2336 ED5B535A              LD DE,(OVERT)       ;E=0 IF OVER 0, D= 00/FF FOR INVERSE 0/1
233A 7E                    LD A,(HL)
233B 1D                    DEC E
233C 2801                  JR Z,DYOVER1        ;JR IF OVER 1
233E
233E A1                    AND C
233F
233F 14           DYOVER1:   INC D
2340 2802                  JR Z,DRPLEND        ;JR IF INVERSE 1
2342
2342 A9                    XOR C
2343 2F                    CPL
2344
2344 77           DRPLEND:   LD (HL),A
2345 CD6C3E                CALL POATTR01       ;CALL ATTR SETTER FOR MODES 0,1
2348
2348 C1           DRPLEND2:  POP BC
2349 D1                    POP DE
234A E1                    POP HL
234B DDE9                  JP (IX)
234D
234D              ;*************************************************************************
234D              ;MODE 2 OR 3 PLOTS FOR DRAW. ALL PLOT H,L
234D              ;ENTRY: HL=COORDS (Y,X), IX=RETURN ADDR, D'=INK COLOUR
234D              ;OVER 0:
234D
234D 37           M3DPOV0:   SCF
234E CB1C                  RR H
2350 CB1D                  RR L              ;HL=Y/2+X/2+8000H=ADDR
2352 7E                    LD A,(HL)
2353 D9                    EXX
2354 3809                  JR C,M3DPOV0OD
2356
2356 AA                    XOR D
2357 E60F                  AND 0FH
2359 AA                    XOR D
235A D9                    EXX
235B 77                    LD (HL),A
235C 29                    ADD HL,HL         ;RESTORE HL
235D DDE9                  JP (IX)
235F
235F AA           M3DPOV0OD: XOR D
2360 E6F0                  AND 0F0H
2362 AA                    XOR D
2363 D9                    EXX
2364 77                    LD (HL),A
2365 29                    ADD HL,HL
2366 2C                    INC L             ;RESTORE HL
2367 DDE9                  JP (IX)
2369
2369              ;MODE 2 OR 3 ROUTINE FOR OVER 1 - XOR INK WITH WHAT IS THERE ALREADY
2369              ;HL=Y/X, D'=INK
2369
2369 37           M3DPOV1:   SCF
236A CB1C                  RR H
236C CB1D                  RR L              ;HL=Y/2+X/2+8000H=ADDR
236E D9                    EXX
236F 7A                    LD A,D
2370 D9                    EXX
2371 3807                  JR C,M3DPOV1OD    ;JR IF ODD PIXEL
2373
2373 E6F0                  AND 0F0H          ;NO EFFECT ON RHS BITS (ODD PIX)
2375 AE                    XOR (HL)
2376 77                    LD (HL),A
2377 29                    ADD HL,HL         ;RESTORE HL
2378 DDE9                  JP (IX)
237A
237A E60F         M3DPOV1OD: AND 0FH          ;NO EFFECT ON LHS BITS (EVEN PIX)
237C AE                    XOR (HL)
237D 77                    LD (HL),A
237E 29                    ADD HL,HL
237F 2C                    INC L             ;RESTORE HL
2380
2380 DDE9         M3DPNUL:   JP (IX)           ;INVERSE 1, OVER 1 ROUTINE DOES NOTHING
2382
```

```
2382            ;MODE 2 OR 3 ROUTINE FOR OVER 2 - OR INK WITH WHAT'S THERE ALREADY
2382            ;HL=Y/X, D'=INK
2382
2382 37         M3DPOV2:    SCF
2383 CB1C                   RR H
2385 CB1D                   RR L                ;HL=Y/2+X/2+8000H=ADDR
2387 D9                     EXX
2388 7A                     LD A,D              ;FETCH INK
2389 D9                     EXX
238A 3807                   JR C,M3DPOV2OD      ;JR IF ODD PIXEL
238C
238C E6F0                   AND 0F0H            ;NO EFFECT ON RHS BITS (ODD PIX)
238E B6                     OR (HL)             ;OR INK WITH SCREEN
238F 77                     LD (HL),A
2390 29                     ADD HL,HL           ;RESTORE HL
2391 DDE9                   JP (IX)
2393
2393 E60F       M3DPOV2OD:  AND 0FH
2395 B6                     OR (HL)
2396 77                     LD (HL),A
2397 29                     ADD HL,HL
2398 2C                     INC L               ;RESTORE HL
2399 DDE9                   JP (IX)
239B
239B            ;MODE 2 OR 3 ROUTINE FOR OVER 3 - 'AND' INK WITH WHAT'S THERE ALREADY
239B            ;HL=Y/X, D'=INK
239B
239B 37         M3DPOV3:    SCF
239C CB1C                   RR H
239E CB1D                   RR L                ;HL=Y/2+X/2+8000H=ADDR
23A0 D9                     EXX
23A1 7A                     LD A,D              ;FETCH INK
23A2 D9                     EXX
23A3 3807                   JR C,M3DPOV3OD      ;JR IF ODD PIXEL
23A5
23A5 F60F                   OR 0FH              ;SO NO EFFECT ON RHS BITS (ODD PIX)
23A7 A6                     AND (HL)            ;AND INK WITH SCREEN
23A8 77                     LD (HL),A
23A9 29                     ADD HL,HL           ;RESTORE HL
23AA DDE9                   JP (IX)
23AC
23AC F6F0       M3DPOV3OD:  OR 0F0H
23AE A6                     AND (HL)
23AF 77                     LD (HL),A
23B0 29                     ADD HL,HL
23B1 2C                     INC L               ;RESTORE HL
23B2 DDE9                   JP (IX)
23B4
23B4            ;*******************************************************************************
23B4            ;SET IY TO APPROPRIATE PLOT ROUTINE. A=INK TO USE IF MODE 2 OR 3. CY IF MODE 2
23B4            ;THIN PLOT, IY NOT SET
23B4
23B4 3ADD5A     SETIY:      LD A,(SETIYV+1)
23B7 A7                     AND A
23B8 3A405A                 LD A,(MODE)
23BB 2048                   JR NZ,STIY6
23BD
23BD FD211E23               LD IY,M0DPLOT
23C1 A7                     AND A
23C2 C8                     RET Z               ;RET IF MODE 0
23C3
23C3 FD212923               LD IY,M1DPLOT
23C7 3D                     DEC A
23C8 C8                     RET Z               ;RET IF MODE 1
23C9
23C9 3D                     DEC A
23CA 2006                   JR NZ,STIY1         ;JR IF NOT MODE 2
23CC
23CC 3A4D5A                 LD A,(THFATT)
23CF A7                     AND A
23D0 37                     SCF                 ;'THIN PLOT'
23D1 C8                     RET Z               ;RET IF THIN PLOT, ELSE USE M2/M3 ROUTINES
23D2
23D2 3A555A     STIY1:      LD A,(GOVERT)
23D5 FD214D23               LD IY,M3DPOV0
23D9 A7                     AND A
23DA 280E                   JR Z,STIY3          ;JR IF OVER 0
23DC
23DC 3D                     DEC A
23DD 2817                   JR Z,STIY5          ;JR IF OVER 1
23DF
23DF FD218223               LD IY,M3DPOV2       ;ORING ROUTINE
23E3 3D                     DEC A
23E4 2804                   JR Z,STIY3          ;JR IF OVER 2
23E6
23E6 FD219B23               LD IY,M3DPOV3       ;ANDING ROUTINE
23EA
23EA 3A545A     STIY3:      LD A,(INVERT)
23ED A7                     AND A
23EE
23EE 3A525A     STIY4:      LD A,(M23INKT)
23F1 C8                     RET Z               ;RET WITH A=INK IF INVERSE 0
23F2
23F2 3A515A                 LD A,(M23PAPT)      ;ELSE A=PAPER
23F5 C9                     RET
23F6
23F6            ;OVER 1
23F6
23F6 FD216923   STIY5:      LD IY,M3DPOV1       ;ROUTINE TO XOR INK WITH SCREEN
23FA 3A545A                 LD A,(INVERT)
23FD A7                     AND A
23FE 28EE                   JR Z,STIY4          ;XOR INK TO SCREEN IF OVER 1,INVERSE 0
2400
```

```
2400 FD218023              LD IY,M3DPNUL     ;DO NOTHING IF INVERSE 1,OVER 1
2404 C9                    RET
2405
2405              ;VECTORED
2405 E5           STIY6:   PUSH HL
2406 2ADC5A                LD HL,(SETIYV)
2409 CD0500                CALL HLJUMP
240C E1                    POP HL
240D A7                    AND A             ;NC - NO THIN PIX
240E C9                    RET
240F
240F                       INCLUDE GRAPH2.SAM     ;BLITZ, FILL, TRANSCR, CRDFID
240F              ;GRAPH2.SAM
240F              ;BLITZ MAIN LOOP
240F
240F EB           FDMAINLP:  EX DE,HL
2410 D5                    PUSH DE           ;LENGTH
2411 E5                    PUSH HL           ;POSN
2412 7E                    LD A,(HL)         ;SGN X OR CMD CODE
2413 5F                    LD E,A
2414 23                    INC HL
2415 56                    LD D,(HL)         ;ABS X OR SINGLE PARAM
2416 23                    INC HL
2417 46                    LD B,(HL)         ;SGN Y, OR ABS Y IF PLOT/DRAW TO/CIRCLE
2418 3C                    INC A             ;SGN=00 OR 01, PLOT CODE=02, DRAW TO=03
2419 FE02                  CP 2
241B 386B                  JR C,DRCL         ;JR IF 00 OR 01 (REL. DRAW)
241D
241D D604                  SUB 4             ;2-> -2 (PLOT), 3-> -1 (DRAWTO), 4->0 (CIRCLE)
241F 2003                  JR NZ,FDNTCIR     ;JR IF CMD WAS NOT CIRCLE
2421
2421 23                    INC HL
2422 7E                    LD A,(HL)         ;GET RADIUS
2423 37                    SCF
2424
2424 302F         FDNTCIR: JR NC,FDATTRS     ;JR IF 5 OR MORE (CODE NOW 1 OR MORE)
2426
2426 08                    EX AF,AF'         ;SAVE RADIUS AND Z FLAG, OR NZ AND -1/-2 (DTO/PLT)
2427 AF                    XOR A             ;'INTEGER'
2428 5F                    LD E,A            ;E (SGN) =00 (POS)
2429 4B                    LD C,E            ;MSB=00
242A CDF01C                CALL STKSTORE     ;STACK AEDC - 00 00 X 00.
242D 3A5C5A                LD A,(ORGOFF)
2430 57                    LD D,A
2431 3EBF                  LD A,191
2433 90                    SUB B             ;A=Y WITH 191 AT TOP, 0 AT BOT
2434 92                    SUB D             ;SUB ORGOFF
2435 57                    LD D,A
2436 7B                    LD A,E            ;A=0
2437 3002                  JR NC,FDPOS
2439
2439 1D                    DEC E             ;SGN=-VE **
243A 0D                    DEC C             ;MSB=FF
243B
243B CDF01C       FDPOS:   CALL STKSTORE     ;STACK AEDC - 00 SGN  Y 00
243E 08                    EX AF,AF'
243F 2006                  JR NZ,FDNTRAD
2441
2441 4F                    LD C,A            ;RADIUS
2442 CD7D20                CALL CIRCLEFD
2445 1854                  JR MNLP4B
2447
2447 3C           FDNTRAD: INC A
2448 2004                  JR NZ,FDPLOT
244A
244A CD4721                CALL DRAWTOFD     ;DRAW_TO COORDS ON FPCS
244D BF                    CP A              ;SET Z
244E
244E C4B222       FDPLOT:  CALL NZ,PLOTFD    ;PLOT THE COORDS ON FPC STACK
2451 0E03                  LD C,3
2453 1848                  JR MAINLPH
2455
2455 3D           FDATTRS: DEC A
2456 2827                  JR Z,FDOVER
2458
2458 3D                    DEC A
2459 2816                  JR Z,FDINK
245B
245B 3D                    DEC A
245C 280B                  JR Z,FDCLS
245E
245E 3D                    DEC A
245F 206C                  JR NZ,FDINVERR
2461
2461 4A           FDPAU:   LD C,D
2462 0600                  LD B,0
2464 F7                    RST 30H           ;** BLITZ PAUSE BUG FIX
2465 0EF0                  DW PAU1
2467 181B                  JR FD1PARAM
2469
2469 7A           FDCLS:   LD A,D
246A CDAE06                CALL CLSBL        ;CLEAR WINDOW OR ALL SCREEN
246D 0E02                  LD C,2
246F 182C                  JR MAINLPH
2471
2471 7A           FDINK:   LD A,D
2472 32525A                LD (M23INKT),A
2475 214E5A                LD HL,ATTRT
2478 AE                    XOR (HL)
2479 E607                  AND 07H
247B AE                    XOR (HL)
247C 77                    LD (HL),A
247D 1805                  JR FD1PARAM
```

```
247F
247F 7A          FDOVER:    LD A,D
2480 23                     INC HL
2481 32555A                 LD (GOVERT),A
2484
2484 0E02        FD1PARAM:  LD C,2
2486 1815                   JR MAINLPH
2488
2488 23          DRCL:      INC HL
2489 7E                     LD A,(HL)          ;ABS Y . NOW E/D=SGN X/X, B/A=SGN Y,Y
248A                                           ;(SGN=00 OR FF)
248A 08                     EX AF,AF'          ;SAVE Y
248B AF                     XOR A              ;'INTEGER'
248C 4B                     LD C,E             ;MSB MATCHES SGN
248D CDF01C                 CALL STKSTORE      ;STACK AEDC
2490 08                     EX AF,AF'
2491 58                     LD E,B
2492 57                     LD D,A
2493 AF                     XOR A
2494 4B                     LD C,E
2495 CDF01C                 CALL STKSTORE      ;STACK AEDC
2498 CD9321                 CALL DRAWFD
249B
249B 0E04        MNLP4B:    LD C,4
249D
249D AF          MAINLPH:   XOR A
249E 47                     LD B,A
249F E1                     POP HL             ;POSN
24A0 09                     ADD HL,BC          ;ADVANCE POSN, NC
24A1 7C                     LD A,H
24A2 3C                     INC A
24A3 CCF23F                 CALL Z,INCURPAGE   ;ONLY IF ADDR IS FFXX IS THERE A NEED TO INC PAGE
24A6 EB                     EX DE,HL
24A7 E1                     POP HL             ;LENGTH
24A8
24A8 ED42        FDLNTST:   SBC HL,BC          ;SUB BYTES PROCESSED IN LAST GRAPHICS CMD
24AA D20F24                 JP NC,FDMAINLP     ;JR TILL PAST END
24AD
24AD F1                     POP AF
24AE 32815B                 LD (GRARF),A
24B1 C9                     RET
24B2
24B2             ;FAST DRAW AND PLOT COMPILER (DRAW A STRING)
24B2
24B2             ;STRING HAS EITHER:
24B2
24B2             ;     SGN X (00/FF), X, SGN Y (00/FF), Y
24B2             ;     01, X, Y  - PLOT
24B2             ;     02, X, Y  - DRAW TO
24B2             ;     03, X,Y,R - CIRCLE
24B2             ;     04, O     - OVER O
24B2             ;     05, I     - INK I
24B2             ;     06, C     - CLS C (IF C=0, CLEAR ENTIRE SCREEN. IF 1, CLEAR WINDOW)
24B2             ;     07, N     - PAUSE N
24B2
24B2 CD663A      BLITZ:     CALL SYNTAXA
24B5
24B5 CDDC3F                 CALL GETSTRING     ;DE=START, BC=LEN, PAGED IN
24B8
24B8 D5          JBLITZ:    PUSH DE
24B9 C5                     PUSH BC
24BA CDD211                 CALL GRATEMPS
24BD C1                     POP BC
24BE D1                     POP DE
24BF 21815B                 LD HL,GRARF
24C2 7E                     LD A,(HL)
24C3 F5                     PUSH AF
24C4 AF                     XOR A              ;NC
24C5 77                     LD (HL),A          ;GR. RECORD OFF, BUT CURRENT STATUS REPLACED AT END
24C6 60                     LD H,B
24C7 69                     LD L,C             ;HL=LENGTH, DE=POSN
24C8 010100                 LD BC,1            ;AT START,SUB 1 FROM LEN, EXIT IF LEN WAS 0. ALSO
24CB                                           ;ALLOWS LATER LEN TEST TO BE ON CY/NC, NOT CY+Z
24CB 18DB                   JR FDLNTST
24CD
24CD CF          FDINVERR:  RST 08H
24CE 23                     DB 35              ;INVALID BLITZ CODE
24CF
24CF
24CF
24CF             ;FILL.SAM
24CF             ;*************************************************************************
24CF             ;TEXTURED FILL COMMAND
24CF             ;     EG FILL X,Y              ;USES CURRENT INK
24CF             ;     EG FILL USING A$,X,Y     ;USES INKS CODED IN PATTERN STRING
24CF             ;     EG FILL INK 3,X,Y
24CF             ;     EG FILL USING A$,INK 3,X,Y GIVES SOLID FILL, IGNORES A$
24CF
24CF             ;FROM JUMP TABLE:
24CF             ;IF DE=0 DO SOLID FILL. ELSE USE 128 BYTES FROM DE AS PATTERN. IF A=0 MAKE
24CF             ;CHECK SCREEN.
24CF
24CF 22C85A      JFILL:        LD (TEMPW1),HL   ;BYTE OFFSET IF MODE 2
24D2 C5                        PUSH BC          ;START COORDS
24D3 0601                      LD B,1           ;USE 128 BYTES SINCE BC<>131
24D5 32CE5A                    LD (TEMPB1),A
24D8 1842                      JR JFILL2
24DA
24DA FE85        FILL:         CP USINGTOK
24DC 2007                      JR NZ,PASTUSNG
24DE
24DE CDA03A                    CALL SEXPTSTR
24E1 CD813A                    CALL INSISCSC    ;REQUIRE COMMA OR SEMICOLON
```

```
24E4 BF                         CP A              ;SET ZERO FLAG
24E5
24E5 C4FC3A     PASTUSNG:       CALL NZ,CONDSTK0  ;STACK 0 IF NO 'USING' AND RUNNING
24E8
24E8 CDF23A                     CALL SYNTAX9      ;CHECK FOR COLOURS, X,Y
24EB FE2C                       CP ","
24ED 200B                       JR NZ,FCKE
24EF
24EF CD573A                     CALL SSYNTAX6     ;0 OR DEFAULT MEANS 'COPY TO CHK SCRN', 1 MEANS
24F2                                              ;'SUPRESS COPYING'
24F2
24F2 111E02                     LD DE,0200H+30
24F5 CD621F                     CALL LIMBYTE      ;ACCEPT 0 OR 1 ONLY
24F8 1804                       JR FSCY
24FA
24FA CD153B     FCKE:           CALL CHKEND
24FD AF                         XOR A
24FE
24FE 32D05A     FSCY:           LD (TEMPB3),A
2501 CD9512                     CALL CHKMD23      ;INSIST ON MODE 2 OR 3
2504 F7                         RST 30H
2505 C8E9                       DW CIFILSR        ;COORDS TO BC, OFFSET IF MODE 2 TO TEMPW1
2507 300F                       JR NC,FILLX4      ;JR IF MODE 3 ELSE HL=PIX OFFSET
2509
2509 7D                         LD A,L
250A CB1C                       RR H
250C 1F                         RRA
250D 1F                         RRA               ;A=OFFSET/4 (BYTE)
250E E67E                       AND 7EH
2510 32C85A                     LD (TEMPW1),A     ;OFFSET 0,2,4 ETC
2513 7D                         LD A,L
2514 E607                       AND 7
2516 81                         ADD A,C
2517 4F                         LD C,A
2518
2518 C5         FILLX4:         PUSH BC           ;COORDS TO START AT
2519 CDDC3F                     CALL GETSTRING    ;SWITCHES IT IN, GETS DE=START, BC=LEN (<4000)
251C
251C 218050     JFILL2:         LD HL,FILBUFF     ;STORES PATTERN
251F 7A                         LD A,D
2520 B3                         OR E
2521 200C                       JR NZ,DEFPAT
2523
2523 3A525A                     LD A,(M23INKT)
2526 77                         LD (HL),A         ;IF NO 'USING', USE SOLID CURRENT INK
2527 54                         LD D,H            ;STRING START OF 0 MEANS USE SOLID PATTERN
2528 5D                         LD E,L
2529 13                         INC DE            ;DE=BUFFER START+1
252A 017F00                     LD BC,127
252D 1810                       JR SOLIDEN        ;BUFFER WILL BE FILLED WITH SOLID INK
252F
252F 78         DEFPAT:         LD A,B
2530 A7                         AND A
2531 2008                       JR NZ,DEFPAT1
2533
2533 79                         LD A,C
2534 FE83                       CP 131
2536 2003                       JR NZ,DEFPAT1     ;JR IF LEN NOT THAT OF A 2*2 GET BLOCK
2538                                              ;(CONTROL CODE, WIDTH, LENGTH, 128 BYTES)
2538
2538 13                         INC DE
2539 13                         INC DE
253A 13                         INC DE            ;POINT TO DATA
253B
253B EB         DEFPAT1:        EX DE,HL
253C 018000                     LD BC,128
253F
253F EDB0       SOLIDEN:        LDIR              ;COPY STRING TO BUFFER
2541
2541 CDB13F                     CALL SPSS         ;SAVE PAGE,SELECT SCREEN (NO USE OF HL)
2544 3AD05A                     LD A,(TEMPB3)
2547 A7                         AND A
2548 E1                         POP HL            ;START COORDS
2549 E5                         PUSH HL
254A CCA826                     CALL Z,TRANSCR    ;NOW COPY SCREEN TO CHECK SCREEN
254D
254D C1                         POP BC            ;START COORDS
254E ED73CC5A                   LD (TEMPW3),SP
2552 31B0FE                     LD SP,FILLSTK     ;END OF SCREEN GIVES MORE SPACE FOR STACK
2555 CD5F25                     CALL FILLMN
2558 ED7BCC5A                   LD SP,(TEMPW3)
255C C3C63F                     JP TRCURP         ;RESET PAGE AND TEMP COLOURS
255F
255F 3EFE       FILLMN:         LD A,0FEH
2561 F5                         PUSH AF           ;STOPPER TO STACK, START COORDS TO HL
2562 C5                         PUSH BC           ;START COORDS
2563
2563 C1         UNSTK:          POP  BC
2564 78                         LD   A,B
2565 FEFE                       CP   0FEH
2567 C8                         RET Z             ;EXIT IF STOPPER
2568
2568 FEC0                       CP   0C0H
256A 30F7                       JR   NC,UNSTK     ;LOOP IF OFF-SCREEN (FF IF TOO LOW, B0 IF TOO HI)
256C                                              ;CARRY ALWAYS SET HERE
256C
256C 69                         LD L,C            ;NOW GET CHK SCREEN ADDRESS FROM COORDS
256D 1F                         RRA               ;A=B
256E CB1D                       RR L
2570 1F                         RRA
2571 CB1D                       RR L
2573 1F                         RRA
2574 CB1D                       RR L
```

```
2576 F6E0                      OR 0E0H
2578 67                        LD H,A          ;HL=Y/8+X/8+FILL CHK SCRN (MUST BE ON 8K PAGE)
2579
2579 79                        LD A,C
257A E607                      AND 07H
257C 5F                        LD E,A          ;BIT OFFSET
257D 1C                        INC E
257E 3E01                      LD A,01H
2580
2580 0F          FRTMASK:      RRCA
2581 1D                        DEC E
2582 20FC                      JR NZ,FRTMASK
2584
2584 57                        LD   D,A
2585 A6                        AND  (HL)
2586 20DB                      JR   NZ,UNSTK   ;JR IF POINT IS FILLED ALREADY
2588
2588 A6          FINDEDG:      AND  (HL)
2589 200D                      JR   NZ,FOUNDEDG
258B
258B CB02                      RLC  D          ;MOVE BIT MASK LEFT - CARRY IF OVERFLOW
258D 0D                        DEC C           ;MOVE X COORD LEFT
258E 7A                        LD   A,D
258F 30F7                      JR   NC,FINDEDG ;JR UNLESS NEED TO ALTER ADDR
2591
2591 2D                        DEC L           ;1 BYTE LEFT
2592 0C                        INC C
2593 2804                      JR Z,FOUNDEDG1  ;JR IF WE FELL OFF LHS SCREEN EDGE
2595
2595 0D                        DEC C
2596 18F0                      JR FINDEDG
2598
2598 0C          FOUNDEDG:     INC C           ;MAKE CURRENT POSN ONE
2599
2599 CB0A        FOUNDEDG1:    RRC  D          ; PIXEL RIGHT OF BLOCKAGE
259B 3001                      JR   NC,NBACK   ;JR IF NO ADDRESS CHANGE NEEDED
259D
259D 2C                        INC  L
259E
259E 22CA5A      NBACK:        LD (TEMPW2),HL
25A1 E5                        PUSH HL
25A2 DDE1                      POP  IX         ;POSN ALSO IN IX
25A4 3A405A                    LD A,(MODE)
25A7 D603                      SUB 3
25A9 2802                      JR Z,FILLFLGS
25AB
25AB 3E80                      LD A,80H
25AD
25AD 5F          FILLFLGS:     LD   E,A        ;INIT FLAGS. BIT 7=1 IF MODE 2
25AE C5                        PUSH BC         ;XY
25AF
25AF D9                        EXX
25B0 C1                        POP BC
25B1 78                        LD A,B          ;A=Y
25B2 87                        ADD A,A
25B3 87                        ADD A,A
25B4 87                        ADD A,A
25B5 F680                      OR 80H          ;A=80-F8 (8*Y MOD 16)
25B7 4F                        LD C,A
25B8 2650                      LD H,FILBUFF/256
25BA D9                        EXX
25BB
25BB 1808                      JR   FILLC
25BD
25BD 0C          FILLB:        INC C
25BE 28A3                      JR   Z,UNSTK    ;JP IF X COORD PAST RHS
25C0
25C0 CB0A                      RRC  D          ;ROTATE MASK
25C2 DC3E26                    CALL C,MVRGT    ;CALL IF TIME TO MOVE 1 BYTE RIGHT
25C5
25C5 2ACA5A      FILLC:        LD   HL,(TEMPW2)
25C8 7A                        LD   A,D
25C9 A6                        AND  (HL)
25CA 2097                      JR   NZ,UNSTK   ;JR IF PIXEL FILLED
25CC
25CC 7A                        LD   A,D        ;ELSE
25CD B6                        OR   (HL)       ;FILL
25CE 77                        LD   (HL),A     ;PIXEL IN CHECK SCREEN
25CF 78                        LD A,B          ;Y
25D0
25D0 D9                        EXX
25D1 37                        SCF
25D2 1F                        RRA
25D3 57                        LD D,A
25D4 D9                        EXX
25D5
25D5 79                        LD A,C          ;X COORD
25D6 CB7B                      BIT 7,E
25D8
25D8 D9                        EXX
25D9 1F                        RRA
25DA 2009                      JR NZ,FILLM2    ;JR IF MODE 2 (NZ SET BY BIT 7,E)
25DC
25DC 5F                        LD E,A          ;DE'=REAL SCREEN ADDR
25DD 060F                      LD B,0FH
25DF 3024                      JR NC,FILLOEC   ;JR IF EVEN X COORD
25E1
25E1 06F0                      LD B,0F0H
25E3 1820                      JR FILLOEC
25E5
25E5 CB2F        FILLM2:       SRA A           ;KEEP BIT 7 THE SAME
25E7 E6BF                      AND 0BFH        ;1011 1111. BIT 7 FROM Y IS KEPT, BITS 5-0=X/4
25E9 5F                        LD E,A
```

```
25EA 3AC85A                LD A,(TEMPW1)    ;OFFSET FROM LHS
25ED 83                    ADD A,E
25EE 5F                    LD E,A           ;DE'=REAL SCREEN ADDR
25EF D9                    EXX
25F0
25F0 79                    LD A,C           ;X COORD
25F1
25F1 D9                    EXX
25F2 063F                  LD B,3FH         ;MASK FOR BITS TO ZERO
25F4 E603                  AND 3
25F6 280C                  JR Z,FILLM2M
25F8
25F8 06CF                  LD B,0CFH
25FA 3D                    DEC A
25FB 2807                  JR Z,FILLM2M
25FD
25FD 06F3                  LD B,0F3H
25FF 3D                    DEC A
2600 2802                  JR Z,FILLM2M
2602
2602 06FC                  LD B,0FCH
2604
2604 7B         FILLM2M:   LD A,E
2605
2605 E607       FILLOEC:   AND 07H          ;A=PHYSICAL SCREEN COLUMN MOD 8
2607 B1                    OR C             ;GET BITS DETERMINED BY Y COORD
2608 6F                    LD L,A           ;HL' IS NOW COMPLETE FILBUFF ADDR
2609 1A                    LD A,(DE)
260A AE                    XOR (HL)
260B A0                    AND B            ;MIX BITS FROM FILBUFF PATTERN AND SCREEN
260C AE                    XOR (HL)         ; ACCORDING TO MASK B
260D 12                    LD (DE),A
260E D9                    EXX
260F
260F DD7EE0                LD   A,(IX-32)   ;A=BYTE ABOVE
2612 CB43                  BIT  0,E
2614 200A                  JR   NZ,FILLE
2616
2616 A2                    AND  D
2617 200C                  JR   NZ,FILLF    ;JR IF BIT FILLED
2619
2619 05                    DEC  B           ;GET Y OF LOCATION ABOVE
261A C5                    PUSH BC          ;SAVE COORDS OF PLACE ABOVE
261B 04                    INC  B
261C CBC3                  SET  0,E         ;'ABOVE IS FILLED'
261E 1805                  JR   FILLF
2620
2620 A2         FILLE:     AND  D
2621 2802                  JR   Z,FILLF    ;JR IF ABOVE NOT FILLED
2623
2623 CB83                  RES  0,E         ;'ABOVE NOT FILLED'
2625
2625 DD7E20     FILLF:     LD   A,(IX+32)   ;GET BYTE BELOW
2628 CB4B                  BIT  1,E
262A 200A                  JR   NZ,FILLG
262C
262C A2                    AND  D
262D 208E                  JR   NZ,FILLB    ;JR IF BELOW IS FILLED
262F
262F 04                    INC  B           ;INC Y
2630 C5                    PUSH BC          ;SAVE COORDS OF PLACE BELOW
2631 05                    DEC  B
2632 CBCB                  SET  1,E         ;'BELOW IS FILLED'
2634 1887                  JR   FILLB
2636
2636 A2         FILLG:     AND  D
2637 2802                  JR   Z,FILBH     ;JR IF NOT FILLED BELOW
2639
2639 CB8B                  RES  1,E         ;'BELOW IS NOT FILLED'
263B
263B C3BD25     FILBH:     JP   FILLB
263E
263E            ;**************************************************************************
263E            ;FILL SR MOVE RIGHT
263E
263E DD2ACA5A   MVRGT:     LD   IX,(TEMPW2)
2642 DD23                  INC  IX
2644 DD22CA5A              LD   (TEMPW2),IX   ;MOVE 1 COL RIGHT IN CHECK SCREEN
2648
2648 DD7E20                LD   A,(IX+32)   ;BYTE BELOW
264B CB4B                  BIT  1,E
264D 2004                  JR   NZ,MVRGT1
264F
264F 3C                    INC  A
2650 2803                  JR   Z,MVRGT2
2652
2652 C9                    RET
2653
2653 A7         MVRGT1:    AND  A
2654 C0                    RET  NZ
2655
2655 DD7EE0     MVRGT2:    LD   A,(IX-32)
2658
2658 CB43                  BIT  0,E
265A 2004                  JR   NZ,MVRGT3
265C
265C 3C                    INC  A
265D 2803                  JR   Z,MVRGT4
265F
265F C9                    RET
2660
2660 A7         MVRGT3:    AND  A
2661 C0                    RET  NZ
```

```
2662
2662 2ACA5A     MVRGT4:       LD   HL,(TEMPW2)
2665 7E                      LD   A,(HL)
2666 A7                      AND  A
2667 C0                      RET  NZ          ;RET IF NOT SIMPLE BLANK IN CHK SCREEN
2668
2668 36FF                    LD   (HL),0FFH   ;ELSE FILL 8 PIX AT ONCE
266A
266A 78                      LD A,B
266B
266B D9                      EXX
266C 37                      SCF
266D 1F                      RRA
266E 57                      LD D,A
266F D9                      EXX
2670
2670 79                      LD A,C           ;X
2671 3C                      INC A            ;MOVE RIGHT ONTO 1ST. PIXEL OF 8-PIX BLOCK
2672 CB7B                    BIT 7,E
2674
2674 D9                      EXX
2675 1F                      RRA
2676 2810                    JR Z,FILLM3B
2678
2678 CB2F                    SRA A            ;KEEP BIT 7 THE SAME
267A E6BF                    AND 0BFH         ;1011 1111. BIT 7 FROM Y IS KEPT, BITS 5-0=X/4
267C 5F                      LD E,A
267D 3AC85A                  LD A,(TEMPW1)    ;OFFSET FROM LHS
2680 83                      ADD A,E
2681
2681 5F                      LD E,A           ;DE'=REAL SCREEN ADDR
2682 E607                    AND 07H          ;A=PHYSICAL SCREEN COLUMN MOD 8
2684 B1                      OR C             ;GET BITS DETERMINED BY Y COORD
2685 6F                      LD L,A           ;HL' IS NOW COMPLETE FILBUFF ADDR
2686 180D                    JR FILLBC        ;COPY 2 BYTES FROM PATTERN IN MODE 2
2688
2688 5F          FILLM3B:    LD E,A           ;DE'=REAL SCREEN ADDR
2689 E607                    AND 07H          ;A=PHYSICAL SCREEN COLUMN MOD 8
268B B1                      OR C             ;GET BITS DETERMINED BY Y COORD
268C 6F                      LD L,A           ;HL' IS NOW COMPLETE FILBUFF ADDR
268D                                          ;COPY 4 BYTES FROM FILLBUF PATTERN IN M3
268D 7E                      LD A,(HL)
268E 12                      LD (DE),A
268F 1C                      INC E
2690 2C                      INC L
2691
2691 7E                      LD A,(HL)
2692 12                      LD (DE),A
2693 1C                      INC E
2694 2C                      INC L
2695
2695 7E          FILLBC:     LD A,(HL)
2696 12                      LD (DE),A
2697 1C                      INC E
2698 2C                      INC L
2699
2699 7E                      LD A,(HL)
269A 12                      LD (DE),A        ;COPY 8 PIXELS
269B D9                      EXX
269C
269C 79                      LD A,C
269D C608                    ADD A,8
269F 2803                    JR Z,MVRGT5    ;JR IF AT RHS OF SCREEN
26A1
26A1 4F                      LD   C,A
26A2 189A                    JR   MVRGT
26A4
26A4 E1          MVRGT5:     POP  HL          ;JUNK RET ADDR
26A5 C36325                  JP   UNSTK
26A8
26A8             ;*********************************************************************
26A8             ;TRANSCR - COPY MODE 3 SCREEN TO CHECK SCREEN (8000-DFFF -> E000-F800)
26A8             ;      OR - COPY PART OF MODE 2 SCREEN TO CHECK SCREEN
26A8             ;SET BITS IN CHECK SCREEN THAT DON'T MATCH FILL ORIGIN COLOUR
26A8             ;ENTRY: HL=COORDS OF FILL ORIGIN
26A8             ;TRANSODD ALSO USED AS ENTRY BY COPY, WITH A=BACKGROUND COLOUR
26A8             ;USES HL,DE,BC,HL',DE',BC'
26A8
26A8 37          TRANSCR:    SCF
26A9 4D                      LD C,L           ;X
26AA CB1C                    RR H
26AC CB1D                    RR L
26AE 3A405A                  LD A,(MODE)
26B1 FE03                    CP 3
26B3 2817                    JR Z,TRANSL1
26B5
26B5 7D                      LD A,L
26B6 CB2F                    SRA A            ;KEEP BIT 7 THE SAME
26B8 E6BF                    AND 0BFH         ;1011 1111. BIT 7 FROM Y IS KEPT, BITS 5-0=X/4
26BA 6F                      LD L,A
26BB 3AC85A                  LD A,(TEMPW1)    ;OFFSET FROM LHS
26BE 85                      ADD A,L
26BF 6F                      LD L,A           ;HL=M2 SCREEN ADDR
26C0 79                      LD A,C           ;X
26C1 E603                    AND 3
26C3 3C                      INC A
26C4 47                      LD B,A
26C5 7E                      LD A,(HL)
26C6
26C6 07          TRAM2RLP:   RLCA
26C7 07                      RLCA
26C8 10FC                    DJNZ TRAM2RLP
26CA
```

```
26CA 1809                      JR TRANSODD
26CC
26CC 7E          TRANSL1:      LD A,(HL)
26CD CB41                      BIT 0,C
26CF 2004                      JR NZ,TRANSODD
26D1
26D1 07                        RLCA
26D2 07                        RLCA
26D3 07                        RLCA
26D4 07                        RLCA
26D5
26D5             ;ENTRY USED BY MODE 2/3 SCREEN DUMP WITH A=BACKGROUND COLOUR
26D5
26D5 D9          TRANSODD:     EXX
26D6 2100E0                    LD HL,0E000H     ;HL' IS DEST SCREEN PTR FOR FILL CHECKING
26D9 1120C0                    LD DE,0C020H     ;D'=192 SCANS, E'=32 BYTES/SCAN IN CHECK SCREEN
26DC D9                        EXX
26DD
26DD 210080                    LD HL,8000H      ;REAL SCREEN START
26E0
26E0 E60F        CHARCOMP2:    AND 0FH          ;GET COLOUR TO SET TO ZEROS IN CHECK SCREEN
26E2 57                        LD D,A
26E3 07                        RLCA
26E4 07                        RLCA
26E5 07                        RLCA
26E6 07                        RLCA             ;GET DESIRED INK IN MS NIBBLE
26E7 5F                        LD E,A
26E8 3A405A                    LD A,(MODE)
26EB FE03                      CP 3
26ED 2833                      JR Z,TRANSM3
26EF
26EF 3AC85A                    LD A,(TEMPW1)
26F2 6F                        LD L,A           ;OFFSET REAL SCREEN ADDR
26F3 7A                        LD A,D
26F4 E603                      AND 3
26F6 57                        LD D,A
26F7
26F7 D9                        EXX
26F8
26F8 4B          TRAM2DLP:     LD C,E           ;RELOAD BYTES/SCAN COUNTER C'
26F9
26F9 0602        TRAM2CLP:     LD B,2
26FB
26FB D9          TRAM2OLP:     EXX
26FC 5E                        LD E,(HL)        ;E=DATA FROM M2 SCRN
26FD 0604                      LD B,4           ;4 DOUBLE BITS/BYTE
26FF
26FF 7B          TRAM2ILP:     LD A,E
2700 07                        RLCA
2701 07                        RLCA
2702 5F                        LD E,A
2703 E603                      AND 3
2705 92                        SUB D
2706 FE01                      CP 1
2708 CB11                      RL C             ;BITS IN C SHOW IF PIXEL IN M2 SCRN IS 'INK'
270A 10F3                      DJNZ TRAM2ILP
270C
270C 23                        INC HL           ;NEXT M2 SCREEN BYTE
270D 79                        LD A,C           ;GET DATA IN CASE THIS IS SECOND BYTE
270E D9                        EXX
270F 10EA                      DJNZ TRAM2OLP    ;DO 2 (B') M2 SCREEN BYTES/CHECK SCREEN BYTE
2711
2711 2F                        CPL              ;PIX THAT MATCH ORIGIN=0
2712 77                        LD (HL),A
2713 23                        INC HL
2714 0D                        DEC C            ;DO REQUIRED BYTES ACROSS (1 OR 32)
2715 20E2                      JR NZ,TRAM2CLP
2717
2717 D9                        EXX
2718 014000                    LD BC,64         ;DROP TO NEXT SCAN IF FILL, IRREL IF CHARCOMP
271B 09                        ADD HL,BC
271C D9                        EXX
271D
271D 15                        DEC D            ;DO D' SCANS
271E 20D8                      JR NZ,TRAM2DLP
2720
2720 D9                        EXX
2721
2721 C9                        RET
2722
2722 0601        TRANSM3:      LD B,1         ;CP WITH THIS TO SET CY IF A=0
2724 D9                        EXX
2725
2725 4B          TRANSDLP:     LD C,E           ;RELOAD BYTES/SCAN COUNTER C'
2726
2726 D9          TRANSCLP:     EXX
2727 48                        LD C,B           ;C=1. CY WHEN BIT ROTATED OUT AFTER 8 ROTS.
2728
2728 7E          TRANSBLP:     LD A,(HL)
2729 E6F0                      AND 0F0H
272B 93                        SUB E            ;GET A=ZERO IF PIXEL MATCHES ORIGIN INK
272C B8                        CP B             ;SET CARRY IF A=0
272D CB11                      RL C
272F 7E                        LD A,(HL)
2730 23                        INC HL
2731 E60F                      AND 0FH
2733 92                        SUB D
2734 B8                        CP B
2735 CB11                      RL C
2737 D22827                    JP NC,TRANSBLP
273A
273A 79                        LD A,C
273B 2F                        CPL              ;PIX THAT MATCH ORIGIN=0
```

```
273C
273C D9                         EXX
273D 77                         LD (HL),A        ;MOVE TO CHECK SCREEN AT (HL')
273E 23                         INC HL
273F 0D                         DEC C
2740 20E4                       JR NZ,TRANSCLP   ;DO C' BYTES/SCAN IN CHECK SCREEN
2742
2742 15                         DEC D            ;AND D' SCANS
2743 20E0                       JR NZ,TRANSDLP
2745
2745 D9                         EXX
2746 C9                         RET
2747
2747                ;*********************************************************************************
2747                ;THIS ROUTINE USED BY SCREEN$ TO COMPRESS A MODE 2/3 CHARACTER TO STANDARD FORM
2747                ;IN SCRNBUF. ENTRY: HL PTS TO POSN IN M2/3 SCREEN, DE PTS TO 8-BYTE BUFFER,
2747                ;CY=6-BIT CHARS, NZ/Z=ODD/EVEN.  USES AF,HL,DE,BC,HL',DE',BC'
2747
2747 F5             CHARCOMP:       PUSH AF
2748 CDA83F                         CALL SPSSR      ;SELECT SCREEN, ROM1 OFF
274B F1                             POP AF
274C D5                             PUSH DE
274D 3007                           JR NC,CHCM2     ;JR IF NOT 6-PIX CHARS
274F
274F 7E                             LD A,(HL)       ;SCREEN DATA
2750 280E                           JR Z,CHCM3      ;JR IF EVEN COLUMN - TOP LHS PIX IN BITS 7,6
2752
2752 0F                             RRCA            ;ELSE DATA IN BITS 3,2
2753 0F                             RRCA
2754 180C                           JR CHCM4        ;NOW IN 1,0
2756
2756 3A405A         CHCM2:          LD A,(MODE)
2759 FE02                           CP 2
275B 7E                             LD A,(HL)       ;SCREEN DATA
275C 2802                           JR Z,CHCM3      ;JR IF MODE 2 - TOP LHS PIXEL DATA IS BITS 7,6
275E
275E 07                             RLCA            ;ELSE IT IS BITS 7,6,5,4
275F 07                             RLCA
2760
2760 07             CHCM3:          RLCA
2761 07                             RLCA            ;DATA NOW IN RH 2 OR 4 BITS
2762
2762 D9             CHCM4:          EXX               ;A=BACKGROUND COLOUR - (TOP LHS PIXEL COLOUR)
2763 E1                             POP HL            ;BUFFER FOR 8-BYTE COMPRESSED FORM FOR SCREEN$
2764 37                             SCF
2765
2765                ;CALLED BY GRAPHICS COPY WITH NC
2765
2765 47             GRCOMP:         LD B,A
2766 D4A83F                         CALL NC,SPSSR   ;SELECT SCREEN
2769 78                             LD A,B
276A 0608                           LD B,8            ;8 SCANS
276C
276C C5             CHARCLP:        PUSH BC
276D F5                             PUSH AF           ;BG COLOUR
276E 110101                         LD DE,0101H       ;1 SCAN AT A TIME, 1 BYTE ACROSS IN RESULT
2771
2771 D9                             EXX
2772 E5                             PUSH HL
2773 22C85A                         LD (TEMPW1),HL    ;OFFSET FOR SCREEN SOURCE ADDR (USED BY MODE 2
2776                                                  ;FILL, AND CHARCOMP2 NEEDS IT)
2776 CDE026                         CALL CHARCOMP2    ;DO A SCAN
2779 E1                             POP HL
277A 7D                             LD A,L            ;NOW DROP TO NEXT SCAN
277B C680                           ADD A,128         ;SCAN LEN
277D 6F                             LD L,A
277E 3001                           JR NC,CHARCNINC
2780
2780 24                             INC H
2781
2781 D9             CHARCNINC:      EXX
2782 F1                             POP AF            ;BG COLOUR
2783 C1                             POP BC
2784 10E6                           DJNZ CHARCLP
2786
2786 C3BF3F                         JP RCURPR
2789
2789
2789                ;CRDFID.SAM
2789                ;GET FIDDLED COORDS - FORCE FATPIX
2789
2789                ;GET FIDDLED FAT COORDS - USED BY GET, ROLL/SCROLL
2789
2789 CD9327         GTFIDFCDS:      CALL GTFCOORDS
278C D0                             RET NC            ;RET IF FAT PIX
278D
278D CB1C                           RR H
278F CB1D                           RR L
2791 4D                             LD C,L            ;HALVE X, MOVE TO C
2792 C9                             RET
2793
2793                ;GET FIDDLED COORDS
2793
2793 CDFB27         GTFCOORDS:      CALL COORDFID     ;APPLY OFFSETS AND RANGES
2796
2796                ;UNSTACK COORDS.
2796                ;ENTRY: X,Y ON FPCS
2796
2796                ;EXIT: IF THIN PIX, HL=X COORD, CHECK FOR 0-511, ELSE C=X, CHECKED FOR 0-255.
2796                ;B IS ALWAYS THE Y COORD, CORRECTED FROM -16 TO 175 ON FPCS TO 0-191, THEN
2796                ;INVERTED SO 0 AT THE TOP. CY SET IF THIN PIX
2796
2796 CDB027         USCOORDS:       CALL USYCOORD
```

```
2799 F5                     PUSH AF             ;Y
279A CD2E1D                 CALL GETINT         ;HL=X COORD. BC=HL
279D 22CA5A                 LD (TEMPW2),HL      ;SAVE FOR 'RECORD' TO USE
27A0 3A4D5A                 LD A,(THFATT)       ;A=0 IF THIN PIX
27A3 A7                     AND A
27A4 7C                     LD A,H              ;A=X MSB
27A5 C1                     POP BC              ;B=Y
27A6 2003                   JR NZ,THCKCHK
27A8
27A8 FE02                   CP 2                ;CHECK X MSB VS 2 - MUST BE ZERO OR 1
27AA D8                     RET C               ;CY SIGNALS THIN PIX
27AB                                            ;A MUST BE 2-FF HERE
27AB
27AB 4D         THCKCHK:    LD C,L              ;C=X
27AC A7                     AND A
27AD C8                     RET Z               ;NC
27AE
27AE 180D                   JR IOORERR1     ;X MSB MUST BE 0 UNLESS THIN PIX
27B0
27B0
27B0            ;UNSTACK Y COORD TO A, CHECK AND CONVERT TO 0-191 SCALE
27B0
27B0 CD651D     USYCOORD:   CALL FPTOA          ;A (AND C)=ABS Y. Z IF POSITIVE
27B3 3808                   JR C,IOORERR1
27B5
27B5 3A5C5A                 LD A,(ORGOFF)       ;16 IF HEIGHT=8
27B8 2005                   JR NZ,ADJNEG        ;JR IF Y IS -VE
27BA
27BA 81                     ADD A,C
27BB 3005                   JR NC,ADJOK1        ;ADD 16 SO 0-175 BECOME 16-191
27BD
27BD CF         IOORERR1:   RST 08H
27BE 1E                     DB 30
27BF
27BF 91         ADJNEG:     SUB C               ;(16 IF HEIGHT=8)-ABS Y
27C0 38FB                   JR C,IOORERR1       ;IF Y IS NEGATIVE, ABS Y MUST BE <=ORGOFF
27C2                                            ;-1 TO -16 NOW 15 TO 0
27C2 47         ADJOK1:     LD B,A
27C3 3EBF                   LD A,191
27C5 90                     SUB B
27C6 38F5                   JR C,IOORERR1
27C8
27C8 32CE5A                 LD (TEMPB1),A       ;A=0-191. SAVE FOR 'RECORD' TO USE
27CB C9                     RET
27CC
27CC            ;TWONUMS
27CC            ;UNSTACK X AND Y DISPS. B=Y DISP, C=X DISP, D=SGN Y, E=SGN X (SGN = 01/FF)
27CC            ; OR IF CY, THIN PIX AND HL=X DISP
27CC
27CC CD651D     TWONUMS:    CALL FPTOA
27CF 38EC                   JR C,IOORERR1       ;ERROR IF ABS Y>255
27D1
27D1 06FF                   LD B,0FFH           ;-VE. REVERSE SGN BECAUSE Y COORD REVERSES
27D3 2802                   JR Z,POSBYTE
27D5
27D5 0601                   LD B,01H            ;+VE
27D7
27D7 C5         POSBYTE:    PUSH BC             ;B=SGN Y, C=Y
27D8 CD3B1D                 CALL FPTOBC         ;BC=X. Z IF +VE. CY IF OOR
27DB 38E0                   JR C,IOORERR1
27DD
27DD 2601                   LD H,1              ;+VE
27DF 2802                   JR Z,POSINT
27E1
27E1 26FF                   LD H,0FFH           ;-VE. H=SGN X
27E3
27E3 D1         POSINT:     POP DE              ;D=SGN Y, E=Y
27E4 3A4D5A                 LD A,(THFATT)       ;THIN/FAT TEMP. ONLY SAYS THIN IF MODE 2.
27E7 A7                     AND A
27E8 78                     LD A,B              ;X MSB
27E9 43                     LD B,E              ;B=Y
27EA 5C                     LD E,H              ;E=SGN X
27EB 2804                   JR Z,THINNUMS
27ED
27ED A7                     AND A
27EE 20CD                   JR NZ,IOORERR1      ;X MSB MUST BE ZERO IF FAT PIX IN USE
27F0
27F0 C9                     RET                 ;B=Y, C=X, D=SGN Y, E=SGN X. NC
27F1
27F1 69         THINNUMS:   LD L,C
27F2 67                     LD H,A              ;HL=X
27F3 FE02                   CP 2
27F5 30C6                   JR NC,IOORERR1
27F7
27F7 C9                     RET                 ;HL=X, E=SGN X, D=SGN Y, B=Y. CY SHOWS THIN STATUS
27F8                                            ;BC MUST BE 01FF OR LESS
27F8
27F8            ;COORDINATE FIDDLE FOR PLOT, DRAW TO, CIRCLE, PUT,GRAB. ENTRY WITH X,Y ON FPCS
27F8
27F8            ;DRAW RANGE FIDDLE ONLY
27F8
27F8 37         DRCOORDFD: SCF                  ;'DON'T APPLY XOS'
27F9 180A                   JR RGFIDEN
27FB
27FB            ;RANGE AND OFFSET FIDDLE. USED BY E.G. PLOT.
27FB
27FB AF         COORDFID:   XOR A               ;NC='OFFSET'
27FC 47                     LD B,A
27FD 4F                     LD C,A              ;BC=0=NORMAL YOS
27FE 08                     EX AF,AF'
27FF 1E39                   LD E,YOSDISP
2801 CD3228                 CALL PSEUDOSR       ;ADD YOS IF NON-NORMAL (<>ZERO)
2804 AF                     XOR A               ;NC='APPLY XOS'
```

```
2805
2805 F5          RGFIDEN:   PUSH AF
2806 37                     SCF                 ;'RANGE'
2807 08                     EX AF,AF'
2808 1E43                   LD E,YRGDISP
280A 01C000                 LD BC,192
280D CD3228                 CALL PSEUDOSR       ;APPLY YRG UNLESS IT IS 192
2810 CD8A18                 CALL SWOP12         ;GET X COORD TO STACK TOP
2813 F1                     POP AF
2814 3809                   JR C,DOXRG          ;NC MEANS APPLY XOS. A=0
2816
2816 47                     LD B,A
2817 4F                     LD C,A
2818 08                     EX AF,AF'
2819 1E4D                   LD E,XOSDISP
281B CD3228                 CALL PSEUDOSR       ;ADD XOS IF NON-NORMAL (<>ZERO)
281E 37                     SCF                 ;'RANGE'
281F
281F 08          DOXRG:     EX AF,AF'
2820 010002                 LD BC,512
2823 3A4D5A                 LD A,(THFATT)
2826 A7                     AND A
2827 2801                   JR Z,GXRANGE        ;RANGE=512 IF THIN
2829
2829 05                     DEC B               ;BC=256
282A
282A 1E57        GXRANGE:   LD E,XRGDISP
282C CD3228                 CALL PSEUDOSR       ;APPLY XRG UNLESS IT IS NORMAL
282F C38A18                 JP SWOP12           ;GET Y TO TOP OF FPCS
2832
2832             ;PSEUDO VARIABLE SUBROUTINE
2832             ;ENTRY: E=OFFSET TO PSEUDO-VAR XOS/YOS/XRG/YRG. F'=CY IF RANGE, NC IF OFFSET
2832             ;BC=NORMAL VALUE OF VAR.
2832             ;ACTION: APPLY PS IF NON-NORMAL
2832
2832 DBFB        PSEUDOSR:  IN A,(URPORT)
2834 F5                     PUSH AF
2835 CD1F1F                 CALL ADDRNV
2838 1600                   LD D,0
283A 19                     ADD HL,DE
283B 54                     LD D,H
283C 5D                     LD E,L              ;DE PTS TO PS START
283D CD5A28                 CALL CHECKPS        ;SEE IF PSEUDO VAR=BC (NORMAL)
2840 CA453C                 JP Z,PPORT          ;JR IF PSEUDO VAR=NORMAL, DON'T APPLY IT.
2843                        ;(POP AF, OUT, RET)
2843 C5                     PUSH BC             ;NORMAL VALUE
2844 EB                     EX DE,HL
2845 CD201D                 CALL HLTOFPCS       ;ELSE STACK PSEUDO-VAR
2848 C1                     POP BC
2849 F1                     POP AF
284A D3FB                   OUT (URPORT),A
284C 08                     EX AF,AF'
284D 3803                   JR C,APPLYRG
284F
284F EF                     DB CALC
2850 01                     DB ADDN             ;ADD OFFSET
2851 34                     DB EXIT2
2852
2852 CDDD1C      APPLYRG:   CALL STACKBC
2855
2855 EF                     DB CALC
2856 06                     DB SWOP
2857 05                     DB DIVN             ;NORM/RG
2858 00                     DB MULT             ;COORD*NORM/RG
2859 34                     DB EXIT2
285A
285A
285A 7E          CHECKPS:   LD A,(HL)
285B 23                     INC HL
285C B6                     OR (HL)
285D C0                     RET NZ              ;NORMAL VALUES ALWAYS START 00 00
285E
285E 23                     INC HL
285F 79                     LD A,C
2860 BE                     CP (HL)
2861 C0                     RET NZ
2862
2862 23                     INC HL
2863 78                     LD A,B
2864 BE                     CP (HL)
2865 C9                     RET
2866
2866             ;GRAPHICS RECORD
2866             ;PLOT, DRAW TO, CIRCLE: CALL HERE WITH CMD CODE IN E, TEMPB1=Y, TEMPW2=X
2866             ;CIRCLE HAS RADIUS IN B REG. REL. DRAW USES E,C,D,B FOR SGN X,X,SGN Y,Y.
2866             ;PEN, OVER, PAUSE, CLS USE E,C FOR CMD CODE, PARAM
2866
2866             ;E=FE/00 IF FAT REL. DRAW
2866             ;E=1      FAT PLOT
2866             ;E=2      FAT DRAW TO
2866             ;E=3      CIRCLE
2866             ;E=4      OVER
2866             ;E=5      PEN
2866             ;E=6      CLS
2866             ;E=7      PAUSE
2866
2866 3A815B      GRAREC:    LD A,(GRARF)
2869 A7                     AND A
286A C8                     RET Z               ;RET IF RECORD OFF
286B
286B 3A745B                 LD A,(CURCMD)
286E FEBB                   CP 187              ;PRINT
2870 C8                     RET Z               ;AVOID RECORD OF E.G. PRINT PEN 5;
```

```
2871
2871 E5                    PUSH HL
2872 D5                    PUSH DE
2873 C5                    PUSH BC
2874 7B                    LD A,E
2875 21004F                LD HL,INSTBUF
2878 3D                    DEC A               ;SGN FE/00 GOES TO FD/FF
2879 FEFD                  CP 0FDH
287B 3817                  JR C,GRAR6          ;JR IF NOT REL. DRAW
287D
287D 3C                    INC A
287E 2804                  JR Z,GRAR4          ;JR IF +VE X SGN (E=0)
2880
2880 1C                    INC E               ;E=FF
2881 AF                    XOR A
2882 91                    SUB C
2883 4F                    LD C,A              ;NEGATE X DISP
2884
2884 14          GRAR4:    INC D               ;Y SGN
2885 2805                  JR Z,GRAR5          ;JR IF -VE Y SGN.. Y IS REALLY +VE - FIDDLED BY
2887                                           ;TWONUMS BECAUSE OF Y AXIS REVERSAL
2887 16FF                  LD D,0FFH
2889 AF                    XOR A
288A 90                    SUB B
288B 47                    LD B,A              ;NEGATE Y DISP
288C
288C 73          GRAR5:    LD (HL),E
288D 78                    LD A,B
288E 42                    LD B,D
288F 57                    LD D,A              ;SWOP B AND D
2890 1E03                  LD E,3              ;PRETEND 'CIRCLE'
2892 180D                  JR GRAR66
2894
2894 FE03        GRAR6:    CP 3
2896 3008                  JR NC,GRAR65
2898
2898 3ACA5A      GRAR62:   LD A,(TEMPW2)
289B 4F                    LD C,A              ;X
289C 3ACE5A                LD A,(TEMPB1)       ;Y
289F 47                    LD B,A              ;COORDS TAKEN FROM STORES USED BY CRDFID
28A0
28A0 73          GRAR65:   LD (HL),E           ;BLITZ CODE
28A1
28A1 23          GRAR66:   INC HL
28A2 71                    LD (HL),C
28A3 23                    INC HL
28A4 70                    LD (HL),B           ;MAY BE JUNK
28A5 23                    INC HL
28A6 72                    LD (HL),D           ;MAY BE JUNK
28A7 7B                    LD A,E
28A8 3D                    DEC A
28A9 0E03                  LD C,3
28AB FE02                  CP 2
28AD 3806                  JR C,GRAR7          ;JR IF PLOT OR DRAWTO - 3 BYTES
28AF
28AF 0E04                  LD C,4
28B1 2802                  JR Z,GRAR7          ;JR IF CIRCLE OR REL DRAW - 4 BYTES
28B3
28B3 0E02                  LD C,2
28B5
28B5 2A515C      GRAR7:    LD HL,(CURCHL)
28B8 E5                    PUSH HL
28B9 C5                    PUSH BC
28BA 3E10                  LD A,16
28BC CD6311                CALL SETSTRM        ;STREAM 16 - TO STRING
28BF C1                    POP BC
28C0 0600                  LD B,0
28C2 11004F                LD DE,INSTBUF
28C5 CD1300                CALL PRINTSTR       ;O/P BC FROM DE TO STRING
28C8 E1                    POP HL
28C9 22515C                LD (CURCHL),HL      ;STREAM 16 DOESN'T SET ANY FLAGS, SO NO NEED
28CC C1                    POP BC              ;TO CALL CHAN-FLAG TO RESET THEM.
28CD D1                    POP DE
28CE E1                    POP HL
28CF C9                    RET
28D0
28D0
28D0                       INCLUDE GRABPUT.SAM    ;GRAB, PUT, FARLDIR, STRMOV
28D0             ;GRAB.SAM - GRAB A$,X,Y,W,L
28D0
28D0 CDF72C      GRAB:     CALL SYNTAX1        ;ASSESS VAR FOR ASSIGNMENT
28D3 213B5C                LD HL,FLAGS
28D6 CB76                  BIT 6,(HL)
28D8 2003                  JR NZ,GNONSH        ;ERROR IF NUMERIC TYPE
28DA
28DA DF                    RST 18H
28DB FE2C                  CP ","
28DD
28DD C2290D      GNONSH:   JP NZ,NONSENSE
28E0
28E0 CDD33A                CALL SEXPT4NUMS     ;SKIP, EXPECT 4 NUMBERS
28E3 CD153B                CALL CHKEND
28E6
28E6 CD9512                CALL CHKMD23        ;INSIST ON MODE 2 OR 3
28E9 111EC0                LD DE,0C000H+30
28EC CD5F1F                CALL LIMDB          ;LEN TO A, DECED (0-191) (ORIG MUST BE 1-192)
28EF 3C                    INC A
28F0 F5                    PUSH AF             ;A=LEN
28F1 CD2E1D                CALL GETINT
28F4 0B                    DEC BC              ;0->FFFF, 256->255
28F5 78                    LD A,B
28F6 A7                    AND A
28F7 C2391D                JP NZ,IOORERR       ;INSIST ON WIDTH OF 1-256
```

```
28FA
28FA 03                     INC BC              ;BC=1-256
28FB 03                     INC BC              ;IF BC IS EVEN, SET BIT 0 (WHICH WILL BE LOST)
28FC                                            ;IF BC IS ODD,ROUND UP E.G. WIDTH 1->2
28FC CB38                   SRL B               ;B=0
28FE CB19                   RR C                ;C=1-128
2900 D1                     POP DE              ;D=LEN
2901 59                     LD E,C              ;WID
2902 D5                     PUSH DE             ;LEN/WID
2903 CD8927                 CALL GTFIDFCDS      ;UNSTK X,Y TO CB. CHECK FOR LEGALITY
2906 D1                     POP DE              ;LEN/WID
2907 CD1029                 CALL JGRAB
290A CDEB1C                 CALL STKSTOS        ;STACK REGISTERS - STRING
290D C30F2B                 JP ASSIGN
2910
2910              ;D=LEN (PIX), E=WIDTH (BYTES), B=Y, C=X
2910              ;EXIT: DE=START (IN CUSCRNP), BC=LEN (INCLUDES 3 LEADER BYTES)
2910
2910 CDB13F       JGRAB:    CALL SPSS           ;GET CURRENT SCREEN AT 8000+
2913 2100E0                 LD HL,RSBUFF-3      ;E000H
2916 3600                   LD (HL),0           ;CONTROL CODE
2918 23                     INC HL
2919 73                     LD (HL),E           ;WID
291A 23                     INC HL
291B 72                     LD (HL),D           ;LEN
291C CD4729                 CALL GPVARS         ;GET A'=WID,D/B'=LEN, HL=SCRN ADDR, BC=128
291F 7B                     LD A,E
2920 F7                     RST 30H
2921 23EA                   DW CRTBFI
2923 7B                     LD A,E
2924 08                     EX AF,AF'           ;(A' WAS CORRUPTED BY RST 30)
2925 CD3F0B                 CALL RSSTBLK        ;STORE ROLL/SCROLL STORE BLOCK SR
2928 1100E0                 LD DE,RSBUFF-3      ;PT TO CC,W,L
292B ED4BCA5A               LD BC,(TEMPW2)      ;LEN OF DATA
292F 03                     INC BC
2930 03                     INC BC
2931 03                     INC BC              ;ALLOW FOR CC,W,L
2932 DBFB                   IN A,(251)
2934 C3C63F                 JP RCURP
2937
2937              ;*************************************************************************
2937              ;SR TO TRUNCATE LENGTH OF A BLOCK IF HANGING OFF BOTTOM OF SCREEN
2937              ;ENTRY: D=BLOCK LENGTH, B=Y COORD
2937
2937 7A           GPTRUNC:  LD A,D
2938 3D                     DEC A               ;LENGTH-1
2939 80                     ADD A,B             ;ADD Y COORD
293A 3805                   JR C,GPTRUNC2
293C
293C D6C0                   SUB 192             ;ALLOW UP TO 191
293E D8                     RET C               ;RET IF WONT FALL OFF BOTTOM
293F
293F D640                   SUB 40H             ;COMPENSATE FOR FOLLOWING 'ADD'
2941
2941 C640         GPTRUNC2: ADD A,40H           ;GET 'NUMBER OF SCANS HANGING OFF' - 40H+
2943 2F                     CPL                 ;ELSE A=0-3FH ->FF-C0
2944 82                     ADD A,D
2945 57                     LD D,A              ;ADJUST LENGTH
2946 C9                     RET
2947
2947              ;GET/PUT VARS SETTER. ENTRY: E=WID, D=LEN, BC=YX COORDS.
2947              ;EXIT: A'/E=WIDTH, D/B'=LEN, HL=SCRN ADDR, BC=128
2947
2947 7B           GPVARS:   LD A,E
2948 08                     EX AF,AF'           ;A'=WIDTH
2949 CD3729                 CALL GPTRUNC        ;SHORTEN D IF NEEDED.
294C 7A                     LD A,D
294D
294D D9                     EXX
294E 47                     LD B,A              ;B'=LENGTH
294F D9                     EXX
2950
2950 60                     LD H,B              ;Y
2951 69                     LD L,C              ;X
2952 37                     SCF
2953 CB1C                   RR H
2955 CB1D                   RR L                ;GET SCREEN ADDR IN 8000 AREA IN HL
2957 018000                 LD BC,128
295A C9                     RET
295B
295B              ;PUT.SAM
295B              ;*************************************************************************
295B              ;E.G. PUT X,Y,A$. OVER 0-3 ALLOWED, ALSO INVERSE
295B              ;     PUT X,Y,A$,M$ USES M$ AS A MASK FOR 'CLIPPING'.
295B
295B CDF23A       PUT:      CALL SYNTAX9        ;INK/PAPER IRREL, PUT OVER/INVERSE WORK
295E CD9B3A                 CALL EXPTCSTR       ;COMMA, STRING
2961 FE2C                   CP ","
2963 200D                   JR NZ,PUTL1
2965
2965 CD653A                 CALL SSYNTAXA       ;SKIP, EXPECT STRING
2968
2968 2100F0                 LD HL,RSBUFF+0FFDH  ;E003+0FFD= F000H
296B CD2B2A                 CALL PSCHKMHL       ;CHECK MASK STRING AND MOVE TO 2ND HALF OF BUFFER
296E 3E05                   LD A,5              ;'MASKED'
2970 1804                   JR PUTLC
2972
2972 CD153B       PUTL1:    CALL CHKEND
2975
2975 AF                     XOR A
2976
2976 F5           PUTLC:    PUSH AF             ;0=NO MASK, 5=MASKED
2977 2100E0                 LD HL,RSBUFF-3      ;E000H
```

```
297A CD2B2A               CALL PSCHKMHL     ;CHECK STRING AND MOVE TO RSBUFF
297D CD8927               CALL GTFIDFCDS    ;B=Y,C=X.
2980 CDB13F               CALL SPSS
2983 F1                   POP AF
2984 A7                   AND A
2985 2812                 JR Z,PUTL2        ;JR IF NOT MASKED PUT. NC
2987
2987 D9                   EXX
2988 2103F0               LD HL,RSBUFF+1000H ;HL'= F003H - PTR TO MASK STRING
298B D9                   EXX
298C
298C 2A01E0               LD HL,(RSBUFF-2)  ;E001/E002
298F ED5B01F0             LD DE,(RSBUFF+0FFEH) ;F001/F002
2993 ED52                 SBC HL,DE
2995 280F                 JR Z,PUT05        ;A=5 STILL... JR IF W,L THE SAME FOR BOTH STRINGS
2997
2997 CF                   RST 08H
2998 26                   DB 38             ;'PUT mask mismatch'
2999
2999 ED5B545A   PUTL2:    LD DE,(INVERT)    ;E=INVERT, D=GOVERT
299D 7B                   LD A,E            ;0-3
299E B2                   OR D
299F 3E04                 LD A,4
29A1 2803                 JR Z,PUT05        ;OVER 0, INVERSE 0 USES SPECIAL ROUTINE NO. 4
29A3
29A3 7A                   LD A,D
29A4 E603                 AND 3             ;NEUROT...
29A6
29A6 2101E0     PUT05:    LD HL,RSBUFF-2    ;STR PTR (E001H)
29A9 1805                 JR PUT06
29AB
29AB            ;A=PUT TYPE 0-5, B=Y, C=X, HL PTS TO W (BYTES), LEN (PIX), HL' TO MASK DATA
29AB            ;BLOCK TRUNCATED IF HANGS OFF SCREEN BOTTOM
29AB
29AB 5F         JPUT:     LD E,A
29AC CDB13F               CALL SPSS
29AF 7B                   LD A,E            ;**
29B0
29B0 87         PUT06:    ADD A,A
29B1 5F                   LD E,A
29B2 87                   ADD A,A
29B3 87                   ADD A,A           ;*8
29B4 83                   ADD A,E           ;*10
29B5 5F                   LD E,A
29B6 1600                 LD D,0
29B8 FD21E829             LD IY,PUTSRTAB
29BC FD19                 ADD IY,DE         ;IY=ADDR OF SUBROUTINE FOR 'OVER' VARIATIONS
29BE
29BE 5E                   LD E,(HL)         ;GET WIDTH FROM STRING (BYTES)
29BF 23                   INC HL
29C0 56                   LD D,(HL)         ;LEN
29C1 23                   INC HL
29C2 E5                   PUSH HL
29C3 CD4729               CALL GPVARS       ;GET A'/E=WID,D/B'=LEN, HL=SCRN ADDR, BC=128
29C6 79                   LD A,C
29C7 93                   SUB E             ;SCAN LEN-WIDTH=DISP TO NXT SCAN
29C8 D1                   POP DE            ;STRING PTR
29C9 DD21DA29             LD IX,PUTRET      ;SET UP IX TO AVOID CALL OVERHEADS
29CD
29CD D9                   EXX
29CE 5F                   LD E,A            ;E'=DISP
29CF 3A545A               LD A,(INVERT)     ;INVERSE MASK 00/FF TO A'
29D2 08                   EX AF,AF'         ;A=WIDTH
29D3 4F                   LD C,A            ;C'=BLOCK WIDTH
29D4
29D4 D9         PUTSCLP:  EXX
29D5
29D5            ;AT THIS POINT:
29D5            ;HL=SCRN PTR, DE=DATA SRC, A=WIDTH
29D5            ;B'=LEN, C'=WIDTH, E'=DISP TO NEXT SCAN, A'=INVERSE MASK
29D5            ;HL' CAN PT TO MASK STRING
29D5            ;B IS USED AS A WIDTH COUNTER, AND C AS AN INVERSE MASK
29D5
29D5 47                   LD B,A            ;BLOCK WIDTH COUNTER SET UP
29D6 08                   EX AF,AF'
29D7 4F                   LD C,A            ;INVERSE MASK
29D8 FDE9                 JP (IY)           ;JUMP TO XOR, OR, LD or AND LOOP
29DA                                        ;ENTRY WITH DE=SRC, HL=SCRN DEST, B=BYTES, C=INVER
29DA                                        ;EXIT WITH B=0
29DA 79         PUTRET:   LD A,C            ;INVERSE
29DB 08                   EX AF,AF'         ;PROTECT INVERSE MASK IN A'
29DC
29DC D9                   EXX
29DD 7B                   LD A,E            ;DISP TO NEXT SCAN FROM E'
29DE D9                   EXX
29DF
29DF 4F                   LD C,A            ;BC=DISP TO NEXT SCAN (SCAN LEN-BLOCK WIDTH)
29E0 09                   ADD HL,BC         ;DROP 1 SCAN
29E1
29E1 D9                   EXX
29E2 79                   LD A,C            ;GET WIDTH VALUE FROM C'
29E3 10EF                 DJNZ PUTSCLP      ;DEC LENGTH COUNTER, LOOP TILL ALL SCANS DONE
29E5
29E5 C3C63F               JP RCURP
29E8
29E8            ;*************************************************************************
29E8            ;PUT SUBROUTINES FOR OVER 0,1,2,3.
29E8            ;ENTRY WITH DE PTING TO DATA SRC, HL TO SCRN, B=BYTES TO DO, C=INVERSE MASK
29E8            ;IX='RET' ADDR
29E8            ;EXIT WITH B=0.
29E8
29E8            PUTSRTAB:
29E8
```

```
29E8 1A         OVER0LP:   LD A,(DE)         ;DATA FROM STRING
29E9 A9                    XOR C             ;INVERSE MASK
29EA 77                    LD (HL),A
29EB 13                    INC DE
29EC 23                    INC HL
29ED 10F9                  DJNZ OVER0LP
29EF DDE9                  JP (IX)
29F1 00                    NOP               ;MAKE ALL SRs HAVE LENGTH OF 10
29F2
29F2 1A         OVER1LP:   LD A,(DE)         ;DATA FROM STRING
29F3 A9                    XOR C             ;INVERSE MASK
29F4 AE                    XOR (HL)          ;OVER 1
29F5 77                    LD (HL),A
29F6 13                    INC DE
29F7 23                    INC HL
29F8 10F8                  DJNZ OVER1LP
29FA DDE9                  JP (IX)
29FC
29FC 1A         OVER2LP:   LD A,(DE)         ;DATA FROM STRING
29FD A9                    XOR C             ;INVERSE MASK
29FE B6                    OR (HL)           ;OVER 2
29FF 77                    LD (HL),A
2A00 13                    INC DE
2A01 23                    INC HL
2A02 10F8                  DJNZ OVER2LP
2A04 DDE9                  JP (IX)
2A06
2A06 1A         OVER3LP:   LD A,(DE)         ;DATA FROM STRING
2A07 A9                    XOR C             ;INVERSE MASK
2A08 A6                    AND (HL)          ;OVER 3
2A09 77                    LD (HL),A
2A0A 13                    INC DE
2A0B 23                    INC HL
2A0C 10F8                  DJNZ OVER3LP      ;ABOUT 64 T'S PER BYTE (USE INC L,=56)
2A0E DDE9                  JP (IX)
2A10
2A10            ;FASTER VERSION FOR WHEN INVERSE 0, OVER 0
2A10
2A10 EB         NOINVER:   EX DE,HL
2A11 48                    LD C,B
2A12 0600                  LD B,0
2A14 EDB0                  LDIR              ;32 T'S PER BYTE
2A16 EB                    EX DE,HL
2A17 DDE9                  JP (IX)
2A19 00                    NOP
2A1A
2A1A            ;USE 'VALID DATA' MASK PTED TO BY HL'. 1'S=VALID
2A1A            ;MAKE MASK BY: FILL INK 0, BORDER AREA: GRAB A$: FILL INK 15, BORDER AREA
2A1A            ;PUT OVER 1, A$. (MAKES BORDER 1'S, FIGURE 0'S). GRAB A$: PUT INVERSE 1, A$
2A1A            ;GRAB A$. GIVES BORDER 0'S, FIGURE 1'S.
2A1A
2A1A 1A         PMASKLP:   LD A,(DE)         ;DATA FROM STRING
2A1B A9                    XOR C             ;INVERSE MASK
2A1C AE                    XOR (HL)          ;XOR SCRN
2A1D D9                    EXX
2A1E A6                    AND (HL)          ;AND MASK
2A1F 23                    INC HL
2A20 D9                    EXX
2A21 13                    INC DE
2A22 AE                    XOR (HL)
2A23 77                    LD (HL),A
2A24 23                    INC HL
2A25 10F3                  DJNZ PMASKLP      ;ABOUT 96 T'S PER BYTE
2A27 DDE9                  JP (IX)
2A29
2A29 CF         PUTBLKERR: RST 08H
2A2A 25                    DB 37             ;'Invalid PUT block'.
2A2B
2A2B            ;PUT STRING CHECK/MOVE TO (HL)
2A2B            ;CHECK PUT STRING STARTS WITH CHR$ 0, LEN <>0. COPY TO (HL) IN SCRN MEM
2A2B
2A2B 22C85A     PSCHKMHL:  LD (TEMPW1),HL    ;DEST
2A2E CD9512                CALL CHKMD23
2A31                    ;  CALL SPSS         ;GET CURRENT SCREEN AT 8000+
2A31 CDDC3F                CALL GETSTRING    ;DE=STRING ADDR, BC=LEN, PAGE IS SELECTED
2A34 78                    LD A,B            ;!!!SPSS ACTION NEGATED!!
2A35 B1                    OR C
2A36 28F1                  JR Z,PUTBLKERR    ;REQUIRE NON-ZERO LENGTH
2A38
2A38 1A                    LD A,(DE)
2A39 A7                    AND A
2A3A 20ED                  JR NZ,PUTBLKERR   ;REQUIRE CHR$ 0 AS 1ST. CHAR
2A3C
2A3C                    ;  CALL SCRMOV
2A3C                    ;  JP RCURP
2A3C
2A3C            ;SCRNMOV - COPY BC BYTES FROM (DE) TO (TEMPW1) IN SPARE SCRN MEMORY
2A3C            ;USES HL,DE,BC,AF,AF'
2A3C
2A3C 2AC85A     SCRMOV:    LD HL,(TEMPW1)
2A3F 09                    ADD HL,BC
2A40 38E7                  JR C,PUTBLKERR    ;JR IF STRING WILL NOT FIT
2A42
2A42 CD791F                CALL SPLITBC
2A45 3A785A                LD A,(CUSCRNP)
2A48 4F                    LD C,A            ;C:(TEMPW1)=DEST
2A49 DBFB                  IN A,(251)        ;ADE=SRC
2A4B 37                    SCF
2A4C 1815                  JR SFLDIR
2A4E
2A4E CB74       FARLDDR:   BIT 6,H
2A50 2003                  JR NZ,FLD3        ;JR IF SRC ALREADY IN SECT D
2A52
```

```
2A52 3D                    DEC A              ;PAGE IN SRC AT C000-FFFF
2A53 CBF4                  SET 6,H
2A55
2A55 CB72        FLD3:     BIT 6,D
2A57 2003                  JR NZ,FLD4
2A59
2A59 0D                    DEC C
2A5A CBF2                  SET 6,D            ;DITTO DEST
2A5C
2A5C A7          FLD4:     AND A              ;'LDDR'
2A5D 06                    DB 06H             ;'JR +1'
2A5E
2A5E             ;LDIR (PAGCOUNT) PAGES AND (MODCOUNT) BYTES FROM AHL TO CDE. MODCOUNT<=3FFF.
2A5E             ;PAGCOUNT OR MODCOUNT CAN BE ZERO WITHOUT PROBLEMS. PAGING UNCHANGED ON EXIT.
2A5E             ;EXIT WITH TEMPW1=PAST DEST, TEMPB2=PAGE OF PAST DEST, DE=PAST SRC END
2A5E             ;HL OR DE CAN BE ABOVE COOOH ON ENTRY WITHOUT PROBLEMS
2A5E
2A5E 37          FARLDIR:  SCF
2A5F
2A5F EB                    EX DE,HL
2A60 22C85A                LD (TEMPW1),HL     ;DEST ADDR
2A63
2A63             ;C/(TEMPW1) =DEST, ADE=SRC
2A63
2A63 67          SFLDIR:   LD H,A
2A64 79                    LD A,C
2A65 32CF5A                LD (TEMPB2),A
2A68
2A68             ;ENTRY: HDE=SRC, TEMPB2/TEMPW1=DEST. USED IF FARLDIR HAS MOVED SOME DATA ALREADY
2A68
2A68             ;FROM CONCATENATE
2A68
2A68 08          FARLDIR2: EX AF,AF'          ;CY' IF LDIR
2A69 CD493C                CALL R1OSR         ;ROM1 OFF, SAVE PORT STATUSES (STATI?)
2A6C 7C                    LD A,H
2A6D CDDF3F                CALL TSURPG        ;SRC PAGE
2A70 3A835B                LD A,(PAGCOUNT)
2A73 A7                    AND A
2A74 280B                  JR Z,FLDIE
2A76
2A76 F5          FARLDILP: PUSH AF
2A77 010040                LD BC,4000H
2A7A CD8E2A                CALL STRMOV1
2A7D F1                    POP AF
2A7E 3D                    DEC A
2A7F 20F5                  JR NZ,FARLDILP     ;DO PAGCOUNT 16K MOVES
2A81
2A81 ED4B845B    FLDIE:    LD BC,(MODCOUNT)
2A85 CD8B2A                CALL STRMOV
2A88 C3423C                JP POPOUT
2A8B
2A8B             ;STRMOV: MOVE BC BYTES FROM (DE) IN CURRENT UR PAGE TO (TEMPW1) IN
2A8B             ;PAGE (TEMPB2), VIA SYS PAGE. WORKS WITH 0000 TO FFFF BYTES
2A8B             ;CY' IF LDIR, NC' IF LDDR
2A8B             ;EXIT WITH DE=PAST SRC END, PAGE SWITCHED IN, (TEMPW1)=PAST DEST END,
2A8B             ;(TEMPB2)=PAGE OF THE LATTER. HL=0
2A8B             ;USED BY FARLDIR
2A8B
2A8B 78          STRMOV:   LD A,B
2A8C B1                    OR C
2A8D C8                    RET Z
2A8E
2A8E 2ABA5B      STRMOV1:  LD HL,(INSLV)
2A91 24                    INC H
2A92 25                    DEC H
2A93 C20500                JP NZ,HLJUMP
2A96
2A96 60                    LD H,B
2A97 69                    LD L,C             ;HL=BYTES REMAINING TO DO
2A98
2A98 010001      STRMOVL:  LD BC,0100H
2A9B 7C                    LD A,H
2A9C A7                    AND A
2A9D 2002                  JR NZ,STRMOV2      ;JR IF 256 OR MORE TO DO STILL
2A9F
2A9F 44                    LD B,H
2AA0 4D                    LD C,L             ;COUNT=REMAINING. EXIT AFTER THIS MOVE
2AA1
2AA1 E5          STRMOV2:  PUSH HL            ;BYTES REMAINING
2AA2 C5                    PUSH BC
2AA3 EB                    EX DE,HL
2AA4 11004C                LD DE,BUFF256
2AA7 08                    EX AF,AF'
2AA8 380C                  JR C,DLDIR         ;JR IF LDIR
2AAA                                          ;DEST=256 BYTE BUFFER, HL=SRC, BC=256 (USUALLY)
2AAA 08                    EX AF,AF'
2AAB 1D                    DEC E              ;PT TO OTHER END OF BUFFER (E=255)
2AAC EDB8                  LDDR
2AAE C1                    POP BC
2AAF CB74                  BIT 6,H
2AB1 CCF93F                CALL Z,DECURPAGE   ;IF SRC FALLEN INTO SECTION C, DEC PAGE AND SET 6,H
2AB4 1807                  JR STRM32
2AB6
2AB6 08          DLDIR:    EX AF,AF'
2AB7 EDB0                  LDIR               ;COPY BYTES TO BASE PAGE
2AB9 C1                    POP BC
2ABA CDEF3F                CALL CHKHL
2ABD
2ABD E5          STRM32:   PUSH HL            ;SRC PTR (8000-BFFF OR C000-FFFF IF LDDR)
2ABE DBFB                  IN A,(251)
2AC0 F5                    PUSH AF            ;SAVE SRC PAGE
2AC1 3ACF5A                LD A,(TEMPB2)
2AC4 CDDF3F                CALL TSURPG        ;GET DEST AT 8000+
```

```
2AC7 ED5BC85A              LD DE,(TEMPW1)    ;DEST PTR
2ACB C5                    PUSH BC           ;BYTE COUNT
2ACC 08                    EX AF,AF'
2ACD 21004C                LD HL,BUFF256
2AD0 380C                  JR C,STRM34
2AD2
2AD2 08                    EX AF,AF'
2AD3 2D                    DEC L             ;POINT TO OTHER END OF BUFFER
2AD4 EDB8                  LDDR
2AD6 EB                    EX DE,HL
2AD7 CB74                  BIT 6,H
2AD9 CCF93F                CALL Z,DECURPAGE
2ADC
2ADC 1807                  JR STRM38
2ADE
2ADE 08          STRM34:   EX AF,AF'
2ADF EDB0                  LDIR              ;COPY BYTES FROM SYS PAGE TO DEST
2AE1 EB                    EX DE,HL
2AE2 CDEF3F                CALL CHKHL
2AE5
2AE5 DBFB        STRM38:   IN A,(URPORT)
2AE7 32CF5A                LD (TEMPB2),A     ;POSSIBLY NEW DEST PAGE (BITS 7-5 MAY BE HI)
2AEA
2AEA 22C85A      STRMOV4:  LD (TEMPW1),HL    ;NEW DEST ADDR (SECTION C IF LDIR, D IF LDDR)
2AED EB                    EX DE,HL
2AEE C1                    POP BC            ;BYTE COUNT
2AEF F1                    POP AF
2AF0 D3FB                  OUT (251),A       ;RESTORE SRC PAGE
2AF2 D1                    POP DE            ;SRC PTR
2AF3 E1                    POP HL            ;BYTES REMAINING TO MOVE
2AF4 A7                    AND A
2AF5 ED42                  SBC HL,BC
2AF7 209F                  JR NZ,STRMOVL
2AF9
2AF9 C9                    RET
2AFA
2AFA                       INCLUDE ASSIGN.SAM     ;STKVAR, ASSIGN, SYNTAX1, DIM, SLICER
2AFA             ;ASSIGN.SAM
2AFA
2AFA 3A3B5C      VALFET1:  LD A,(FLAGS)
2AFD
2AFD F5          VALFET2:  PUSH AF
2AFE CDE314                CALL SCANNING
2B01 3A3B5C                LD A,(FLAGS)
2B04 57                    LD D,A
2B05 F1                    POP AF
2B06 AA                    XOR D
2B07 E640                  AND 40H
2B09 C2290D                JP NZ,NONSENSE
2B0C
2B0C 7A                    LD A,D
2B0D 17                    RLA
2B0E D0                    RET NC            ;RET IF SYNTAX TIME
2B0F
2B0F             ;RUN TIME ROUTINE
2B0F
2B0F CD372B      ASSIGN:   CALL ASSISR
2B12 C3D73F                JP SELCHADP
2B15
2B15 E5          CGXRG:    PUSH HL
2B16 CD201D                CALL HLTOFPCS
2B19 47                    LD B,A
2B1A
2B1A EF                    DB CALC           ;XRG
2B1B 26                    DB ONELIT
2B1C 02                    DB 2              ;XRG,2
2B1D 23                    DB STKBREG        ;XRG,2,(0/29H)
2B1E 1F                    DB JPFALSE        ;JP IF ZERO, MULT
2B1F 04                    DB 4
2B20
2B20 05                    DB DIVN           ;XRG/2
2B21 20                    DB JUMP           ;JP EXIT
2B22 02                    DB 2
2B23
2B23 00                    DB MULT           ;XRG*2
2B24 33                    DB EXIT
2B25
2B25 D1                    POP DE
2B26 1820                  JR ASENV          ;DELETE VALUE, COPY TO VARS
2B28
2B28
2B28             ;FROM PARPRO
2B28
2B28 3A3F51      CRTVAR35: LD A,(TLBYTE)
2B2B
2B2B             ;CALLED BY MERGE, SETUPVARS
2B2B
2B2B 4F          CRTVAR4:  LD C,A
2B2C 213B5C                LD HL,FLAGS
2B2F CBF6                  SET 6,(HL)        ;'NUMERIC'
2B31
2B31 CDC213                CALL NUMLOOK
2B34 CDFA2C                CALL SYN14C
2B37
2B37 CD1B1F      ASSISR:   CALL ADDRDEST     ;ADDR OF LAST PTR LSB IF NEW NUMERIC VAR
2B3A                                         ;ADDR OF 1ST BYTE OF 5 IF EXISTING NUMBER
2B3A                                         ;ADDR OF SAVARS TERMINATOR IF NEW STRING (NOT USED)
2B3A                                         ;ADDR OF FIRST CHAR IF EXISTING STRING
2B3A 3A3B5C                LD A,(FLAGS)
2B3D 87                    ADD A,A
2B3E 3A715C                LD A,(FLAGX)
2B41 F2CD2B                JP P,ASSTR        ;JR IF STRING
2B44
```

```
2B44 EB                    EX DE,HL
2B45 1F                    RRA
2B46 3809                  JR C,ASNN          ;JR IF NUMBER IS 'NEW'
2B48
2B48             ;ASSIGN TO EXISTING NUMERIC VAR. DE PTS TO EXISTING VALUE IN NVARS.
2B48
2B48 CD121D      ASENV:    CALL FDELETE       ;DELETE FPC DATA, LEAVE HL PTING TO IT
2B4B
2B4B 010500      LDI5:     LD BC,5
2B4E EDB0                  LDIR               ;COPY OVER OLD VALUE
2B50 C9                    RET
2B51
2B51             ;ASSIGN NEW NUMBER. DE PTS TO LSB OF LAST LINK PTR. CALCULATE NEW VALUE TO POINT
2B51             ;TO NUMEND, WHERE NEW VARIABLE WILL BE PLACED
2B51
2B51 3AA55A      ASNN:     LD A,(DESTP)
2B54 E61F                  AND 1FH
2B56 4F                    LD C,A
2B57 2A855A                LD HL,(NUMEND)
2B5A 3A845A                LD A,(NUMENDP)
2B5D 91                    SUB C
2B5E 2808                  JR Z,ANSP          ;JR IF SAME PAGE (USUAL CASE) ELSE A=1,2,3 ETC
2B60
2B60 010040                LD BC,4000H
2B63
2B63 09          ANSPL:    ADD HL,BC
2B64 3D                    DEC A
2B65 20FC                  JR NZ,ANSPL        ;ADJUST ADDR OF NUMEND UP TILL BOTH NUMBERS
2B67                                          ;ARE 'NORMALISED' WITH EACH OTHER. (WRAP-ROUND
2B67 A7                    AND A              ;DOESN'T MATTER TILL >64K)
2B68
2B68 ED52        ANSP:     SBC HL,DE          ;GET DISP FROM LSB OF PTR TO FIRST FREE BYTE IN
2B6A                                          ;NUMS-SAVARS GAP
2B6A 2B                    DEC HL             ;GET DISP FROM *MSB*.
2B6B E5                    PUSH HL            ;SAVE DISP VALUE
2B6C                                          ;(WE CAN'T ALTER THE LINK YET - THERE MAY NOT BE
2B6C                                          ;ENOUGH MEMORY TO CREATE THE VARIABLE)
2B6C 3A845A                LD A,(NUMENDP)
2B6F 4F                    LD C,A
2B70 2A825A                LD HL,(SAVARS)
2B73 3A815A                LD A,(SAVARSP)
2B76 B9                    CP C               ;Z,NC OR NZ,NC (SAVARS *ALWAYS* HIGHER)
2B77 2802                  JR Z,ABSP          ;JR IF (AS UNUSUAL) NUMEND AND SAVARS-START ARE
2B79                                          ;IN SAME PAGE
2B79 CBF4                  SET 6,H            ;ELSE SAVARS ARE JUST 1 PAGE HIGHER - ADD 4000H
2B7B                                          ;TO HL SO BOTH NUMBERS 'NORMALISED'
2B7B ED4B855A    ABSP:     LD BC,(NUMEND)
2B7F ED42                  SBC HL,BC
2B81 EB                    EX DE,HL           ;DE=FREE GAP BETWEEN NUMS AND STR/ARRAYS
2B82                                          ;HL=LSB OF PTR ADDR
2B82 7A                    LD A,D
2B83 A7                    AND A
2B84 2014                  JR NZ,ANOK         ;JR IF AT LEAST 256 BYTES FREE
2B86
2B86 7B                    LD A,E
2B87 FE3C                  CP 60
2B89 300F                  JR NC,ANOK         ;JR IF AT LEAST 60 BYTES FREE (ENOUGH FOR
2B8B                                          ;LARGEST NUMERIC VARIABLE)
2B8B
2B8B CD271F                CALL ADDRSAV
2B8E CDB91F                CALL DECPTR
2B91 010002                LD BC,0200H
2B94 CD1B1E                CALL MAKEROOM      ;OPEN 512 BYTES BEFORE SAVARS
2B97 CD1B1F                CALL ADDRDEST      ;HL=PTR LSB
2B9A
2B9A             ;OK TO CREATE NEW NUMERIC
2B9A
2B9A D1          ANOK:     POP DE             ;DISP
2B9B 73                    LD (HL),E
2B9C 23                    INC HL
2B9D 72                    LD (HL),D          ;MAKE LINK OF PREVIOUS LAST-VAR-OF-THIS-LETTER
2B9E                                          ;PT TO NEW LAST VAR.
2B9E CD231F                CALL ADDRNE        ;PT HL TO NUMEND (LOCN OF NEW VAR.
2BA1 3A6051                LD A,(TLBYTE+33)
2BA4 77                    LD (HL),A
2BA5 23                    INC HL
2BA6 06FF                  LD B,0FFH
2BA8 70                    LD (HL),B
2BA9 23                    INC HL
2BAA 70                    LD (HL),B
2BAB 23                    INC HL             ;PTR=FFFF (LAST VAR OF THIS FIRST LETTER)
2BAC EB                    EX DE,HL
2BAD 216251                LD HL,FIRLET+34    ;PT TO SECOND LETTER OF NAME
2BB0 E61F                  AND 1FH
2BB2 2804                  JR Z,ASNCL         ;JR IF SINGLE-LET VAR
2BB4
2BB4 4F                    LD C,A
2BB5 04                    INC B              ;BC=LEN OF NAME (LESS FIRST LETTER)
2BB6 EDB0                  LDIR               ;COPY TO VARS
2BB8
2BB8 CD482B      ASNCL:    CALL ASENV         ;COPY FPC VALUE TO VARS
2BBB
2BBB ED53855A    NELOAD:   LD (NUMEND),DE     ;NUMEND IS PAST LAST BYTE OF VALUE
2BBF CB72                  BIT 6,D
2BC1 C8                    RET Z              ;RET IF STILL 8000-BFFF
2BC2
2BC2 CBB2                  RES 6,D
2BC4 3A845A                LD A,(NUMENDP)
2BC7 3C                    INC A
2BC8 32845A                LD (NUMENDP),A
2BCB 18EE                  JR NELOAD
2BCD
2BCD             ;ASSIGN A STRING
```

```
2BCD
2BCD 1F          ASSTR:     RRA                ;TEST BIT 0,(FLAGX)
2BCE DA7B2C                 JP C,ASNST         ;JP IF IT IS A NEW STRING
2BD1
2BD1             ;ASSIGN TO EXISTING STRING VAR
2BD1
2BD1 ED4B725C               LD BC,(STRLEN)     ;LENGTH OF DESTINATION
2BD5 3AA55A                 LD A,(DESTP)
2BD8 17                     RLA
2BD9 386E                   JR C,ASDEL         ;JR IF UNSLICED SIMPLE STRING - DELETE OLD VERSION
2BDB
2BDB 78                     LD A,B
2BDC B1                     OR C
2BDD C8                     RET Z              ;RET IF E.G. LET A$(4 TO 3)="TEST" - DEST LEN=0
2BDE
2BDE E5                     PUSH HL            ;DEST ADDR
2BDF C5                     PUSH BC            ;DEST SIZE
2BE0 CD011D                 CALL STKFETCH      ;ADE/BC =STRING START/LEN
2BE3 E1                     POP HL             ;DEST SIZE
2BE4 ED42                   SBC HL,BC          ;DEST SIZE-SRC SIZE (NC HERE)
2BE6 3006                   JR NC,AES1         ;JR IF TRUNC NOT NEEDED
2BE8
2BE8 09                     ADD HL,BC          ;HL=DEST SIZE AGAIN
2BE9 44                     LD B,H
2BEA 4D                     LD C,L             ;BC=TRUNCATED SRC LEN NEEDED TO FILL DEST
2BEB 210000                 LD HL,0            ;'PADS' NEEDED=0
2BEE
2BEE E3          AES1:      EX (SP),HL         ;PADS NEEDED TO STACK, DEST ADDR TO HL
2BEF EB                     EX DE,HL           ;DE=DEST, HL=SRC
2BF0 08                     EX AF,AF'
2BF1 CD791F                 CALL SPLITBC       ;LD PAGCOUNT/MODCOUNT WITH BC
2BF4 DBFB                   IN A,(251)
2BF6 4F                     LD C,A             ;CDE=DEST
2BF7 08                     EX AF,AF'          ;AHL=SRC
2BF8 CD5E2A                 CALL FARLDIR       ;COPY STRING TO DEST
2BFB C1                     POP BC             ;PADS
2BFC 78                     LD A,B
2BFD B1                     OR C
2BFE C8                     RET Z
2BFF
2BFF 3ACF5A                 LD A,(TEMPB2)
2C02 CDDF3F                 CALL TSURPG
2C05 2AC85A                 LD HL,(TEMPW1)     ;PT TO PAST LAST BYTE FARLDIRED EARLIER
2C08 AF                     XOR A
2C09 B9                     CP C               ;NC ONLY IF C=0
2C0A 88                     ADC A,B            ;A=B+1 UNLESS C WAS ZERO, WHEN A=B
2C0B 41                     LD B,C
2C0C 4F                     LD C,A
2C0D 3E20                   LD A,20H
2C0F
2C0F 77          ASPSL:     LD (HL),A
2C10 23                     INC HL
2C11 10FC                   DJNZ ASPSL
2C13
2C13 0D                     DEC C
2C14 C8                     RET Z
2C15
2C15 CDEF3F                 CALL CHKHL
2C18 18F5                   JR ASPSL
2C1A
2C1A             ;E.G. RECORD TO A$ OR:  RECORD STOP
2C1A
2C1A FEB1        RECORD:    CP 0B1H            ;IS IT 'RECORD STOP'? STOPTOK
2C1C 2006                   JR NZ,RECORD2
2C1E
2C1E AF                     XOR A
2C1F 32815B                 LD (GRARF),A       ;GRAPHICS RECORD FLAG 0 (OFF)
2C22 E7                     RST 20H            ;SKIP 'RECORD'
2C23 C9                     RET
2C24
2C24 FE8E        RECORD2:   CP TOTOK
2C26 200C                   JR NZ,RCNONS
2C28
2C28 E7                     RST 20H
2C29 CDDA14                 CALL LVFLAGS
2C2C FA290D                 JP M,NONSENSE      ;ERROR IF NUMERIC
2C2F
2C2F 3805                   JR C,RECORD3       ;JR IN RUN TIME (ALLOWS STR ARRAYS TO BE DELETED)
2C31
2C31 CB71                   BIT 6,C
2C33 C8                     RET Z              ;RET IF SIMPLE STRING, ELSE ERROR IF E.G. A$(3)
2C34
2C34 CF          RCNONS:    RST 08H
2C35 1D                     DB 29
2C36
2C36 08          RECORD3:   EX AF,AF'
2C37 C45B2C                 CALL NZ,ASDEL2     ;IF FOUND, DELETE VAR PTED TO BY STRLOC
2C3A
2C3A 11765B                 LD DE,STRM16NM
2C3D CD432D                 CALL SCOPN1        ;COPY NAME TO STRM16NM.( MAY COPY 12 BYTES,
2C40                                           ;HITTING GRARF - BUT IRREL)
2C40 7A                     LD A,D
2C41 32815B                 LD (GRARF),A       ;GRAPHICS RECORD FLAG=NZ (ON)
2C44 CD402D                 CALL SCOPNM        ;EXIT WITH BC=0 (NULL LEN)
2C47 1852                   JR ASNS1           ;ASSIGN NULL STRING TO NAME
2C49
2C49             ;ASSIGN STRING, THEN DELETE OLD STRING. E.G. LET A$=A$ OR LET A$=A$+"X"
2C49
2C49 CD7B2C      ASDEL:     CALL ASNST         ;CREATE NEW VERSION FIRST SO E.G. LET A$=A$ WORKS
2C4C CD1B1F                 CALL ADDRDEST      ;HL PTS TO TEXT OF OLD STRING
2C4F 11F2FF                 LD DE,-14
2C52 19                     ADD HL,DE          ;PT TO TLBYTE IN VARS
2C53 CDBA1F                 CALL CHKPTR
```

```
2C56 1806                  JR ASDEL3
2C58
2C58               ;CALLED BY DIM TO DELETE EXISTING ARRAYS OR STRINGS, WITH A=PAGE
2C58
2C58 CDDF3F        ASDL1:  CALL SELURPG
2C5B
2C5B               ;CALLED BY END PROC TO DELETE LOCAL STRINGS/ARRAYS, RECORD TO DELETE EXISTING
2C5B               ;STRING/ARRAY. ENTRY WITH STRLOCN PAGED IN
2C5B
2C5B 2ABC5B        ASDEL2: LD HL,(STRLOCN)
2C5E
2C5E E5            ASDEL3: PUSH HL            ;PTR TO TLBYTE
2C5F 010B00                LD BC,11
2C62 09                    ADD HL,BC          ;PT TO LEN (PAGES)
2C63 CD6C2C                CALL ADD14
2C66 44                    LD B,H
2C67 4D                    LD C,L
2C68 E1                    POP HL             ;PTR TO TLBYTE
2C69 C3531E                JP RECL2BIG        ;DELETE STR/ARRAY AND 14-BYTE HEADER (ABC AT HL)
2C6C
2C6C               ;CALLED BY TAPEMN
2C6C
2C6C 7E            ADD14:  LD A,(HL)
2C6D 23                    INC HL
2C6E 5E                    LD E,(HL)          ;LEN MOD 16K
2C6F 23                    INC HL
2C70 56                    LD D,(HL)
2C71 EB                    EX DE,HL           ;AHL=LEN (PAGES, LEN MOD 16K)
2C72 010E00                LD BC,14
2C75 09                    ADD HL,BC          ;ADD 14 TO GET LEN INCLUDING HDR
2C76 CB74                  BIT 6,H
2C78 C8                    RET Z              ;RET IF MOD <16K
2C79
2C79 3C                    INC A
2C7A C9                    RET
2C7B
2C7B               ;ASSIGN A NEW STRING
2C7B
2C7B CD011D        ASNST:  CALL STKFETCH      ;DATA FOR STRING ASSIGNMENT
2C7E
2C7E E61F                  AND 1FH
2C80 67                    LD H,A
2C81 3A8D5A                LD A,(WKENDP)
2C84 6F                    LD L,A
2C85 7C                    LD A,H
2C86 BD                    CP L
2C87 3812                  JR C,ASNS1         ;JR IF SRC PAGE LOWER THAN WKEND
2C89
2C89 2007                  JR NZ,ASNS0        ;JR IF SRC PAGE HIGHER
2C8B
2C8B 2A8E5A                LD HL,(WKEND)
2C8E ED52                  SBC HL,DE
2C90 3009                  JR NC,ASNS1        ;JR IF SRC <=WKEND
2C92
2C92 ED537D5A      ASNS0:  LD (FIRST),DE      ;ELSE SRC>WKEND AND SHOULD NOT BE AUTO-ADJUSTED
2C96 327F5A                LD (LAST),A
2C99 3EFF                  LD A,0FFH          ;SIGNAL 'XPTR NOT USED'
2C9B
2C9B               ;CALLED BY 'RECORD' TO CREATE NULL STRING
2C9B
2C9B F5            ASNS1:  PUSH AF
2C9C C5                    PUSH BC            ;STRING LEN
2C9D ED53A35A              LD (XPTR),DE
2CA1 32A25A                LD (XPTRP),A       ;SAVE START IN AUTO-ADJ VAR (MAKEROOM MAY MOVE IT)
2CA4 3E0E                  LD A,14            ;ALLOW FOR TYPE/NAME LEN (1) NAME (10), TXT LEN (3)
2CA6 81                    ADD A,C            ;ADD TO TEXT LEN TO GET ROOM NEEDED
2CA7 4F                    LD C,A
2CA8 3004                  JR NC,ASNS2
2CAA
2CAA 04                    INC B
2CAB CA283A                JP Z,STLERR        ;TOTAL LEN MUST NOT EXCEED FFFFH
2CAE
2CAE 78            ASNS2:  LD A,B
2CAF 07                    RLCA
2CB0 07                    RLCA
2CB1 E603                  AND 03H
2CB3 CD8E2F                CALL SAROOM
2CB6 C1                    POP BC             ;STRING LEN
2CB7 CD433A                CALL MBC           ;COPY PAG/MOD COUNT TO VARS
2CBA DBFB                  IN A,(251)
2CBC 4F                    LD C,A             ;CDE=DEST
2CBD F1                    POP AF
2CBE 3C                    INC A
2CBF 3AA25A                LD A,(XPTRP)
2CC2 2AA35A                LD HL,(XPTR)
2CC5 2006                  JR NZ,ASNS3        ;JR IF XPTR NOT USED
2CC7
2CC7 3A7F5A                LD A,(LAST)
2CCA 2A7D5A                LD HL,(FIRST)
2CCD
2CCD 32A45A        ASNS3:  LD (XPTR+1),A      ;PAGE SHOULD HAVE BIT 7 LOW - CANCEL XPTR
2CD0 C35E2A                JP FARLDIR         ;LDIR PAGCOUNT/MODCOUNT BYTES
2CD3
2CD3
2CD3               ;ASSESS FOR-NEXT VAR (USED BY 'FOR' AND 'NEXT')
2CD3
2CD3 CDDA14        SYNTAX4: CALL LVFLAGS
2CD6 F2290D                JP P,NONSENSE      ;ERROR IF STRING
2CD9
2CD9 CB69                  BIT 5,C
2CDB C2290D                JP NZ,NONSENSE     ;OR NUMERIC ARRAY NAME
2CDE
2CDE 3013                  JR NC,SYNT41       ;JR IF SYNTAX TIME
```

```
2CE0
2CE0 08                    EX AF,AF'
2CE1
2CE1 2817       SYN42:     JR Z,SYN14C       ;JR IF DOESN'T EXIST
2CE3
2CE3 CB71                  BIT 6,C           ;C IS FROM VARS
2CE5 2013                  JR NZ,SYN14C      ;JR IF (EXISTING) FOR-NEXT VAR
2CE7
2CE7 DDE5                  PUSH IX
2CE9 E1                    POP HL            ;ADDR OF PTR LSB
2CEA DDCBFFEE              SET 5,(IX-1)      ;TYPE BYTE MARKED AS 'UNUSED' ** BUG FIX
2CEE             ;         LD C,0FFH         ;'INVIS' AND 'UNUSED' - DON'T EXIST
2CEE CDD113                CALL NVMLP
2CF1 18EE                  JR SYN42          ;LOOP FOR ALL COPIES
2CF3
2CF3            ;          LD C,0            ;NON-ARRAY TYPE
2CF3            ;          DB 3EH            ;'JR+1'
2CF3
2CF3 08         SYNT41:    EX AF,AF'
2CF4
2CF4 1804                  JR SYN14C
2CF6
2CF6 E7         SSYNTAX1:  RST 20H
2CF7
2CF7            ;USED BY LET/READ/INPUT TO ASSESS VAR ABOUT TO BE ASSIGNED TO.
2CF7
2CF7 CDAA13     SYNTAX1:   CALL LOOKVARS     ;IF FND, C=T/L FROM VARS, ELSE C=DESIRED T/L
2CFA
2CFA EB         SYN14C:    EX DE,HL          ;IF FND, DE PTS TO START OF NUMBER, OR LEN INFO OF
2CFB                                         ;STRINGS/ARRAYS (PAGE/LEN MOD 16K)
2CFB            ;FROM PARAM PROCESSING
2CFB
2CFB 21715C     SYN1PP:    LD HL,FLAGX
2CFE 3600                  LD (HL),0         ;VAR NOT NEW (BIT 0=0)
2D00 2008                  JR NZ,TSYNT12     ;JR IF VAR EXISTS, OR SYNTAX TIME
2D02
2D02                                         ;ELSE C='DESIRED' TYPE, BITS 6 AND 5 ARE 0
2D02                                         ;IF SIMPLE UNSLICED STRING, OR A SIMPLE NUMBER
2D02
2D02 34                    INC (HL)          ;'NEW VARIABLE'
2D03
2D03 79                    LD A,C
2D04 E660                  AND 60H
2D06 281D                  JR Z,TSYNT14      ;DESTP BIT 7 WILL BE 0 ('KEEP OLD VALUE')
2D08
2D08                                         ;ERROR IF TRYING TO USE AN UNDIMED ARRAY, OR SLICE
2D08                                         ;A NEW STRING.
2D08 CF         VNFERR:    RST 08H
2D09 02                    DB 2
2D0A
2D0A            ;VAR EXISTS, OR SYNTAX TIME
2D0A
2D0A 3A3B5C     TSYNT12:   LD A,(FLAGS)
2D0D 87                    ADD A,A           ;P IF STRING, CY IF RUNNING
2D0E F2152D                JP P,TSYNT13      ;JP IF STRING
2D11
2D11 CB69                  BIT 5,C
2D13 2810                  JR Z,TSYNT14      ;JR IF A SIMPLE NUMBER, NOT AN ARRAY
2D15                                         ;ELSE STKVAR HANDLES NUMERIC ARRAYS
2D15
2D15 CD532D     TSYNT13:   CALL STKVAR       ;PASS ARRAY/STRING DATA TO FPCS. EXIT WITH HL
2D18                                         ;PTING TO VALUE, IF NUMERIC ARRAY, ELSE FPCS
2D18                                         ;HOLDS STRING DETAILS
2D18 3A3B5C                LD A,(FLAGS)
2D1B 87                    ADD A,A           ;CY IF RUNNING, -VE IF NUMERIC
2D1C FA262D                JP M,TSYNT15      ;JP IF NUMERIC - HL PTS TO VALUE, PAGED IN
2D1F
2D1F DC011D                CALL C,STKFETCH   ;GET DE=START, BC=LEN, A=PAGE (IN VARS). BIT 7=1
2D22 EB                    EX DE,HL          ;IF 'OLD COPY TO BE DELETED' AS IN 'LET A$="SS"'
2D23 1805                  JR TSYN16
2D25
2D25 EB         TSYNT14:   EX DE,HL
2D26
2D26            ;FOR NUMERICS AND NEW STRING/ARRAY VARS, STRLEN=TYPE/LEN FROM VARS AND JUNK;
2D26            ;FOR EXISTING STRING/ARRAY VARS, STRLEN=LENGTH
2D26            ;   IF 'OLD COPY TO BE DELETED' BIT 7 OF DESTP=1
2D26            ;   TLBYTE=REQUESTED TYPE/LEN, FIRLET=NAME
2D26
2D26 DBFB       TSYNT15:   IN A,(251)
2D28 E61F                  AND 1FH
2D2A
2D2A ED43725C   TSYN16:    LD (STRLEN),BC
2D2E 22A65A                LD (DEST),HL      ;ADDRESS OF NUMERIC OR STRING VALUE, OR PTR/STOPPER
2D31 32A55A                LD (DESTP),A      ;PAGE OF STRING START, OR CURRENT PAGE IF NUMBER
2D34 46                    LD B,(HL)
2D35 23                    INC HL
2D36 7E                    LD A,(HL)
2D37 3C                    INC A
2D38 B0                    OR B
2D39 23                    INC HL
2D3A B6                    OR (HL)
2D3B 23                    INC HL
2D3C B6                    OR (HL)           ;IF VAR STARTS 00 FF 00 00, A=0
2D3D 32B15A                LD (DFTFB),A      ;IRREL IF NVAR NON-EXISTENT
2D40
2D40            ;CALLED BY 'DIM'
2D40
2D40 116051     SCOPNM:    LD DE,TLBYTE+33
2D43
2D43            ;CALLED BY 'RECORD TO'
2D43
2D43 213F51     SCOPN1:    LD HL,TLBYTE
2D46 7E                    LD A,(HL)
```

```
2D47 E61F                AND 1FH            ;NAME LEN-1 IF NUMERIC, TRUE NAME LEN IF STR/ARRAY
2D49 C602                ADD A,2            ;ALLOW FOR TLBYTE AND (PERHAPS) ANOTHER LETTER
2D4B 4F                  LD C,A
2D4C
2D4C                ;CALLED BY LENGTH SR
2D4C
2D4C 0600           SCOPN2:    LD B,0
2D4E EDB0                LDIR               ;COPY NAME TO BUFFER THAT WON'T BE USED BY EVAL
2D50 C3D73F              JP SELCHADP
2D53
2D53
2D53                ;FIND START AND LEN OF AN EXISTING STRING, OR START OF A NUMBER IN AN ARRAY
2D53                ;ON ENTRY: DE PTS TO PAGES OF VAR LEN, THEN LENGTH MOD 16K. C=T/L. CY IF RUNNING
2D53                ;CHAD POINTS PAST '$' OR '(' (UNLESS ERROR)
2D53
2D53 EB             STKVAR:    EX DE,HL           ;HL PTS TO PAGES OF LEN IF RUNNING
2D54
2D54 383B           STKVAR2:   JR C,SVRUNT        ;JR IF RUNNING
2D56
2D56 CB71                BIT 6,C
2D58 200F                JR NZ,SVSSL        ;JR IF STRING ARRAY OR SLICED STRING
2D5A
2D5A CB69                BIT 5,C
2D5C C8                  RET Z              ;RET IF SIMPLE UNSLICED STRING - NO ACTION
2D5D                                        ;CONTINUE WITH NUMERIC ARRAYS
2D5D FE                  DB 0FEH            ;'JR+1'
2D5E
2D5E E7             SVDSL:     RST 20H
2D5F
2D5F                ;CALLED BY 'DIM' SYNTAX CHECK - CHECK  N,N,...N)
2D5F
2D5F CDE43A         SVDSK:     CALL EXPT1NUM
2D62 FE2C                CP ","
2D64 28F8                JR Z,SVDSL
2D66
2D66 C3953A         SVIBH:     JP INSISCBRK       ;')'
2D69
2D69
2D69                ;CHECK STRING ARRAY SYNTAX; E.G. ), N,X TO Y) OR N,N,TO Y) OR N,N,Y TO)
2D69
2D69 DF             SVSSL:     RST 18H
2D6A FE29                CP ")"
2D6C 281F                JR Z,SVSL3         ;ALLOW '()'
2D6E
2D6E FE                  DB 0FEH            ;'JR+1'
2D6F
2D6F E7             SVSSLP:    RST 20H
2D70
2D70 FE8E                CP TOTOK
2D72 2810                JR Z,SVSL2
2D74
2D74 CDE43A              CALL EXPT1NUM
2D77 FE2C                CP ","
2D79 28F4                JR Z,SVSSLP
2D7B
2D7B FE8E                CP TOTOK
2D7D 2805                JR Z,SVSL2
2D7F
2D7F CD953A              CALL INSISCBRK
2D82 180A                JR SLPXHP
2D84
2D84 E7             SVSL2:     RST 20H            ;SKIP 'TO'
2D85 FE29                CP ")"
2D87 2804                JR Z,SVSL3         ;SKIP ')' IF IT IS ONE
2D89
2D89 CD923A              CALL EX1NUMCB      ;ELSE ACCEPT 'N)'
2D8C FE                  DB 0FEH            ;'JR+1'
2D8D
2D8D E7             SVSL3:     RST 20H            ;SKIP ')'
2D8E
2D8E C33E2E         SLPXHP:    JP SLLPEX          ;SET 'STRING' STATUS (DISTURBED BY 'EXPT1NUM')
2D91
2D91                ;*********************************************************************
2D91                ;STACK VAR - RUN TIME
2D91
2D91 79             SVRUNT:    LD A,C
2D92 E660                AND 60H
2D94 2027                JR NZ,SVARRAYS     ;JR IF ARRAY, ELSE HANDLE SIMPLE STRING BY
2D96                                        ;CONVERTING THE PAGE/LEN MOD 16K DATA IN VARS TO 2
2D96                                        ;BYTES (SIMPLE STRINGS HAVE LEN 0000-FFFF)
2D96 7E                  LD A,(HL)          ;PAGES (0-3)
2D97 23                  INC HL
2D98 4E                  LD C,(HL)
2D99 23                  INC HL
2D9A 0F                  RRCA
2D9B 0F                  RRCA               ;??00 0000
2D9C B6                  OR (HL)
2D9D 47                  LD B,A             ;BC=LEN
2D9E 1680                LD D,80H
2DA0
2DA0 EB             SVSIMPLE:  EX DE,HL
2DA1 13                  INC DE             ;PT TO TEXT
2DA2 DBFB                IN A,(251)
2DA4 CB72                BIT 6,D
2DA6 2803                JR Z,SVSS2         ;JR UNLESS STR. STARTED AT E.G. BFFF, PTR NOW C001
2DA8
2DA8 CBB2                RES 6,D
2DAA 3C                  INC A
2DAB
2DAB E61F           SVSS2:     AND 1FH
2DAD B4                  OR H               ;BIT 7 SET (DELETE OLD COPY) IF SIMPLE STRING
2DAE                                        ;BIT 7 RES (OVERWRITE) IF 1-DIM STRING ARRAY
2DAE CDF01C              CALL STKSTORE      ;DE=ST, BC=LEN, A=START PAGE
```

```
2DB1                                            ;BIT 6,(FLAGS) IS CORRECT ALREADY
2DB1 CDD73F             CALL SELCHADP
2DB4 3A3F51             LD A,(TLBYTE)
2DB7 CB77               BIT 6,A
2DB9 C8                 RET Z                   ;RET IF NO BRACKET AFTER NAME - NO SLICING
2DBA
2DBA C3392E             JP SLCL2
2DBD
2DBD 23         SVARRAYS:  INC HL
2DBE 23                 INC HL
2DBF 23                 INC HL
2DC0 46                 LD B,(HL)               ;NO. OF DIMS
2DC1 CB69               BIT 5,C
2DC3 2010               JR NZ,SVCDIS            ;JR IF NUMERIC ARRAY
2DC5
2DC5 1007               DJNZ SVCKS              ;JR IF MULTI-DIM STRING ARRAY (B=DIMS-1)
2DC7
2DC7 50                 LD D,B                  ;D=0 SO BIT 7,A LEFT AS ZERO (SIGNALS 'OVERWRITE')
2DC8 23                 INC HL
2DC9 4E                 LD C,(HL)
2DCA 23                 INC HL
2DCB 46                 LD B,(HL)               ;BC=LEN OF SINGLE DIMENSION ($)
2DCC 18D2               JR SVSIMPLE             ;HANDLE LIKE A SIMPLE STRING
2DCE
2DCE 3A3F51     SVCKS:     LD A,(TLBYTE)        ;CHECK THAT SLICING WAS USED TO REFER TO MULTI-DIM
2DD1 E640               AND 40H                 ;STRING ARRAY
2DD3 2821               JR Z,SWERHP             ;ERROR IF NOT.
2DD5
2DD5 DBFB       SVCDIS:    IN A,(URPORT)
2DD7 F5                 PUSH AF                 ;PAGE OF ARRAY DIM DATA
2DD8 C5                 PUSH BC                 ;B=DIM COUNT (EXCLUDING FINAL DIM IF STRING)
2DD9 23                 INC HL                  ;PT TO FIRST DIM SIZE
2DDA E5                 PUSH HL
2DDB AF                 XOR A
2DDC CDDA1C             CALL STACKA             ;ZERO TOTAL OF FPCS
2DDF E1                 POP HL                  ;PTR TO DIMN. DATA IN BUFFER.
2DE0 C1                 POP BC                  ;B=DIMS (1 OR MORE, EXCLUDING LAST DIM IF STRING)
2DE1
2DE1 F1         SVLOOP:    POP AF
2DE2 F5                 PUSH AF
2DE3 D3FB               OUT (URPORT),A          ;PAGE IN DIMS
2DE5 C5                 PUSH BC                 ;DIM COUNTER IN B
2DE6 4E                 LD C,(HL)
2DE7 23                 INC HL
2DE8 46                 LD B,(HL)               ;BC=NEXT DIM SIZE=LIMIT VALUE FOR SUBSCRIPT
2DE9 23                 INC HL
2DEA E5                 PUSH HL                 ;PTR TO DIMN. DATA
2DEB C5                 PUSH BC                 ;DIM SIZE
2DEC CDD73F             CALL SELCHADP           ;PAGE IN SUBSCRIPT
2DEF CDDD1C             CALL STACKBC            ;DIM SIZE
2DF2 C1                 POP BC                  ;BC=DIM SIZE AGAIN
2DF3 CDA02F             CALL GETSUBS            ;GET SUBSCRIPT IN HL, CHECKING IT'S >0 AND <=LIMIT,
2DF6                                            ;THEN DECING IT.
2DF6 D2312E     SWERHP:    JP NC,SWER2          ;ERROR IF OUTSIDE LIMITS
2DF9
2DF9 CDD61C             CALL STACKHL
2DFC
2DFC EF                 DB CALC                 ;TOTAL,DIM SIZE,SUBS VAL
2DFD 1C                 DB SWOP13               ;SUBS VAL,DIM SIZE,TOTAL
2DFE 00                 DB MULT
2DFF 01                 DB ADDN                 ;TOTAL*DIM SIZE+SUBS VALUE
2E00 33                 DB EXIT
2E01
2E01 D1                 POP DE                  ;ARRAY DATA PTR
2E02 C1                 POP BC                  ;DIM COUNTER
2E03 DF                 RST 18H
2E04 05                 DEC B
2E05 2808               JR Z,SVEXLP             ;JR IF ALL DIMS DONE
2E07
2E07 FE2C               CP ","
2E09 2026               JR NZ,SWER2             ;INSIST ON A COMMA NOW
2E0B
2E0B E7                 RST 20H                 ;SKIP ','
2E0C EB                 EX DE,HL                ;HL PTS TO DIMN. DATA
2E0D 18D2               JR SVLOOP
2E0F
2E0F            ;PAGE OF ARRAY START IS ON STACK, DE=ADDR OF ARRAY START
2E0F
2E0F CB69       SVEXLP:    BIT 5,C
2E11 2031               JR NZ,SVNUMER           ;JR IF NUMERIC ARRAY
2E13
2E13 F1                 POP AF
2E14 D3FB               OUT (URPORT),A          ;PAGE IN DIMS
2E16 EB                 EX DE,HL                ;ALLOW LAST SUBSCRIPT OR SLICER, FOR STRINGS
2E17 4E                 LD C,(HL)
2E18 23                 INC HL
2E19 46                 LD B,(HL)
2E1A 23                 INC HL
2E1B EB                 EX DE,HL                ;DE PTS TO ARRAY START
2E1C C5                 PUSH BC                 ;LAST SUBSCRIPT LEN
2E1D CD532E             CALL SVSR               ;GET START ADDR OF DESIRED STRING IN AHL
2E20 C1                 POP BC
2E21 EB                 EX DE,HL
2E22 CDE91C             CALL STKST0             ;STORE STRING ADDR, BIT 7,A=0 ('DON'T ERASE OLD')
2E25 CDD73F             CALL SELCHADP
2E28 DF                 RST 18H
2E29 FE29               CP ")"
2E2B 2806               JR Z,SVDIM              ;JR IF NO SLICE OF STRING SO FAR. E.G. A$(3)
2E2D
2E2D FE2C               CP ","
2E2F 2807               JR Z,SLCL               ;OK TO HAVE E.G. A$(3,2 TO 5)
2E31
2E31 CF         SWER2:     RST 08H              ;ANYTHING ELSE IS AN ERROR
```

```
2E32 04                    DB 4              ;'Subscript wrong'
2E33
2E33 E7          SVDIM:    RST 20H
2E34 FE28                  CP "("
2E36 2006                  JR NZ,SLLPEX
2E38
2E38 E7          SLCL:     RST 20H           ;SKIP '(' OR ','
2E39
2E39 CD692E      SLCL2:    CALL SLICING
2E3C 18F5                  JR SVDIM
2E3E
2E3E 213B5C      SLLPEX:   LD HL,FLAGS
2E41 CBB6                  RES 6,(HL)        ;'STRING'
2E43 C9                    RET
2E44
2E44
2E44             ;END OF NUMERIC ARRAY
2E44
2E44 FE29        SVNUMER:  CP ")"
2E46 20E9                  JR NZ,SWER2
2E48
2E48 E7                    RST 20H           ;SKIP ')'
2E49 010500                LD BC,5
2E4C F1                    POP AF            ;PAGE
2E4D CD532E                CALL SVSR
2E50 C3DF3F                JP TSURPG
2E53
2E53             ;STACK BC (LAST DIMN. LEN, OR 5 FOR NUMBERS), MULT, ADD ARRAY 'TEXT' START
2E53             ;EXIT WITH AHL=ADDR OF ELEMENT
2E53
2E53 F5          SVSR:     PUSH AF           ;PAGE OF ARRAY START
2E54 D5                    PUSH DE           ;ARRAY START
2E55 CDDD1C                CALL STACKBC
2E58
2E58 EF                    DB CALC           ;TOTAL, LAST DIM SIZE
2E59 00                    DB MULT           ;DISP TO ELEMENT WANTED
2E5A 33                    DB EXIT
2E5B
2E5B CD8C3F                CALL UNSTLEN      ;AHL=PAGES/ MOD 16K FORM
2E5E D1                    POP DE
2E5F C1                    POP BC
2E60 48                    LD C,B            ;CDE=ADDR OF TEXT START (8000-C???)
2E61 CB72                  BIT 6,D
2E63 2801                  JR Z,SVSR2        ;JR IF PAGE OK
2E65
2E65 0C                    INC C             ;ELSE INC PAGE (ADDAHLCDE IGNORES BIT 6)
2E66
2E66 C3DE1F      SVSR2:    JP ADDAHLCDE
2E69
2E69
2E69 CDC63A      SLICING:  CALL RUNFLG
2E6C DC011D                CALL C,STKFETCH   ;GET ADE=START, BC=LEN, IF RUNNING
2E6F
2E6F F5                    PUSH AF           ;PAGE
2E70 DF                    RST 18H
2E71 E1                    POP HL            ;H=PAGE
2E72 FE29                  CP ")"
2E74 2857                  JR Z,SLSTORE      ;JR IF SLICE WAS () (ENTIRE STRING)
2E76
2E76 32CF5A                LD (TEMPB2),A     ;NZ SHOWS NO ERROR IN SUBSCRIPT YET
2E79 D5                    PUSH DE           ;STRING START
2E7A E5                    PUSH HL           ;H=PAGE OF START
2E7B 110000                LD DE,0           ;DEFAULT SLICER START
2E7E FE8E                  CP TOTOK
2E80 2811                  JR Z,SLSEC        ;JR IF E.G: ( TO X) - USE DE=1
2E82
2E82 CDA02F                CALL GETSUBS      ;ELSE EVAL E.G. S OF (S TO T) USING BC AS LIMIT
2E85 EB                    EX DE,HL          ;DE=SUBS. VAL, CHECKED >0 AND <=LEN, THEN DECED
2E86 DF                    RST 18H
2E87 FE8E                  CP TOTOK
2E89 2808                  JR Z,SLSEC        ;WE HAVE FIRST NUMBER IN DE - JR IF 'TO' FOLLOWS IT
2E8B
2E8B FE29                  CP ")"
2E8D
2E8D 20A2        NONSH:    JR NZ,SWER2     ;WAS NONS
2E8F
2E8F 62                    LD H,D
2E90 6B                    LD L,E            ;LAST NUMB=FIRST NUMB IF EG (5)
2E91 181A                  JR SLDEF          ;JR WITH NUMBERS IN HL AND DE
2E93
2E93 E7          SLSEC:    RST 20H
2E94 FE29                  CP ")"
2E96 60                    LD H,B
2E97 69                    LD L,C
2E98 2B                    DEC HL            ;HL=LEN-1 (VALUES ALL USE 'DECED' FORM)
2E99 2812                  JR Z,SLDEF        ;JR IF EG (X TO ) OR ( TO ) - USE LEN AS 2ND NUMB.
2E9B
2E9B D5                    PUSH DE           ;FIRST NUM
2E9C CDA02F                CALL GETSUBS      ;EVAL SECOND NUMBER, CHECKING >0, <=LEN, DECING
2E9F D1                    POP DE            ;FIRST NUMBER
2EA0 3804                  JR C,SLSE2        ;JR IF IN RANGE OR SYNTAX TIME
2EA2
2EA2 7C                    LD A,H
2EA3 B5                    OR L
2EA4 2816                  JR Z,SLND         ;NULL STRING, NOT ERROR, IF E.G. (2 TO 0)
2EA6
2EA6 E5          SLSE2:    PUSH HL
2EA7 DF                    RST 18H
2EA8 E1                    POP HL            ;SECOND NUMB IN HL, FIRST IN DE
2EA9 FE29                  CP ")"
2EAB 20E0                  JR NZ,NONSH
2EAD
2EAD ED52        SLDEF:    SBC HL,DE         ;SUB 2ND,1ST (NC HERE)
```

```
2EAF 010000               LD BC,0           ;NUL LEN IF EG (5 TO 2)
2EB2 380A                 JR C,SLNUL
2EB4
2EB4 3ACF5A               LD A,(TEMPB2)
2EB7 A7                   AND A
2EB8 CA312E               JP Z,SWER2
2EBB
2EBB 23                   INC HL
2EBC
2EBC 44          SLND:    LD B,H
2EBD 4D                   LD C,L            ;BC=STR LEN
2EBE
2EBE F1          SLNUL:   POP AF
2EBF E1                   POP HL            ;STRING START IN AHL (HL=8000-BFFF)
2EC0 19                   ADD HL,DE         ;ADD START, FIRST SLICER NUMBER-1 (DEFAULT=0)
2EC1 DCA41F               CALL C,PGOA       ;ADJUST FOR PAGE OVER FLOW IF NEEDED
2EC4 CB74                 BIT 6,H
2EC6 2803                 JR Z,SLDF2
2EC8
2EC8 CBB4                 RES 6,H
2ECA 3C                   INC A
2ECB
2ECB EB          SLDF2:   EX DE,HL          ;ADE=SLICER START, BC=LEN
2ECC 67                   LD H,A
2ECD
2ECD 3A3B5C      SLSTORE: LD A,(FLAGS)
2ED0 E6BF                 AND 0BFH
2ED2 323B5C               LD (FLAGS),A      ;'STRING'
2ED5 17                   RLA
2ED6 D0                   RET NC            ;RET IF NOT RUNNING
2ED7
2ED7 7C                   LD A,H            ;PAGE
2ED8 C3E91C               JP STKST0         ;STACK STRING, 'NO DELETE OF OLD'
2EDB
2EDB             ;DIM.SAM
2EDB
2EDB CDAA13      DIM:     CALL LOOKVARS
2EDE DBFB                 IN A,(URPORT)     ;PAGE OF STRING/ARRAY IF FOUND
2EE0 08                   EX AF,AF'         ;SAVE NZ IF FOUND
2EE1 C5                   PUSH BC
2EE2 CD402D               CALL SCOPNM       ;CHADP BACK IN NOW
2EE5 C1                   POP BC            ;C=TYPE BYTE OF ARRAY
2EE6 3A3F51               LD A,(TLBYTE)     ;TYPE BYTE FOR DIM NAME (VARS MAY HOLD STRING)
2EE9 E660                 AND 60H
2EEB CA290D               JP Z,NONSENSE     ;ERROR IF NO OPENING BRACKET USED
2EEE
2EEE CDC63A               CALL RUNFLG
2EF1 3809                 JR C,DIMRUN       ;JR IF RUNNING
2EF3
2EF3 CD5F2D               CALL SVDSK        ;CHECK N,N,...N)
2EF6
2EF6 FE2C        DIM2:    CP ","
2EF8 C0                   RET NZ            ;RET UNLESS ANOTHER ARRAY FOLLOWS
2EF9
2EF9 E7                   RST 20H           ;SKIP ','
2EFA 18DF                 JR DIM            ;ALLOW 'DIM A(8),B(6,5),A$(2)' ETC.
2EFC
2EFC 08          DIMRUN:  EX AF,AF'         ;PAGE OF STRLOCN
2EFD C5                   PUSH BC           ;TYPE BYTE SAVED IN C
2EFE C4582C               CALL NZ,ASDL1     ;DELETE ARRAY POINTED TO BY STRLOCN IF IT EXISTS
2F01
2F01 CDD73F               CALL SELCHADP
2F04 C1                   POP BC            ;C=TYPE/LEN BYTE
2F05 CB69                 BIT 5,C
2F07 010100               LD BC,1
2F0A 2802                 JR Z,DIM4         ;JR IF A STRING ARRAY OR STRING FOUND IN VARS
2F0C
2F0C 0E05                 LD C,5
2F0E
2F0E CDDD1C      DIM4:    CALL STACKBC      ;ON EXIT B STILL=0...DIM COUNT
2F11 FE                   DB 0FEH           ;'JR+1'
2F12
2F12 E7          DIMSZLP: RST 20H           ;SKIP ','
2F13
2F13 CDA02F               CALL GETSUBS      ;GET SUBS-1 IN HL
2F16 23                   INC HL
2F17 E5                   PUSH HL           ;STACK DIM SIZE ON MACHINE STACK BEHIND DIM COUNTER
2F18 C5                   PUSH BC
2F19 CDD61C               CALL STACKHL      ;AND ON FPCS
2F1C
2F1C EF                   DB CALC
2F1D 00                   DB MULT           ;GET E.G. 5*DIM1*DIM2 OR 1*DIM1
2F1E 33                   DB EXIT
2F1F
2F1F C1                   POP BC
2F20 04                   INC B             ;INC DIM COUNTER
2F21 DF                   RST 18H
2F22 FE2C                 CP ","
2F24 28EC                 JR Z,DIMSZLP
2F26
2F26 CD953A               CALL INSISCBRK    ;')'
2F29
2F29 C5                   PUSH BC           ;B=DIMS
2F2A 68                   LD L,B
2F2B 2600                 LD H,0            ;HL=DIMS
2F2D 29                   ADD HL,HL         ;GET SPACE NEEDED BY WORD DIM SIZE INFO, PLUS 1
2F2E 23                   INC HL            ;FOR NO. OF DIMS
2F2F CDD61C               CALL STACKHL
2F32
2F32 EF                   DB CALC           ;'TEXT' SIZE, DIM INFO SIZE
2F33 06                   DB SWOP           ;DIM INFO SIZE, 'TEXT' SIZE
2F34 25                   DB DUP            ;DIS, TS, TS
2F35 1C                   DB SWOP13         ;TS, TS, DIS
```

```
2F36 01                    DB ADDN            ;TS, TS+DIS
2F37 25                    DB DUP             ;TS, TS+DIS, TS+DIS
2F38 26                    DB ONELIT
2F39 0E                    DB 14              ;TS, TS+DIS, TS+DIS, 14
2F3A 01                    DB ADDN            ;TS, TS+DIS, TS+DIS+14
2F3B 33                    DB EXIT            ;ARRAY 'TEXT' SIZE, SIZE LESS HDR, TOTAL ARRAY SIZE
2F3C
2F3C CD8C3F                CALL UNSTLEN       ;GET ABC=TOTAL LEN (PAGE FORM)
2F3F CD8E2F                CALL SAROOM        ;OPEN ABC BYTES AT END OF SAVARS, LDIR T/L BYTE,
2F42                                          ;NAME TO START, EXIT WITH DE PTING TO PAST NAME
2F42 D5                    PUSH DE
2F43 CD8C3F                CALL UNSTLEN       ;SIZE EXCLUDING 14-BYTE HEADER
2F46 EB                    EX DE,HL           ;ADE=SIZE INFO FOR AFTER 14-BYTE HDR
2F47 E1                    POP HL
2F48 77                    LD (HL),A          ;PAGES
2F49 23                    INC HL
2F4A 73                    LD (HL),E
2F4B 23                    INC HL
2F4C 72                    LD (HL),D          ;LEN MOD 16K
2F4D 23                    INC HL
2F4E F1                    POP AF
2F4F 77                    LD (HL),A          ;DIM COUNT
2F50 5F                    LD E,A
2F51 1600                  LD D,0
2F53 19                    ADD HL,DE
2F54 19                    ADD HL,DE          ;PT TO LOCN FOR MSB OF LAST DIM
2F55 54                    LD D,H
2F56 5D                    LD E,L             ;SAVE IT IN DE TOO
2F57
2F57 C1          DIMENTLP: POP BC             ;POP A DIM SIZE
2F58 70                    LD (HL),B
2F59 2B                    DEC HL
2F5A 71                    LD (HL),C          ;ENTER IT IN ARRAY HEADER
2F5B 2B                    DEC HL             ;(DIM SIZES COME OFF STACK IN REVERSE ORDER)
2F5C 3D                    DEC A
2F5D 20F8                  JR NZ,DIMENTLP
2F5F
2F5F D5                    PUSH DE
2F60 CD8C3F                CALL UNSTLEN
2F63 CD2120                CALL AHLNORM       ;GET LEN-TO-CLEAR AS 19-BIT NUMBER
2F66 D1                    POP DE
2F67 EB                    EX DE,HL           ;ADE=19 BIT NO.
2F68 23                    INC HL             ;HL PTS TO START OF AREA TO CLEAR
2F69 43                    LD B,E             ;B=LEN MOD 256
2F6A 5A                    LD E,D
2F6B 57                    LD D,A             ;DE=256-BYTE PAGES
2F6C 78                    LD A,B
2F6D A7                    AND A
2F6E 2801                  JR Z,DIMNAC
2F70
2F70 13                    INC DE             ;INC DE (UNLESS B=0). ALLOWS B AND DE TO ACT AS
2F71                                          ;SEPARATE COUNTERS. DE IS *NEVER* ZERO
2F71 3A6051      DIMNAC:   LD A,(TLBYTE+33)
2F74 E640                  AND 40H            ;Z IF NUMERIC
2F76 0E20                  LD C," "
2F78 2001                  JR NZ,GARC         ;JR IF STR ARRAY, C=' ' FOR CLEARING ARRAY
2F7A
2F7A 4F                    LD C,A             ;USE ZERO TO CLEAR NUMERIC ARRAYS
2F7B
2F7B CDEF3F      GARC:     CALL CHKHL         ;CHECK WE ARE IN 8000-BFFF AREA
2F7E
2F7E 71          DIMCLP:   LD (HL),C
2F7F 23                    INC HL
2F80 10FC                  DJNZ DIMCLP
2F82
2F82 1B                    DEC DE
2F83 7A                    LD A,D
2F84 B3                    OR E
2F85 20F4                  JR NZ,GARC
2F87
2F87 CDD73F                CALL SELCHADP
2F8A DF                    RST 18H
2F8B C3F62E                JP DIM2
2F8E
2F8E
2F8E F5          SAROOM:   PUSH AF
2F8F CDB61F                CALL ADDRELND      ;ADDRESS ELINE AND DEC PTR - PT TO END OF SAVARS
2F92 F1                    POP AF
2F93 CD1C1E                CALL MKRBIG        ;OPEN ABC BYTES (PAGE FORM) AT (HL)
2F96 EB                    EX DE,HL           ;DE PTS TO ROOM
2F97 216051                LD HL,TLBYTE+33
2F9A 010B00                LD BC,11
2F9D EDB0                  LDIR               ;COPY TYPE/LEN AND NAME TO SAVARS
2F9F C9                    RET
2FA0
2FA0             ;ENTRY: CHAD PTS TO A SUBSCRIPT VALUE. BC=LIMIT
2FA0
2FA0 C5          GETSUBS:  PUSH BC
2FA1 CDE43A                CALL EXPT1NUM
2FA4 3802                  JR C,GTSBC
2FA6
2FA6 C1                    POP BC
2FA7 C9                    RET                ;JUST CHECK FOR A NUMBER IN SYNTAX TIME
2FA8
2FA8             ;ENTRY: CHAD PTS TO A SUBSCRIPT VALUE. RUN TIME!
2FA8             ;ACTION: GET A SUBSCRIPT VALUE IN HL, CHECKING IT IS <= LIMIT IN BC, AND <>0
2FA8             ;BC AND DE UNCHANGED. HL DECED BEFORE RETURN
2FA8
2FA8             ;GTSUBS:    PUSH BC
2FA8              ;          CALL EXPT1NUM
2FA8
2FA8 CD2E1D      GTSBC:    CALL GETINT        ;IN HL, A=L
2FAB C1                    POP BC             ;LIMIT VALUE
```

```
2FAC B4                     OR H
2FAD 2805                   JR Z,SWSIG        ;SUBSCRIPT 0 IS ALWAYS AN ERROR
2FAF
2FAF 2B                     DEC HL            ;REDUCE BY 1 TO GIVE ALLOWED RANGE 0 TO LIMIT-1
2FB0 ED42                   SBC HL,BC         ;IF BC=FFFF, 1-FFFF IS OK
2FB2 09                     ADD HL,BC
2FB3 D8                     RET C
2FB4
2FB4 AF           SWSIG:    XOR A
2FB5 32CF5A                 LD (TEMPB2),A
2FB8 C9                     RET
2FB9
2FB9                        INCLUDE FN.SAM         ;DEF FN, FN, COMPILE, DEF PROC, LOCAL, PROC
2FB9              ;FN.SAM
2FB9              ;COMPILE DEF PROCS
2FB9
2FB9 3A765A       ELCOMAL:  LD A,(REFFLG)
2FBC FE01                   CP 1              ;CY IF ZERO
2FBE 3F                     CCF               ;CY IF NZ (FN USED)
2FBF
2FBF DCD22F       COMALL:   CALL C,COMDF      ;COMPILE FNS
2FC2
2FC2 CD1930       COMDP:    CALL COMLEN       ;SWITCH IN PROG, GET BC AND B' AS PROG OR ELINE LEN
2FC5
2FC5 16FD         CMDPL:    LD D,0FDH
2FC7 CD4C30                 CALL LKCALL       ;LOOK FOR PROC CALLING BUFFER FROM HL ONWARDS
2FCA D8                     RET C             ;RET IF NO MORE
2FCB                                          ;ELSE HL POINTS TO BUFFER, PAGED IN
2FCB C5                     PUSH BC           ;BYTES LEFT
2FCC CDCC30                 CALL LOOKDP       ;LOOK FOR DEF PROC name, ALTER CALLING BUFFER TO
2FCF C1                     POP BC            ;PAGE/ADDR IF FOUND, ELSE FLAG AS 'NO DEF PROC'
2FD0 18F3                   JR CMDPL
2FD2
2FD2              ;COMPILE DEF FNS
2FD2              ;FIRST, DO A PASS TO MAKE A TABLE OF ALL DEF FNS AS PAGE/ADDR; LOOKING
2FD2              ;THROUGH THE PROGRAM EACH TIME IS TOO SLOW.
2FD2
2FD2 21004F       COMDF:    LD HL,INSTBUF
2FD5 22C85A                 LD (TEMPW1),HL    ;INIT PTR TO TABLE STORE
2FD8 CD321F                 CALL ADDRPROG     ;SWITCH IN PROG
2FDB 7E                     LD A,(HL)
2FDC 3C                     INC A
2FDD C8                     RET Z             ;RET IF NO PROGRAM
2FDE
2FDE 23                     INC HL
2FDF 23                     INC HL
2FE0 23                     INC HL
2FE1
2FE1 23           DFPPL:    INC HL
2FE2 22975A                 LD (CHAD),HL      ;CHAD STARTS AT 1ST CHAR OF FIRST LINE, LATER
2FE5                                          ;IS RESET TO JUST AFTER EACH DEF FN FOUND
2FE5 1EC8                   LD E,0C8H         ;DEFFNTOK
2FE7 CD9C1D                 CALL SRCHPROG     ;LOOK FOR DEF FN FROM CHAD ON
2FEA 301D                   JR NC,CMDF2       ;JR IF ALL DONE
2FEC                                          ;ELSE HL POINTS PAST 'DEF FN'
2FEC EB                     EX DE,HL
2FED 1B                     DEC DE            ;DE PTS TO 'DEF FN'
2FEE 2AC85A                 LD HL,(TEMPW1)
2FF1 0103AF                 LD BC,-INSTBUF-509
2FF4 09                     ADD HL,BC
2FF5 3810                   JR C,TMDERR       ;'Too many definitions' IF PTR >=INSTBUF+509
2FF7                                          ;ALLOWS UP TO 170 DEF FNS.
2FF7 ED42                   SBC HL,BC
2FF9 DBFB                   IN A,(251)
2FFB 77                     LD (HL),A
2FFC 23                     INC HL
2FFD 73                     LD (HL),E
2FFE 23                     INC HL
2FFF 72                     LD (HL),D
3000 23                     INC HL
3001 22C85A                 LD (TEMPW1),HL
3004 EB                     EX DE,HL
3005 18DA                   JR DFPPL
3007
3007              ;USED BY 'DEF FN' COMPILER AND 'DEF KEYCODE'
3007
3007 CF           TMDERR:   RST 08H
3008 34                     DB 52             ;'Too many definitions'
3009
3009 CD1930       CMDF2:    CALL COMLEN       ;SWITCH IN PROG, GET BC AND B' AS PROG OR ELINE LEN
300C
300C 16FE         CMDFL:    LD D,0FEH
300E CD4C30                 CALL LKCALL       ;LOOK FOR CALLING BUFFER FROM HL ONWARDS
3011 D8                     RET C             ;RET IF NO MORE
3012                                          ;ELSE HL POINTS TO BUFFER, PAGED IN
3012 C5                     PUSH BC           ;PROG LEN LEFT (MOD 16K)
3013 CDAA30                 CALL LOOKDF       ;LOOK FOR DEF FN name, ALTER CALLING
3016 C1                     POP BC            ;BUFFER TO PAGE/ADDR IF FOUND, OR FLAG 'NO DEF FN'
3017 18F3                   JR CMDFL
3019
3019
3019 3A405B       COMLEN:   LD A,(COMPFLG)
301C 17                     RLA
301D 380E                   JR C,PRGLEN       ;JR IF PROGRAM BEING COMPILED
301F
301F CD351F                 CALL ADDRELN      ;ELSE IT IS ELINE
3022 E5                     PUSH HL           ;ELINE
3023 EB                     EX DE,HL
3024 4F                     LD C,A            ;CDE=ELINE
3025 2A915A                 LD HL,(WORKSP)
3028 3A905A                 LD A,(WORKSPP)
302B 180C                   JR CPLENC
302D
```

```
302D                ;SWITCH IN PROG, GET B'=8K BLOCKS IN PROG LEN, PLUS 1, BC=LEN MOD 8K
302D
302D CD321F     PRGLEN:     CALL ADDRPROG
3030 E5                     PUSH HL             ;PROG
3031 EB                     EX DE,HL
3032 4F                     LD C,A              ;CDE=PROG
3033 2A885A                 LD HL,(NVARS)
3036 3A875A                 LD A,(NVARSP)
3039
3039 CDE71F     CPLENC:     CALL SUBAHLCDE      ;GET PROG LEN (PAGE FORM)
303C E5                     PUSH HL
303D 29                     ADD HL,HL
303E 29                     ADD HL,HL
303F 29                     ADD HL,HL
3040 17                     RLA                 ;A=8K BLOCKS
3041 D9                     EXX
3042 3C                     INC A
3043 47                     LD B,A
3044 D9                     EXX
3045 C1                     POP BC
3046 78                     LD A,B
3047 E61F                   AND 1FH
3049 47                     LD B,A              ;BC=LEN MOD 8K
304A E1                     POP HL              ;PROG
304B C9                     RET
304C
304C                ;LKCALL - LOOK FOR PROC OR FN CALL BUFFER
304C                ;BY CHECKING FOR NOT-0E 0E FD/FE FD/FE PG+80H/ADDR/LETTER.
304C                ;NO CONFUSION WITH NORMAL 0E FORMS BECAUSE THOSE ARE NOT FOLLOWED BY A LETTER,
304C                ;OR PRECEDED BY 0EH.
304C                ;NO CONFUSION WITH LINE NUMBERS BECAUSE 0E FE FE 80 = LINE LEN >32768!
304C
304C                ;ENTRY: HL=START OF SEARCH, D=TARGET (FD/FE=PROC/FN), BC=LEN TO SEARCH, MOD 8K
304C                ;B'=8K BLOCKS (ALLOWS ROOM AFTER TARGET FOUND, TO INCREMENT PTRS)
304C                ;EXIT: HL POINTS TO LOCN FOR 'PAGE' IN CALLING BUFFER, BUFFER HOLDS LEN/NAME
304C                ;OF CALLING NAME, IF NC, ELSE CY SHOWS NO TARGETS FOUND.
304C
304C 78         LKCALL:     LD A,B
304D B1                     OR C
304E 284E                   JR Z,LPC5           ;JR IF TIME FOR NEXT BLOCK
3050
3050 7A                     LD A,D              ;FE IF FN BUFFER WANTED, FD IF PROC CALL BUFFER
3051 EDB1                   CPIR
3053 2049                   JR NZ,LPC5          ;JR IF FD/FE NOT FOUND (BC=0)
3055
3055 BE                     CP (HL)             ;CHECK FOR SECOND FDH/FEH
3056 20F4                   JR NZ,LKCALL        ;KEEP LOOKING IF RED HERRING
3058
3058 2B                     DEC HL
3059 2B                     DEC HL
305A 3E0E                   LD A,0EH
305C BE                     CP (HL)             ;THERE SHOULD BE A PRECEDING 0EH
305D 2804                   JR Z,LPC3
305F
305F 23         LPC2:       INC HL
3060 23                     INC HL
3061 18E9                   JR LKCALL           ;IF THERE IS NOT, CONTINUE SEARCH
3063
3063 2B         LPC3:       DEC HL
3064 BE                     CP (HL)             ;THERE SHOULD ONLY BE *ONE* PRECEDING 0EH
3065 23                     INC HL              ;(EXCLUDE SPURIOUS 0E 0E FE FE 80 41, SAY)
3066 23                     INC HL
3067 23                     INC HL
3068 28E2                   JR Z,LKCALL         ;LOOP BACK IF SPURIOUS (JUST AN ODD NUMBER)
306A
306A 23                     INC HL              ;PT TO PROBABLE PAGE
306B 7E                     LD A,(HL)
306C 2B                     DEC HL              ;TO SECOND FD/FE AGAIN
306D 17                     RLA
306E 30DC                   JR NC,LKCALL        ;JR IF NOT 0E FE FE (>=80H).
3070                                            ;(AT SYNTAX CHECK, CALLING BUFFER IS CREATED WITH
3070                                            ;'PAGE' FD OR FEH)
3070 E5                     PUSH HL             ;PTR TO SECOND FD/FE
3071 C5                     PUSH BC             ;LEN LEFT TO SEARCH FOR OTHER BUFFERS
3072 2B                     DEC HL              ;FIRST FD/FE
3073 2B                     DEC HL              ;0EH
3074 01FF00                 LD BC,00FFH         ;NAME LEN WILL INC TO ZERO START VALUE. B=0
3077
3077 2B         FDFLP:      DEC HL
3078 0C                     INC C               ;NAME LEN
3079 7E                     LD A,(HL)
307A FE24                   CP "$"
307C 28F9                   JR Z,FDFLP          ;NAME CAN END IN '$'
307E
307E CD413B                 CALL ALNUMUND       ;CHECK FOR VALID NAME CHARS (ALPHA-NUMERICS OR '_')
3081 38F4                   JR C,FDFLP          ;LOOP BACK PAST THE NAME, TO THE FFH FN LEADER,
3083                                            ;SPACE, CC, MSB OF LINE LEN OR ':', OR FFH AT
3083                                            ;SAVARS END (IF PROC)
3083 3C                     INC A
3084 2007                   JR NZ,LPC4          ;JR IF NOT FF 42 ('FN')
3086
3086 7A                     LD A,D
3087 FEFD                   CP 0FDH
3089 2802                   JR Z,LPC4           ;JR IF PROC
308B
308B 23                     INC HL              ;SKIP FFH
308C 0D                     DEC C               ;DEC LEN BECAUSE OF FALSE INCLUSION OF 42H
308D
308D 23         LPC4:       INC HL              ;PT TO FIRST NAME CHAR
308E 114051                 LD DE,NMBUFF
3091 79                     LD A,C
3092 3D                     DEC A               ;**
3093 E61F                   AND 1FH             ;**
```

```
3095 3C                    INC A            ;** LIMIT LDIR LEN
3096 4F                    LD C,A           ;**
3097 12                    LD (DE),A        ;LEN AT BUFFER START
3098 13                    INC DE
3099 EDB0                  LDIR             ;COPY NAME TO BUFFER+1
309B C1                    POP BC           ;BYTES LEFT
309C E1                    POP HL           ;PTR TO SECOND FD/FE (WHERE CPIR HALTED)
309D C9                    RET
309E
309E 0620        LPC5:     LD B,20H         ;C=0. DO ANOTHER 8K. (IF BLOCKS NOT ZERO YET)
30A0 CDEF3F                CALL CHKHL
30A3 D9                    EXX
30A4 05                    DEC B
30A5 D9                    EXX
30A6 20A4                  JR NZ,LKCALL     ;DO B' BLOCKS
30A8 37                    SCF              ;NO MORE CALLING BUFFERS
30A9 C9                    RET
30AA
30AA             ;LOOK DEF FN
30AA             ;LOOK FOR 'DEF FN' FOLLOWED BY A SPECIFIC NAME, AND PATCH FN CALL BUFFER
30AA
30AA E5          LOOKDF:   PUSH HL          ;CALLING BUFFER ADDR (SECOND FD/FE)
30AB DBFB                  IN A,(251)
30AD F5                    PUSH AF          ;CALLING BUFFER PAGE
30AE 21004F                LD HL,INSTBUF    ;START OF TABLE OF PAGE/ADDR FOR EACH DEF FN
30B1
30B1 ED4BC85A    LKDFLP:   LD BC,(TEMPW1)   ;END OF TABLE (PAST LAST ENTRY)
30B5 A7                    AND A
30B6 ED42                  SBC HL,BC        ;NC IF PTR HAS REACHED END
30B8 09                    ADD HL,BC
30B9 303D                  JR NC,LKDP4      ;JR IF ALL TABLE ENTRIES TRIED WITHOUT SUCCESS.
30BB                                        ;MARK BUFFER 'NO DEF FN'
30BB 7E                    LD A,(HL)        ;PORT VALUE
30BC 23                    INC HL
30BD 5E                    LD E,(HL)
30BE 23                    INC HL
30BF 56                    LD D,(HL)
30C0 23                    INC HL
30C1 D3FB                  OUT (251),A
30C3 E5                    PUSH HL
30C4 CD1331                CALL MATCHER     ;MATCH DE+1 VS. BUFFER
30C7 E1                    POP HL
30C8 38E7                  JR C,LKDFLP      ;LOOP IF MATCH FAILED
30CA
30CA 1823                  JR LKDP3         ;ELSE JR AND PATCH BUFFER
30CC
30CC
30CC             ;LOOK DEF PROC
30CC             ;LOOK FOR 'DEF PROC' FOLLOWED BY A SPECIFIC NAME, AND PATCH PROC CALL BUFFER
30CC             ;AT HL WITH PAGE/ADDR OF ADDR AFTER NAME, OR PAGE FFH IF NOT FOUND
30CC             ;ENTRY: NAME LEN IS IN BUFFER, FOLLOWED BY NAME. HL POINTS TO PROC CALL BUFFER,
30CC             ;SWITCHED IN
30CC
30CC E5          LOOKDP:   PUSH HL          ;CALLING BUFFER ADDR
30CD DBFB                  IN A,(251)
30CF F5                    PUSH AF          ;CALLING BUFFER PAGE
30D0 CD321F                CALL ADDRPROG    ;START AT (PROG)
30D3 FE                    DB 0FEH          ;'JR +1'
30D4
30D4 19          LKDPLP:   ADD HL,DE        ;PT TO START OF NEXT LINE
30D5
30D5 01CA21                LD BC,2100H+0CAH ;DEFPROCTOK
30D8 CD3C34                CALL LKFC        ;LOOK FOR A DEF PROC AT LINE STARTS
30DB 381B                  JR C,LKDP4       ;JR IF NONE FOUND - MARK BUFFER 'NO DEF PROC'
30DD                                        ;ELSE HL PTS TO FIRST CHAR IN LINE,
30DD                                        ;CHAD PTS TO 'DEF PROC'
30DD
30DD D5                    PUSH DE          ;LEN OF TEXT
30DE E5                    PUSH HL          ;START OF TEXT IN THIS LINE
30DF ED5B975A              LD DE,(CHAD)
30E3 CD1331                CALL MATCHER     ;MATCH (DE+1) VS. (BUFFER+1)
30E6 E1                    POP HL
30E7 D1                    POP DE
30E8 38EA                  JR C,LKDPLP      ;JR IF FAILED TO MATCH
30EA
30EA 2B                    DEC HL
30EB 2B                    DEC HL
30EC 2B                    DEC HL
30ED 2B                    DEC HL
30EE EB                    EX DE,HL         ;ELSE PT DE TO LINE START
30EF
30EF 0680        LKDP3:    LD B,80H
30F1
30F1 DBFB        LKDP35:   IN A,(251)
30F3 E61F                  AND 1FH
30F5 B0                    OR B
30F6 47                    LD B,A           ;B=PAGE WITH DEF PROC NAME, BIT 7 SET
30F7 21                    DB 21H           ;'JR+2'
30F8
30F8 06FF        LKDP4:    LD B,0FFH        ;'NO DEF PROC/DEF FN'
30FA
30FA F1                    POP AF
30FB D3FB                  OUT (251),A      ;BACK TO PROC CALL PAGE
30FD E1                    POP HL
30FE E5                    PUSH HL
30FF 23                    INC HL
3100 70                    LD (HL),B        ;PAGE
3101 23                    INC HL
3102 73                    LD (HL),E
3103 23                    INC HL
3104 72                    LD (HL),D        ;ADDR OF PAST NAME OR LINE START (OR JUNK IF B=FF)
3105 E1                    POP HL           ;PTR TO SECOND FD/FE (WHERE CPIR STOPPED)
3106 C9                    RET
```

```
3107
3107             ;MATCH (HL) VS (FIRLET) OVER T/L+1 BYTES (NAMELEN)
3107             ;EXIT AS MATCHER - NOTE *DE* PTS PAST NAME
3107
3107 113F51      MATCHERF:  LD DE,TLBYTE
310A
310A             ;CALLED BY FOR WITH DE=TLBYTE+33
310A
310A EB          MATCHFN:   EX DE,HL
310B 7E                     LD A,(HL)
310C E61F                   AND 1FH
310E 3C                     INC A
310F 47                     LD B,A            ;B=NAME LEN
3110 23                     INC HL
3111 1806                   JR MTCCM
3113
3113             ;MATCH NAME AT (DE+1) VS NAME AT BUFFER+1 OVER (BUFFER) BYTES. SPACES IN
3113             ;(DE+1) NAME IRRELEVANT (BUT PROC AND FN NAMES HAVE NO SPACES). SPACES
3113             ;SHOULD NOT BE PRESENT IN BUFFER NAME.
3113             ;EXIT: NC IF MATCHED OK, DE PTS PAST CANDIDATE NAME, HL PAST BUFFER NAME
3113             ;USES HL, B, A. MOVES DE
3113
3113 214051      MATCHER:   LD HL,NMBUFF      ;NAME WE ARE LOOKING FOR IS AT BUFFER+1
3116 46                     LD B,(HL)         ;NAME LEN
3117 23                     INC HL
3118
3118 13          MSKIP:     INC DE
3119
3119 1A          MTCCM:     LD A,(DE)
311A FE20                   CP 20H
311C 28FA                   JR Z,MSKIP        ;SKIP ANY SPACES IN CANDIDATE NAME
311E                                          ;MIGHT BE USED BY PROC SRS??!!
311E AE                     XOR (HL)
311F 23                     INC HL
3120 E6DF                   AND 0DFH          ;IGNORE CASE MISMATCH.
3122 37                     SCF
3123 C0                     RET NZ            ;RET IF FAILED (CY)
3124
3124 10F2                   DJNZ MSKIP        ;MATCH B SIGNIF. CHARS
3126
3126 13                     INC DE
3127 1A                     LD A,(DE)         ;WE MATCHED OK SO FAR - BUT CANDIDATE NAME MUST
3128 C3413B                 JP ALNUMUND       ;END NOW - GET CY IF NOT TERMINATED
312B
312B             ;FN HANDLING (CALLED BY EVALUATOR)
312B             ;E.G. PRINT FN OCTAL(345)
312B             ;     PRINT FN OCTAL 0E FE FE PG+80H ADDRL ADDRH (345 0E 1 2 3 4 5)
312B             ;     PAGE AND ADDR ARE PTR TO '(' OR '=' IN DEF FN
312B
312B CDC63A      IMFN:      CALL RUNFLG
312E D2FF31                 JP NC,FNSYN       ;JR IF NOT RUNNING
3131
3131 E7                     RST 20H           ;SKIP 'FN'
3132 3E0E                   LD A,0EH
3134
3134 BE          FNRL:      CP (HL)
3135 23                     INC HL
3136 20FC                   JR NZ,FNRL         ;LOOP TILL 0EH FOUND
3138
3138 23                     INC HL            ;SKIP FEH
3139 23                     INC HL            ;SKIP FEH
313A 46                     LD B,(HL)         ;PAGE OF DEF FN, +80H
313B CB68                   BIT 5,B
313D 2027                   JR NZ,MDFERR      ;ERROR IF NO DEF FN
313F
313F 23                     INC HL
3140 5E                     LD E,(HL)
3141 23                     INC HL
3142 56                     LD D,(HL)         ;ADDR OF PAST NAME IN DEF FN
3143 22975A                 LD (CHAD),HL      ;PT CHAD TO END OF BUFFER
3146 E7                     RST 20H
3147 FE28                   CP "("
3149 200A                   JR NZ,FNBF2
314B
314B CDBC33                 CALL FORESP       ;GET SIGNIF CHAR AFTER '('
314E FE29                   CP ")"
3150 2002                   JR NZ,FNBF        ;JR IF NO '()' TO SKIP
3152
3152 E7                     RST 20H           ;SKIP '('
3153 E7                     RST 20H           ;SKIP ')'
3154
3154 BF          FNBF:      CP A              ;Z
3155
3155 08          FNBF2:     EX AF,AF'         ;Z IF FN HAS BRACKETS (IF EMPTY, CHAD PTS PAST)
3156 EB                     EX DE,HL
3157 78                     LD A,B
3158 CDDF3F                 CALL TSURPG       ;HL POINTS TO '(' OR '=' IN DEF FN E.G.
315B                                          ;DEF FN TEST=123 OR DEF FN TEST(A,B)=A*B
315B 7E                     LD A,(HL)
315C D63D                   SUB "="
315E 2008                   JR NZ,FNBC
3160
3160 F5                     PUSH AF           ;DEFADD=00XX
3161 08                     EX AF,AF'
3162 205D                   JR NZ,FNR6        ;EVAL RESULT OF NO-PARAM DEF FN IF FN HAS NO
3164                                          ;BRACKETS EITHER. ELSE ERROR
3164 CF          PARAMERR:  RST 08H
3165 1A                     DB 26             ;'Parameter error'
3166
3166 CF          MDFERR:    RST 08H
3167 07                     DB 7              ;'FN without DEF FN'
3168
3168 08          FNBC:      EX AF,AF'
```

```
3169 20F9                   JR NZ,PARAMERR    ;IF DEF FN HAS BRACKETS, FN MUST TOO
316B
316B CDBC33                 CALL FORESP       ;SKIP '(', GET NEXT SIGNIF CHAR
316E D629                   SUB ")"
3170 F5                     PUSH AF
3171 2845                   JR Z,FNRLE        ;JR IF NO PARAMS IN DEF FN - DEFADD=00XX
3173
3173 F1                     POP AF
3174 E5                     PUSH HL           ;ADDR USED FOR DEFADD
3175
3175 23          FNRLA:     INC HL            ;INITIALLY, SKIP VAR LETTER
3176 7E                     LD A,(HL)
3177 FE0E                   CP 0EH
3179 20FA                   JR NZ,FNRLA       ;LOOP TILL 0EH MARKER OF PARAM BUFFER IN DEF FN FND
317B
317B DBFB                   IN A,(251)
317D F5                     PUSH AF           ;PAGE OF DEF FN
317E 2B                     DEC HL
317F 7E                     LD A,(HL)         ;'$' OR NUMERIC VAR LETTER PRECEDES 0EH BUFFER
3180 D623                   SUB "$"-1
3182 23                     INC HL
3183 23                     INC HL            ;PT TO 5-BYTE BUFFER
3184 F5                     PUSH AF           ;A=1 IF '$'
3185 E5                     PUSH HL
3186 CDD73F                 CALL SELCHADP     ;LOOK AT CHAD (FN '(' OR ',')
3189 CD073B                 CALL SEXPTEXPR    ;EVAL FN ARG. Z IF STRING
318C D1                     POP DE            ;5 BYTE BUFFER IN DEF FN
318D C1                     POP BC            ;B=1 IF STRING VALUE EXPECTED
318E 2802                   JR Z,FNR3         ;JR IF STRING VALUE
3190
3190 1002                   DJNZ FNR4         ;JR IF TYPE OK
3192
3192 10D0        FNR3:      DJNZ PARAMERR     ;JR IF TYPE MISMATCH
3194
3194 FE29        FNR4:      CP ")"
3196 2003                   JR NZ,FNR5
3198
3198 08                     EX AF,AF'
3199 E7                     RST 20H           ;SKIP FINAL ')' IF WE REACHED IT
319A 08                     EX AF,AF'
319B
319B 08          FNR5:      EX AF,AF'         ;SAVE CHAR AFTER FN EXPR.
319C F1                     POP AF
319D D3FB                   OUT (251),A       ;DEF FN PAGE SELECTED
319F CD121D                 CALL FDELETE      ;DEL. EXPR (FROM FN PARAM). HL PTS TO IT ON EXIT.
31A2 010500                 LD BC,5
31A5 EDB0                   LDIR              ;COPY TO DEF FN BUFFER
31A7 EB                     EX DE,HL          ;HL POINTS PAST 5 BYTES IN DEF FN BRACKETS
31A8 CDBD33                 CALL FORESP1      ;LOOK FOR SIGNIFICANT CHAR IN DEF FN - ')' OR ','
31AB 47                     LD B,A
31AC 08                     EX AF,AF'         ;CHAR AFTER FN EXPR - ')' OR ','
31AD B8                     CP B
31AE 20B4                   JR NZ,PARAMERR
31B0
31B0 FE2C                   CP ","
31B2 28C1                   JR Z,FNRLA        ;JR IF ANOTHER PARAM SHOULD FOLLOW
31B4
31B4 FE29                   CP ")"
31B6 20AC                   JR NZ,PARAMERR
31B8
31B8 23          FNRLE:     INC HL
31B9 7E                     LD A,(HL)
31BA FE3D                   CP "="
31BC 20FA                   JR NZ,FNRLE       ;LOOP TILL DEF FN '=' FOUND
31BE
31BE 010100                 LD BC,1           ;INC CHAD TO SKIP FINAL FN ')'
31C1
31C1 ED5B975A    FNR6:      LD DE,(CHAD)
31C5 3A965A                 LD A,(CHADP)
31C8 47                     LD B,A
31C9 3A535C                 LD A,(DEFADDP)
31CC 4F                     LD C,A
31CD 22975A                 LD (CHAD),HL      ;PT CHAD TO '=' IN DEF FN
31D0 2A545C                 LD HL,(DEFADD)
31D3 E3                     EX (SP),HL
31D4 22545C                 LD (DEFADD),HL    ;PT DEFADD TO DEF FN BRACKETS
31D7 DBFB                   IN A,(251)
31D9 32535C                 LD (DEFADDP),A
31DC 32965A                 LD (CHADP),A
31DF D5                     PUSH DE           ;CHAD
31E0 C5                     PUSH BC           ;SAVE CHAD (POINTER TO ')' AFTER FN, OR SOME OTHER
31E1                                          ;CHAR IF NO PARAMS. SAVE DEFADD TO ALLOW FN CALLING
31E1                                          ;FN.
31E1
31E1 CDDF3F                 CALL TSURPG
31E4 CD073B                 CALL SEXPTEXPR    ;SKIP '=', EVAL
31E7 C1                     POP BC
31E8 79                     LD A,C
31E9 32535C                 LD (DEFADDP),A
31EC 78                     LD A,B
31ED F5                     PUSH AF
31EE CDCE3F                 CALL SETCHADP
31F1 F1                     POP AF            ;$/N FLAG
31F2 E1                     POP HL
31F3 22975A                 LD (CHAD),HL
31F6 E1                     POP HL
31F7 22545C                 LD (DEFADD),HL
31FA
31FA C0          FNTYP:     RET NZ            ;RET IF NUMERIC - 'FN' IS NOMINALLY NUMERIC, SO
31FB                                          ;RET SETS FLAGS OK
31FB C1                     POP BC            ;ELSE JUNK RET
31FC C39415                 JP STRCONT        ;GOTO STRING FLAG SETTER
31FF
```

```
31FF             ;CHECK FN SYNTAX
31FF
31FF E7          FNSYN:     RST 20H           ;SKIP 'FN'
3200 32765A                 LD (REFFLG),A     ;NZ SHOWS FN USED IN THIS LINE (FOR COMPILER)
3203 CD2732                 CALL FNNAME
3206 F5                     PUSH AF           ;A=1 IF STRING
3207 3EFE                   LD A,0FEH
3209 CD5333                 CALL MKCLBF
320C FE28                   CP "("
320E 2013                   JR NZ,FNSY5       ;JR IF NO PARAMS E.G. FN TEST
3210
3210 E7                     RST 20H           ;SKIP '('
3211 FE29                   CP ")"
3213 280D                   JR Z,FNSY4        ;JR IF NO PARAMS E.G. FN TEST ()
3215
3215 CDE314      FNARL:     CALL SCANNING     ;ALLOW NUMERIC OR STRING EXPR
3218 79                     LD A,C
3219 FE29                   CP ")"
321B 2805                   JR Z,FNSY4
321D
321D CD853A                 CALL INSISCOMA
3220 18F3                   JR FNARL
3222
3222 E7          FNSY4:     RST 20H           ;SKIP ')'
3223
3223 F1          FNSY5:     POP AF
3224 3D                     DEC A             ;Z IF STRING
3225 18D3                   JR FNTYP          ;IF NOT STRING, RET TO SET NUMERIC RESULT
3227                                          ;ELSE JUNK RET ADDR FOR NUMERIC IMMED FNS
3227                                          ;FN NAME ENDED '$'; SET FLAG FOR STRING
3227
3227
3227             ;CHECK FN NAME (CALLED BY FN AND DEF FN)
3227
3227 CDEC3A      FNNAME:    CALL GETALPH      ;INSIST NAME STARTS WITH A LETTER
322A
322A 23          DFNLP:     INC HL
322B 7E                     LD A,(HL)
322C CD413B                 CALL ALNUMUND
322F 38F9                   JR C,DFNLP        ;NAME MAY CONTINUE WITH LETTERS, NUMBERS OR '_'
3231
3231 D624                   SUB "$"
3233 2001                   JR NZ,FNN2
3235
3235 23                     INC HL            ;SKIP '$'
3236
3236 3C          FNN2:      INC A             ;A=1 IF STRING-TYPE NAME
3237 C9                     RET
3238
3238 CDC63A      LOCAL:     CALL RUNFLG
323B 1620                   LD D," "          ;NULL PREVENTS 'LOCAL REF X' BEING ACCEPTED
323D 3055                   JR NC,DPSY2        ;JR IF SYNTAX TIME
323F
323F AF                     XOR A
3240 32A85A                 LD (PRPTRP),A
3243 215004                 LD HL,CARET
3246 22A95A                 LD (PRPTR),HL     ;PT HL SO LOW THAT IT STAYS ON CR IN ROM
3249 CD3B1F                 CALL ADDRCHAD
324C 32AB5A                 LD (DPPTRP),A
324F 22AC5A                 LD (DPPTR),HL
3252 CD2C33                 CALL DPRA         ;GET PROC RET ADDR IN HL, PAGE/TYPE IN A, STAT IN C
3255 47                     LD B,A
3256 C5                     PUSH BC
3257 E5                     PUSH HL
3258 CD7334                 CALL PROP2        ;PROPAR, WITHOUT PLACING TERMINATOR
325B CDB437                 CALL PTTODP       ;**
325E 2AC45B                 LD HL,(BSTKEND)
3261 2B                     DEC HL
3262 2B                     DEC HL
3263 2B                     DEC HL
3264 2B                     DEC HL
3265 D1                     POP DE            ;ADDR
3266 C1                     POP BC
3267 70                     LD (HL),B         ;TYPE/PAGE
3268 79                     LD A,C
3269 C31219                 JP SEDA           ;STACK STAT, ADDR
326C
326C CDC63A      DEFPROC:   CALL RUNFLG
326F 3008                   JR NC,DPROC2
3271
3271 D1                     POP DE            ;NEXT STAT RET ADDR
3272 11CB8D                 LD DE,THENTOK*256+0CBH   ;ENDPROCTOK
3275                                          ;(THENTOK=NULL)
3275 CD470D                 CALL SEARCH       ;CONTINUE AFTER END PROC, OR ERROR
3278 0D                     DB 13             ;'No END PROC'
3279
3279 DF          DPROC2:    RST 18H
327A CDEC3A                 CALL GETALPH      ;INSIST ON A LETTER AS FIRST CHAR OF DEF PROC NAME
327D
327D 23          DPNMLP:    INC HL
327E 7E                     LD A,(HL)
327F CD413B                 CALL ALNUMUND
3282 38F9                   JR C,DPNMLP
3284
3284 22975A                 LD (CHAD),HL
3287 DF                     RST 18H
3288 FEB9                   CP 0B9H           ;DATATOK
328A 2002                   JR NZ,DPSY1
328C
328C E7                     RST 20H           ;SKIP 'DATA'
328D C9                     RET
328E
```

```
328E CD7A3A     DPSY1:      CALL CRCOLON
3291 C8                     RET Z               ;RET IF PARAMS FINISHED
3292
3292 16CE                   LD D,0CEH           ;REFTOK
3294
3294            ;'LOCAL' SYNTAX CHECK ENTERS HERE WITH D=NULL (SPACE)
3294
3294 BA         DPSY2:      CP D                ;'REF' IF DEF PROC, NULL IF LOCAL
3295 2001                   JR NZ,DPSY3         ;IF D IS NULL, *ALWAYS* JR
3297
3297 E7                     RST 20H             ;SKIP 'REF'
3298
3298 D5         DPSY3:      PUSH DE
3299 CDC732                 CALL VARAR          ;CHECK FOR '()' FORMS
329C C4AA13                 CALL NZ,LOOKVARS    ;CALL IF NOT ONE
329F CD793A                 CALL RCRC           ;RST 18, CRCOLON
32A2 D1                     POP DE
32A3 C8                     RET Z               ;RET IF PARAMS FINISHED
32A4
32A4 CD853A                 CALL INSISCOMA
32A7 18EB                   JR DPSY2
32A9
32A9
32A9            ;CHECK FOR VAR NAME SUCH AS frogs, price of bread, name$, abc2
32A9            ;ENTRY: CHAD PTS TO NAME START.
32A9            ;EXIT: CHECK FOR LEN OK FOR SPECIAL FORM E.G.  TEST$(), ALPHA()
32A9            ;C=LEN (EXCLUDING SPACES). CY IF OK, NC IF TOO LONG, OR NOT A VAR NAME.
32A9
32A9 DF         VARNAME:    RST 18H
32AA CD1E3B                 CALL ALPHA
32AD D0                     RET NC              ;RET WITH NC IF NOT LEGAL FIRST CHAR
32AE
32AE E5                     PUSH HL
32AF 01000B                 LD BC,0B00H         ;NAME LEN MAX (STR/ARRAY) OF 10. INIT LEN=0
32B2
32B2 E7         VNMLP:      RST 20H
32B3 0C                     INC C               ;LEN
32B4 CD413B                 CALL ALNUMUND
32B7 38F9                   JR C,VNMLP
32B9
32B9 FE24                   CP "$"
32BB 2001                   JR NZ,VNM2
32BD
32BD E7                     RST 20H             ;SKIP '$'
32BE
32BE 79         VNM2:       LD A,C              ;NAME LEN
32BF B8                     CP B
32C0 E3                     EX (SP),HL
32C1 22975A                 LD (CHAD),HL        ;ORIG CHAD
32C4 E1                     POP HL
32C5 7E                     LD A,(HL)           ;A=CHAR AFTER NAME
32C6 C9                     RET                 ;CY IF LEN OK
32C7
32C7
32C7            ;CHECK FOR ODD FORM SUCH AS FRED$(), DOGSARRAY().
32C7            ;USED BY 'LOCAL' AND 'DEF PROC'
32C7            ;IF NOT SPECIAL FORM, CHAD IS UNCHANGED, NZ. ELSE CHAD/HL POINT PAST, A=CHAR
32C7
32C7 CDA932     VARAR:      CALL VARNAME
32CA 3D                     DEC A               ;NZ - A IS A SIGNIF CHAR.
32CB D0                     RET NC              ;RET WITH ORIG CHAD IF NOT LEGAL FOR '()' FORM
32CC
32CC FE27                   CP "("-1
32CE C0                     RET NZ              ;RET IF NORMAL VAR.
32CF
32CF CDBC33                 CALL FORESP
32D2 FE29                   CP ")"
32D4 C0                     RET NZ              ;RET IF E.G. ALPHA(8) - EXPRESSION
32D5
32D5 22975A                 LD (CHAD),HL
32D8 E7                     RST 20H             ;SKIP ')'
32D9 BF                     CP A                ;Z=SPECIAL FORM SKIPPED
32DA C9                     RET
32DB
32DB            ;PROCEDURES
32DB            ;ENTRY: CHAR BETWEEN 0-8FH FOUND WHEN CMD EXPECTED
32DB
32DB C690       PROCS:      ADD A,90H           ;CORRECT FOR PREVIOUS SUB 90H
32DD CDEC3A                 CALL GETALPH
32E0 DF                     RST 18H
32E1 CDC63A                 CALL RUNFLG
32E4 304E                   JR NC,PROCSY        ;JR IF SYNTAX CHECK
32E6
32E6 3E0E                   LD A,0EH
32E8
32E8 BE         PRRL:       CP (HL)
32E9 23                     INC HL
32EA 20FC                   JR NZ,PRRL          ;LOOP TILL CALLING BUFFER FOUND
32EC
32EC 23                     INC HL              ;SKIP FD
32ED 23                     INC HL              ;SKIP FD
32EE 46                     LD B,(HL)           ;PAGE
32EF CB68                   BIT 5,B
32F1 203F                   JR NZ,MDPERR        ;(BIT 7 IS ALWAYS SET, BIT 6=EXTERNAL CMD, BIT 5
32F3                                            ;IS SET IF 'NO DEF PROC')
32F3 23                     INC HL
32F4 5E                     LD E,(HL)
32F5 23                     INC HL
32F6 56                     LD D,(HL)           ;ADDR OF DEF PROC LINE OR EXEC CODE
32F7 23                     INC HL
32F8 22A95A                 LD (PRPTR),HL
32FB DBFB                   IN A,(251)
32FD 32A85A                 LD (PRPTRP),A       ;SET UP PROC PTR
```

```
3300 78                  LD A,B
3301 32AB5A              LD (DPPTRP),A      ;DEF PROC PTR PAGE
3304 CDDF3F              CALL TSURPG        ;SELECT DEF PROC OR EXEC CODE PAGE
3307 210500              LD HL,5
330A 19                  ADD HL,DE          ;SKIP TO FIRST POSSIBLE DEF PROC NAME START POSN
330B CDBD33              CALL FORESP1       ;SKIP ANY SPACES/CC
330E
330E 23          PRPNM:  INC HL
330F 7E                  LD A,(HL)
3310 CD413B              CALL ALNUMUND
3313 38F9                JR C,PRPNM         ;LOOP PAST DEF PROC NAME
3315
3315 22AC5A              LD (DPPTR),HL      ;PAST NAME
3318 D5                  PUSH DE            ;DEF PROC LINE ADDR
3319 CD6834              CALL PROPAR        ;PROCESS PARAMETERS
331C 0640                LD B,40H
331E CDF118              CALL BSTKE         ;STACK A 'PROC-TYPE' RET ADDR
3321 34                  INC (HL)           ;RET TO *NEXT* STATEMENT
3322 E1                  POP HL             ;DEF PROC LINE ADDR
3323 D1                  POP DE             ;JUNK NEXT STAT
3324 3AAB5A              LD A,(DPPTRP)
3327 0E02                LD C,2             ;STAT 2
3329 C34A19              JP RLEPC2
332C
332C                ;UNSTACK PROC RETURN ADDR, OR ERROR
332C
332C 0640        DPRA:   LD B,40H           ;'PROC' TYPE
332E CD201A              CALL RETLOOP       ;GET C=STAT, HL=ADDR, NZ IF ERROR, A=PAGE
3331 C8                  RET Z
3332
3332 CF          MDPERR: RST 08H
3333 0C                  DB 12              ;'Missing DEF PROC'
3334
3334 23          PROCSY: INC HL
3335 7E                  LD A,(HL)
3336 CD413B              CALL ALNUMUND
3339 38F9                JR C,PROCSY
333B
333B 3EFD                LD A,0FDH
333D CD5333              CALL MKCLBF        ;MAKE BUFFER AFTER NAME
3340 CD7A3A              CALL CRCOLON
3343 C8                  RET Z              ;RET IF NO PARAMS
3344
3344 CDC732      PCSYL:  CALL VARAR         ;CHECK FOR '()' FORMS, SKIP IF SEEN
3347 C4E314              CALL NZ,SCANNING   ;IF NOT '()' FORM ,EVAL
334A CD793A              CALL RCRC          ;RST 18, CRCOLON
334D C8                  RET Z
334E
334E CD853A              CALL INSISCOMA
3351 18F1                JR PCSYL
3353
3353                ;MAKE CALLING BUFFER AT HL (0E A A A ? ?), PT CHAD AFTER IT TO SIGNIF CHAR, GET
3353                ;IT IN A. ENTRY: A=BYTE TO FILL FIRST 3 LOCNS WITH.
3353
3353 F5          MKCLBF: PUSH AF
3354 CD6433              CALL MAKESIX       ;OPEN 6 BYTES AFTER NAME FOR ADDRESS BUFFER,
3357                                        ;START WITH 0EH
3357 F1                  POP AF
3358 77                  LD (HL),A
3359 23                  INC HL
335A 77                  LD (HL),A
335B 23                  INC HL
335C 77                  LD (HL),A
335D 23                  INC HL
335E 23                  INC HL             ;BUFFER= 0E FE FE FE ?? ??
335F 22975A              LD (CHAD),HL       ;PT CHAD TO PAST BUFFER
3362 E7                  RST 20H
3363 C9                  RET
3364
3364 010600      MAKESIX:LD BC,6
3367 CD1B1E              CALL MAKEROOM
336A 360E                LD (HL),0EH
336C 23                  INC HL
336D C9                  RET
336E
336E                ;LOOK FOR A FN VARIABLE
336E                ;CALLED FROM EVAL WHEN DEFADD IS SET DURING EVAL (OF DEF FN RESULT)
336E                ;ENTRY: HL PTS TO FIRST CHAR OF NAME
336E                ;RETS WITH NC IF NOT FOUND, HL SAME - CONTINUE SEARCH IN NORMAL VARS AREA.
336E                ;ELSE CY, VAR ALREADY STACKED, NAME SKIPPED, STR/NUM SET, A='$' IF STRING
336E
336E CDC63A      LKFNVAR:CALL RUNFLG
3371 D0                  RET NC             ;RET TO CALLER (EVAL) WITH NC IF SYNTAX TIME
3372
3372 E5                  PUSH HL            ;IN CASE VAR NOT FOUND
3373 46                  LD B,(HL)          ;VAR LETTER
3374 E7                  RST 20H
3375 FE24                CP "$"
3377 0E0E                LD C,0EH
3379 2002                JR NZ,LKFV0
337B
337B 4F                  LD C,A             ;C='$'
337C E7                  RST 20H
337D
337D CD413B      LKFV0:  CALL ALNUMUND
3380 3824                JR C,LKFVF         ;NO SEARCH FOR E.G. 'TEST', 'X1', 'ABC$' - JUST
3382                                        ;SINGLE LETTER VARS
3382 21535C              LD HL,DEFADD-1
3385 CD401F              CALL ASV2          ;PT TO PAST '(' IN DEF FN
3388
3388 23          LKFV1:  INC HL
3389 7E                  LD A,(HL)
338A FE0E                CP 0EH
```

```
338C 20FA                  JR NZ,LKFV1       ;LOOK FOR A BUFFER MARKER
338E
338E 2B                    DEC HL
338F 7E                    LD A,(HL)
3390 B9                    CP C              ;C='$' IF STRING WANTED, OR 0EH FOR NULL
3391 2003                  JR NZ,LKFV2       ;JR IF A NUMERIC BUFFER FOUND
3393
3393 2B                    DEC HL
3394 7E                    LD A,(HL)         ;STRING LETTER
3395 23                    INC HL
3396
3396 A8          LKFV2:    XOR B             ;DESIRED LETTER
3397 E6DF                  AND 0DFH
3399 280E                  JR Z,LKFVM        ;JR IF MATCHED
339B
339B 110700                LD DE,7
339E 19                    ADD HL,DE         ;PT PAST 5-BYTE BUFFER
339F CDBD33                CALL FORESP1
33A2 FE2C                  CP ","
33A4 28E2                  JR Z,LKFV1        ;KEEP CHECKING IF MORE BUFFERS FOLLOW
33A6
33A6 E1          LKFVF:    POP HL            ;ORIG HL
33A7 A7                    AND A             ;NC - SEARCH FAILED
33A8 C9                    RET
33A9
33A9 23          LKFVM:    INC HL            ;PT TO 0EH
33AA 23                    INC HL            ;PT TO BUFFER
33AB 79                    LD A,C
33AC CD201D                CALL HLTOFPCS     ;STACK BUFFER DATA
33AF C1                    POP BC            ;ORIG HL
33B0 213B5C                LD HL,FLAGS
33B3 CBB6                  RES 6,(HL)
33B5 FE24                  CP "$"
33B7 37                    SCF
33B8 C8                    RET Z             ;CY SHOWS FOUND
33B9
33B9 CBF6                  SET 6,(HL)
33BB C9                    RET
33BC
33BC             ;INC PTR HL AND RET WHEN IT POINTS TO SIGNIF CHAR
33BC
33BC 23          FORESP:   INC HL
33BD
33BD             ;AVOIDS INITIAL INC
33BD
33BD 7E          FORESP1:  LD A,(HL)
33BE FE21                  CP 21H
33C0 D0                    RET NC
33C1
33C1 18F9                  JR FORESP
33C3
33C3             ;CALLED BY LOAD, DELETE, KEYIN, RENUM
33C3
33C3 3EFF        SCOMP:    LD A,0FFH
33C5 32405B                LD (COMPFLG),A
33C8 C9                    RET
33C9
33C9             ;FROM DELETE/MERGE
33C9
33C9 3A475C      GT4R:     LD A,(SUBPPC)
33CC 2A455C                LD HL,(PPC)
33CF 11                    DB 11H            ;'JR+2'
33D0
33D0             ;FROM RENUM/KEYIN
33D0
33D0 F1          GT4P:     POP AF            ;STAT
33D1 E1                    POP HL            ;LINE
33D2
33D2             ;ENTRY: A=STAT-1 TO GO TO, HL=LINE
33D2
33D2 3C                    INC A
33D3 CD9B19                CALL GOTO4
33D6 3EFF                  LD A,0FFH
33D8 32465C                LD (PPC+1),A      ;ENSURE "GOTO" SEARCHES PROG FROM START
33DB
33DB CDC333      DOCOMP:   CALL SCOMP
33DE
33DE             ;CALLED AS ELINE IS EXECUTED, AND BY CLEAR/RUN. CHAD IS AUTO-ADJUSTED BECAUSE
33DE             ;VAR CREATION MIGHT MOVE IT
33DE
33DE 3A965A      COMPILE:  LD A,(CHADP)
33E1 32995A                LD (KCURP),A
33E4 2A975A                LD HL,(CHAD)
33E7 229A5A                LD (KCUR),HL      ;NEEDS AUTO-ADJUST
33EA 2AAF5A                LD HL,(CLA)
33ED E5                    PUSH HL
33EE 3AAE5A                LD A,(CLAPG)
33F1 F5                    PUSH AF
33F2 3A405B                LD A,(COMPFLG)
33F5 A7                    AND A
33F6 282C                  JR Z,COMPILEL     ;JR IF ONLY ELINE NEEDS COMPILE
33F8
33F8             ;DO LABELS
33F8 CD321F                CALL ADDRPROG     ;START AT (PROG)
33FB
33FB 01EA21      DOLBLP:   LD BC,2100H+0EAH  ;LABELTOK
33FE CD3C34                CALL LKFC         ;FIND A LABEL
3401 381A                  JR C,LABSD        ;END IF ALL LABELS PROCESSED
3403                                         ;ELSE HL POINTS TO FIRST CHAR IN LINE, CHAD
3403 DBFB                  IN A,(251)        ;PTS TO 'LABEL', DE=LINE LEN
3405 32965A                LD (CHADP),A      ;KEEP CHAD UP TO DATE FOR ASSIGN ETC.?!!?
3408 E5                    PUSH HL
3409 19                    ADD HL,DE         ;PTR TO NEXT LINE
```

```
340A E3                     EX (SP),HL
340B 2B                     DEC HL
340C 2B                     DEC HL
340D 2B                     DEC HL
340E 4E                     LD C,(HL)
340F 2B                     DEC HL
3410 46                     LD B,(HL)           ;BC=LINE NUMBER OF LINE WITH LABEL
3411 CDDD1C                 CALL STACKBC
3414 CDEA38                 CALL SVNUMV         ;ASSESS NUMERIC VARIABLE
3417 CD0F2B                 CALL ASSIGN         ;ASSIGN FPC VALUE TO VAR
341A E1                     POP HL
341B 18DE                   JR DOLBLP           ;CONTINUE SEARCH AT NEXT LINE
341D
341D                LABSD:                      ;'COMPILING PROGRAM'SHOWN BY BIT 7 COMPFLG HI
341D CDBF2F                 CALL COMALL         ;COMPILE PROCS/FN (CY HERE)
3420
3420 AF                     XOR A
3421 32405B                 LD (COMPFLG),A      ;'COMPILING ELINE'SHOWN BY BIT 7 LOW
3424
3424 CDB92F         COMPILEL:  CALL ELCOMAL     ;*ALWAYS* COMPILE ELINE FOR PROCS, AND FNS
3427                                            ;IF NEEDED.
3427 F1                     POP AF
3428 32AE5A                 LD (CLAPG),A
342B E1                     POP HL
342C 22AF5A                 LD (CLA),HL
342F 2A9A5A                 LD HL,(KCUR)
3432 22975A                 LD (CHAD),HL
3435 3A995A                 LD A,(KCURP)
3438 C3CE3F                 JP SETCHADP
343B
343B                ;LKFC.SAM - LOOK FIRST CHAR
343B                ;LOOK FOR CHAR IN C, AT START OF LINES, SKIPPING ANY PRECEDING CC/SPACES
343B                ;ENTRY: B=21H FOR CC/SPACE SKIP, C=CHAR, HL=START
343B                ;EXIT: NC IF FOUND, HL=FIRST CHAR IN LINE (MAY NOT BE C) (CHAD)=TARGET
343B                ;      CY IF SEARCHED PROG AND FAILED.
343B
343B 19             LKFCLP:    ADD HL,DE
343C
343C                ;ENTRY POINT
343C
343C 7E             LKFC:      LD A,(HL)
343D C601                      ADD A,1
343F D8                        RET C            ;RET IF END OF PROGRAM
3440
3440 CDEF3F                    CALL CHKHL       ;KEEP LINE START IN SECTION C
3443 23                        INC HL
3444 23                        INC HL
3445 5E                        LD E,(HL)
3446 23                        INC HL
3447 56                        LD D,(HL)        ;DE=LINE LNE
3448 23                        INC HL
3449 7E                        LD A,(HL)        ;FIRST CHAR - QUICK CHECK FOR MOST COMMON CASE
344A                                            ;(NO PRECEDING JUNK)
344A B9             LKFC2:     CP C
344B 2817                      JR Z,LKFC5       ;JR IF GOT IT - BUT MIGHT NEED TO SKIP CC/SPACE
344D
344D B8                        CP B             ;CP 21H
344E 30EB                      JR NC,LKFCLP     ;LOOP IF CHAR WAS SIGNIF - THIS LINE WON'T DO
3450
3450 E5                        PUSH HL          ;SAVE LINE START
3451
3451 FE0D           LKFCSK:    CP 0DH
3453 280C                      JR Z,LKFC4       ;STOP TRYING IF LINE ENDS
3455
3455 23                        INC HL           ;SKIP CC/SPACE
3456 7E                        LD A,(HL)
3457 B8                        CP B             ;CP 21H
3458 38F7                      JR C,LKFCSK      ;SKIP ALL CC/SPACES
345A
345A 22975A                    LD (CHAD),HL     ;ALTER CHAD IN CASE WE HAVE A MATCH
345D E1                        POP HL           ;FIRST CHAR IN LINE PTR
345E B9                        CP C             ;CP FIRST SIGNIF CHAR IN LINE WITH DESIRED.
345F C8                        RET Z
3460
3460 E5                        PUSH HL
3461
3461 E1             LKFC4:     POP HL           ;FIRST CHAR IN LINE PTR
3462 18D7                      JR LKFCLP
3464
3464 22975A         LKFC5:     LD (CHAD),HL
3467 C9                        RET
3468
3468
3468                           INCLUDE NPARPRO.SAM
3468                ;PARPRO.SAM
3468                ;PARAM PROCESSING
3468                ;E.G.      TEST 1,A,C,2
3468                ;DEF PROC TEST A,B,C,D
3468
3468                ;CHECK DEF PROC.
3468                ;IF NON-REF NUMERIC:
3468                ;  LOOK FOR VAR:
3468                ;    IF NOT FOUND:
3468                ;      GOTO STEP 2
3468                ;    IF FOUND:
3468                ;    1.SAVE ADDR OF TYPE/LEN BYTE SO IT CAN BE MADE "INVISIBLE" WHEN ALL PARAMS
3468                ;      HAVE BEEN PROCESSED, AND "REVEALED" AGAIN AT END PROC TIME.
3468                ;      KEEP LOOKING SO LAST LINK IN LETTER"S "CHAIN" FOUND.
3468                ;    2.CREATE NEW VARIABLE WITH SAME NAME, USING VALUE FROM PROC CALL. THIS IS
3468                ;      "INVIS" TILL ALL PARAMS PROCESSED, SINCE BIT 7 SET ON TLBYTE.
3468                ;      KEEP PTR TO NEW VARIABLE SO IT CAN BE MARKED "UNUSED" (BIT 5 SET) AT END
3468                ;      PROC TIME. PROCS USE SPECIAL ASSIGN ROUTINE THAT LOOKS FOR AN "UNUSED"
3468                ;      VAR OF THE SAME TYPE (FOR-NEXT/NORMAL) AND NAME LEN IN THE LETTER LIST
```

```
3468             ;      AND DOES A NORMAL ASSIGN IF THERE ISN"T ONE, OR OVERWRITES THE NAME AND
3468             ;      VALUE OF THE UNUSED VAR AND RESETS BIT 5 TO SHOW "USED". THIS AVOIDS
3468             ;      ACCUMULATION OF "UNUSED" VARS.
3468             
3468             ;IF REF NUMERIC:
3468             ;  TREAT LIKE NON-REF NUMERIC, EXCEPT THAT DURING ASSIGNMENT OF LOCAL VERSION,
3468             ;  THE "VALUE" SUPPLIED BY THE PROC LIST MUST BE A VARIABLE NAME. IT IS MADE
3468             ;  INVISIBLE WHEN PARAMS HAVE BEEN PROCESSED, AND ITS ADDR IS STORED SO THAT AT
3468             ;  END PROC IT CAN BE "REVEALED" AND ITS VALUE RESET FROM THE LOCAL VERSION.
3468             
3468             ;IF NON-REF STRING:
3468             ;  LOOK FOR VAR:
3468             ;    IF NOT FOUND:
3468             ;      FLAG "NO INVIS VERSION"
3468             ;      GOTO STEP 2
3468             ;    IF FOUND:
3468             ;    1.SAVE ADDR OF TYPE/LEN BYTE SO IT CAN BE MADE "INVISIBLE" WHEN ALL PARAMS
3468             ;      HAVE BEEN PROCESSED, AND FLAG "INVIS EXISTS".
3468             ;    2.CREATE NEW VARIABLE WITH SAME NAME, USING VALUE FROM PROC CALL. THIS WILL
3468             ;      BE IGNORED IF VAR. USED LATER IN PROC CALL LIST (IT IS AT SAVARS END).
3468             ;      KEEP NAME OF NEW VARIABLE SO IT CAN BE DELETED AT END PROC TIME, IF IT
3468             ;      EXISTS, AND SO A SEARCH CAN BE MADE FOR ANY "INVIS" VERSION IF FLAG
3468             ;      SAYS ONE EXISTS
3468             
3468             ;IF REF STRING/ARRAY:
3468             ;  LOOK FOR VAR:
3468             ;    IF NOT FOUND:
3468             ;      GOTO STEP 2
3468             ;    IF FOUND:
3468             ;    1.SAVE ADDR OF TYPE/LEN BYTE SO IT CAN BE MADE "INVISIBLE" WHEN ALL PARAMS
3468             ;      HAVE BEEN PROCESSED, AND FLAG "INVIS EXISTS".
3468             ;    2.LOOK FOR VAR NAMED IN PROC CALL. COPY TLBYTE AND NAME TO STORE. COPY
3468             ;      NAME GIVEN IN DEF PROC TO TLBYTE. (RENAME).
3468             ;      AT END PROC, FIND VAR, COPY ORIG NAME BACK FROM STORE.
3468             
3468             ;PROCESS PARAMS
3468             ;ENTRY: PRPTR PTS TO PROC PARAMS, DPPTR PTS TO DEF PROC PARAMS
3468             
3468 2AC45B      PROPAR:    LD HL,(BSTKEND)
346B 2B                     DEC HL
346C AF                     XOR A
346D 77                     LD (HL),A
346E 2B                     DEC HL
346F 77                     LD (HL),A          ;BSTK TERMINATOR
3470 22C45B                 LD (BSTKEND),HL
3473             
3473 AF          PROP2:     XOR A
3474 F5                     PUSH AF            ;MACHINE STACK TERMINATOR
3475 21004B                 LD HL,HDR          ;USE TAPE HDR AREA AS RENAME STACK
3478 77                     LD (HL),A
3479 22725B                 LD (RNSTKE),HL
347C CDB437                 CALL PTTODP        ;POINT TO DEF PROC LIST
347F CA5936                 JP Z,PPM2          ;JP IF NO LIST
3482             
3482 FEB9                   CP 0B9H            ;DATATOK
3484 2027                   JR NZ,PPML
3486             
3486 F1                     POP AF
3487 2AA95A                 LD HL,(PRPTR)
348A 3AA85A                 LD A,(PRPTRP)
348D 180C                   JR RESTORE3
348F             
348F             ;RESTORE CMD STUCK IN HERE TO ALLOW JR!
348F             
348F CD503A      RESTORE:   CALL SYNTAX3       ;EXPT1NUM OR USE ZERO IF CR/COLON
3492             
3492 CD2E1D                 CALL GETINT
3495             
3495 CD4D1A      RESTORE2:  CALL FNDLINE       ;RETURNS ADDR OF LINE BC. PAGE MAY BE SWITCHED
3498 2B                     DEC HL
3499 DBFB                   IN A,(251)
349B             
349B 228B5A      RESTORE3:  LD (DATADD),HL
349E 328A5A                 LD (DATADDP),A
34A1 C9                     RET
34A2             
34A2             ;RESTORE-ZERO. USED BY CLEAR AND LOAD (PROGRAM)
34A2             
34A2 210000      RESTOREZ:  LD HL,0
34A5 18EE                   JR RESTORE2
34A7             
34A7             
34A7 CDB437      PPLOOP:    CALL PTTODP        ;POINT TO DEF PROC LIST
34AA CA5936                 JP Z,PPM2          ;JP IF LIST ENDED
34AD             
34AD 2AC45B      PPML:      LD HL,(BSTKEND)
34B0 11E9FF                 LD DE,-23          ;MAX SPACE NEEDED BY 1 PARAM
34B3 19                     ADD HL,DE          ;FIND NEW BSTKEND. CY
34B4 ED5BC85B               LD DE,(HEAPEND)    ;LOWER IN MEM THAN BSTKEND
34B8 ED52                   SBC HL,DE          ;BSTKEND-(HEAPEND+1) - BSTK CANNOT COME AS FAR DOWN
34BA                                           ;HEAPEND.
34BA DA0019                 JP C,BSFERR        ;ERROR IF STACK SPACE TOO SMALL
34BD             
34BD D6CE                   SUB 0CEH           ;REFTOK
34BF 32765A                 LD (REFFLG),A      ;Z IF REF VAR
34C2 2001                   JR NZ,PPNREF
34C4             
34C4 E7                     RST 20H            ;SKIP "REF"
34C5             
34C5 CDDA14      PPNREF:    CALL LVFLAGS       ;LOOK FOR VAR IN DEF PROC LIST
34C8 EB                     EX DE,HL           ;PROTECT ADDR OF PREV LINK (IF NOT FND)
34C9 2A975A                 LD HL,(CHAD)
34CC 22AC5A                 LD (DPPTR),HL      ;(DPPTRP IS OK STILL)
```

```
34CF F2A235               JP P,PPAS         ;JP IF STRING
34D2
34D2 CB69                 BIT 5,C           ;BIT 5=1 IF NUMERIC ARRAY
34D4 C2A235               JP NZ,PPAS        ;JP IF NUMERIC ARRAY
34D7
34D7 08                   EX AF,AF'
34D8 2822                 JR Z,PPA2         ;JR IF NVAR IN DEF PROC LIST DOESN"T EXIST
34DA
34DA DDE5                 PUSH IX
34DC E1                   POP HL
34DD 2B                   DEC HL            ;PT TO TLBYTE OF VAR
34DE E5                   PUSH HL           ;SAVE ADDR OF VAR TO MAKE "INVIS"
34DF DBFB                 IN A,(URPORT)
34E1 F6E0                 OR 0E0H           ;BIT 7 SET SO "ADDR TO REVEAL" STORED. 111xxxxx
34E3 F5                   PUSH AF
34E4 23                   INC HL
34E5
34E5 CDD113     PPNLP:    CALL NVMLP        ;KEEP LOOKING. C IS TLBYTE
34E8 2810                 JR Z,PPA15        ;JR IF NO SECOND COPY (D=0)
34EA
34EA 3E00                 LD A,0
34EC 324051               LD (FIRLET),A
34EF E5                   PUSH HL
34F0 77                   LD (HL),A
34F1 23                   INC HL
34F2 36FF                 LD (HL),0FFH
34F4 23                   INC HL
34F5 77                   LD (HL),A
34F6 23                   INC HL
34F7 77                   LD (HL),A         ;SECOND COPY=MINUS ZERO
34F8 D1                   POP DE            ;VALUE LOCN
34F9              ;       JR PPA2           ;VARIABLE EXISTS. IF ASSIGNMENT IS MADE FROM
34F9 21                   DB 21H            ;PROC VALUE, MINUS ZERO OVERWRITTEN. IF NOT,
34FA                                        ;DEFAULT RECOGNISES MINUS ZERO AS "NON-EXISTENT"
34FA
34FA 4A         PPA15:    LD C,D            ;SIGNAL SIMPLE NUMERIC (F-N BIT SEEN AS ARRAY)
34FB EB                   EX DE,HL
34FC
34FC CDFB2C     PPA2:     CALL SYN1PP       ;SET UP FOR ASSIGNMENT - SETS "NEW VAR" FLAG
34FF CDBC37               CALL PTTOPR       ;POINT TO PROC LIST
3502 CA2036               JP Z,PPD2         ;JP IF PROC LIST ENDED
3505
3505              ;LOOK FOR "UNUSED" NUMERIC VAR
3505              ;ENTRY: TLBYTE/FIRLET SET
3505
3505 ED4B3F51   UNVLK:    LD BC,(TLBYTE)    ;C=TLBYTE, B=FIRLET
3509 78                   LD A,B
350A 87                   ADD A,A
350B 2844                 JR Z,PPA3         ;JR IF SEARCH NOT WANTED COS 2ND COPY EXISTS
350D
350D D6C2                 SUB 61H*2
350F 5F                   LD E,A            ;LETTER TRANSFORMED TO WORD OFFSET (A=0, B=2..)
3510 1600                 LD D,0
3512 CD1F1F               CALL ADDRNV       ;PT. HL AT NUMERIC VARS, PAGED IN
3515 CBE9                 SET 5,C           ;"UNUSED"
3517 79                   LD A,C            ;DESIRED TYPE/LEN
3518 19                   ADD HL,DE         ;INDEX INTO TABLE OF WORD PTRS.
3519
3519 5E         UNVLP:    LD E,(HL)
351A 23                   INC HL            ;PTR=FFFFH IF NO MORE VARS START WITH REQUIRED
351B 56                   LD D,(HL)         ; LETTER. CAUSES CARRY AND CHECK FOR FF IN NVSPOV
351C 19                   ADD HL,DE         ;ELSE DE IS A PTR TO NEXT VAR STARTING
351D                                        ; WITH REQUIRED LETTER.
351D 3832                 JR C,PPA3         ;RET IF LIST ENDED OR SEVERE PAGE OVERFLOW
351F
351F              ;        BIT 6,H
351F              ;        CALL NZ,INCURPAGE
351F CDEF3F               CALL CHKHL
3522 79                   LD A,C            ;IN CASE CALL WAS MADE
3523 BE                   CP (HL)
3524 23                   INC HL            ;PT TO PTR LSB
3525 20F2                 JR NZ,UNVLP
3527
3527 2B                   DEC HL            ;PT TO TLBYTE OF UNUSED VAR
3528 E5                   PUSH HL
3529 7E                   LD A,(HL)
352A CBFE                 SET 7,(HL)        ;hidden** SO NOT USED TWICE!
352C 23                   INC HL
352D 23                   INC HL
352E 23                   INC HL            ;PT TO OTHER LETTERS OF NAME (OR VALUE)
352F E61F                 AND 1FH
3531 280A                 JR Z,PPA25        ;JR IF PTING TO VALUE (1-LET VAR)
3533
3533 EB                   EX DE,HL
3534 214151               LD HL,FIRLET+1
3537 4F                   LD C,A
3538 0600                 LD B,0
353A EDB0                 LDIR              ;COPY NAME TO "UNUSED" VAR NAME AREA
353C EB                   EX DE,HL
353D
353D 22A65A     PPA25:    LD (DEST),HL      ;PTR TO VALUE AREA OF REUSED VAR
3540 E1                   POP HL            ;TLBYTE PTR
3541 E5                   PUSH HL           ;ADDR OF VAR TO MARK "USED"
3542 DBFB                 IN A,(URPORT)
3544 E61F                 AND 1FH
3546 32A55A               LD (DESTP),A
3549 F620                 OR 20H            ;001XXXXX
354B F5                   PUSH AF           ;PAGE, AND BITS FOR "MARK AS USED"
354C AF                   XOR A
354D 32715C               LD (FLAGX),A      ;"VAR EXISTS" (BIT 0)
3550 DD                   DB 0DDH           ;"JR+3"
3551
3551 2A855A     PPA3:     LD HL,(NUMEND)    ;VAR WILL BE CREATED HERE SINCE AN "UNUSED" VAR
```

```
3554                                            ;AREA IS NOT BEING RE-USED.
3554
3554 0640       PPA4:       LD B,40H            ;BIT 7 OF PAGE WILL BE RESET. BIT 6 SET SO NOT
3556                                            ;SEEN AS "TERMINATOR"
3556 CDCF36                 CALL RUAHL          ;RECORD ADDR OF VAR ASSIGNED TO, ALLOWING
3559                                            ;MARKING AS "UNUSED" BY END PROC
3559 CDBC37                 CALL PTTOPR         ;PT TO PROC
355C 3A765A                 LD A,(REFFLG)
355F A7                     AND A
3560 203B                   JR NZ,PPA5          ;JR UNLESS VAR BEING "ASSIGNED" TO IS REF-TYPE
3562                                            ;NZ NEEDED HERE!!!
3562
3562 CDDA14                 CALL LVFLAGS        ;INSIST PROC SUPPLIES A VAR NAME
3565 F29036                 JP P,PARERR         ;ERROR IF STRING
3568
3568 08                     EX AF,AF'
3569 2012                   JR NZ,PPRNE         ;JR IF VAR EXISTS
356B
356B EF                     DB CALC
356C E1                     DB STKZERO
356D 25                     DB DUP
356E 33                     DB EXIT
356F
356F CD372B                 CALL ASSISR         ;IF PROC VAR DOES NOT EXIST CREATE DP VAR
3572                                            ;WITH VALUE ZERO USING NAME AT T/L+33. C=0
3572
3572 CD282B                 CALL CRTVAR35       ;LOOK FOR NON-EXISTENT PROC VAR AGAIN.
3575                                            ;SO NO MATCH.
3575                                            ;SET UP FOR ASSIGN TO PROC VAR.
3575                                            ;CREATE PROC VAR WITH VALUE ZERO
3575 3A3F51                 LD A,(TLBYTE)
3578 4F                     LD C,A
3579 CDC213                 CALL NUMLOOK        ;POINT TO (NOW EXISTING) PROC VAR
357C AF                     XOR A               ;Z
357D
357D 08         PPRNE:      EX AF,AF'           ;SAVE Z IF ASSIGN TO DP VAR ALREADY MADE
357E DD2B                   DEC IX              ;PT TO TLBYTE OF PROC VAR BEING REFFED
3580 DDE5                   PUSH IX             ;SAVE ADDR OF VAR TO MAKE "INVIS"
3582 DBFB                   IN A,(URPORT)
3584 F6E0                   OR 0E0H             ;BIT 7 SET SO "ADDR TO REVEAL" STORED. 111xxxxx
3586 F5                     PUSH AF
3587 06A0                   LD B,0A0H           ;SET BITS 5 AND 7 OF PAGE TO SHOW "REF" AND "PROC"
3589 E5                     PUSH HL
358A EB                     EX DE,HL
358B 2AC45B                 LD HL,(BSTKEND)
358E CBEE                   SET 5,(HL)          ;SET BIT 5 OF PREVIOUS "VAR TO MARK UNUSED" TO SHOW
3590 EB                     EX DE,HL            ;"REF". BIT 7 BEING RES SHOWS "DEF PROC"
3591 CDCF36                 CALL RUAHL          ;RECORD ADDR OF VALUE BEING ASSIGNED *FROM* SO IT
3594                                            ;CAN BE ASSIGNED BACK *TO* AT END PROC E.G.
3594                                            ;       TEST Z
3594                                            ; DEF PROC TEST REF X
3594                                            ; VALUE OF Z HAS BEEN ASSIGNED TO X. END PROC
3594                                            ; ASSIGNS X'S VALUE TO Z, MARKS X "UNUSED"
3594 E1                     POP HL              ;PTR TO VALUE
3595 08                     EX AF,AF'
3596 C4201D                 CALL NZ,HLTOFPCS    ;(NO CHANGE TO Z FLAG)
3599 C40F2B                 CALL NZ,ASSIGN
359C BF                     CP A                ;Z -  JR PPD1H
359D
359D            ;PPA5:      CALL SELCHADP       ;ENSURE SEARCH OF VARS HASN'T ALTERED PAGE **
359D            ;PPA5:      CALL RCRC           ;RST 18H, CALL CRCOLON
359D            ;           JR NZ,PPA55
359D
359D            ;           DB CALC
359D            ;           DB STKZERO
359D            ;           DB EXIT
359D
359D            ;           INC HL
359D            ;           DEC (HL)            ;NZ FLAG, VALUE=MINUS ZERO
359D            ;           JR PPA45
359D
359D C4FA2A     PPA5:       CALL NZ,VALFET1     ;ASSIGN VALUE FROM PROC LIST TO VAR FROM DEF PROC
35A0                                            ;EXITS WITH DEST PAGED IN
35A0 187A       PPD1H:      JR PPD1
35A2
35A2            ;STRINGS/ARRAYS
35A2
35A2 0610       PPAS:       LD B,10H            ;BIT 7 RES=NO VAR TO REVEAL, BIT 4 SET=STRING
35A4 08                     EX AF,AF'
35A5 280D                   JR Z,PAS2           ;JR IF DEF PROC STRING/ARRAY DOESN"T EXIST
35A7
35A7 0690                   LD B,90H            ;BIT 7 SET SO "GLOBAL S/A TO REVEAL" SHOWN
35A9                                            ;BIT 4 SET=STRING/ARRAY
35A9 2ABC5B                 LD HL,(STRLOCN)     ;PTS TO TLBYTE OF S/A JUST FOUND
35AC E5                     PUSH HL             ;SAVE ADDR OF DEF PROC S/A TO MAKE "INVIS"
35AD DBFB                   IN A,(URPORT)
35AF E61F                   AND 1FH             ;BIT 7 RES SO NO RECORD MADE OF ADDR TO REVEAL
35B1 F660                   OR 60H              ;SET BIT 5 - ENSURE NOT ZERO. 011xxxxx
35B3 F5                     PUSH AF
35B4
35B4 CDED36     PAS2:       CALL PPSUB          ;RECORD DEF PROC NAME ON BSTK, WITH FLAG BITS
35B7 CDBC37                 CALL PTTOPR         ;POINT TO PROC LIST
35BA 2864                   JR Z,PPD2           ;JR IF PROC LIST ENDED
35BC
35BC CD402D                 CALL SCOPNM         ;COPY DEF PROC NAME TO TLBYTE+33
35BF 3A765A                 LD A,(REFFLG)
35C2 A7                     AND A
35C3 2045                   JR NZ,PPD0          ;JR IF NOT A REF VAR
35C5
35C5            ;REF STRINGS/ARRAYS
35C5
35C5 3A3B5C                 LD A,(FLAGS)
35C8 F5                     PUSH AF
```

```
35C9 CDDA14               CALL LVFLAGS      ;LOOK FOR PROC NAME
35CC F2D535               JP P,REFSTR       ;JP IF PROC VAR=STRING
35CF
35CF F1                   POP AF            ;BIT 6,A=1 IF DP NAME=NUMERIC
35D0 CB11                 RL C              ;BIT 6,C=1 IF PROC VAR=ARRAY
35D2 A1                   AND C             ;BIT 6=1 IF NUM ARRAYS
35D3 2F                   CPL
35D4 0E                   DB 0EH            ;"JR+1"
35D5
35D5 F1          REFSTR:  POP AF
35D6
35D6 E640                 AND 40H
35D8 C29036               JP NZ,PARERR
35DB
35DB 08                   EX AF,AF'
35DC 281C                 JR Z,PAS3         ;JR IF VAR DOESN'T EXIST
35DE
35DE 2A725B               LD HL,(RNSTKE)
35E1 ED5BC45B             LD DE,(BSTKEND)   ;PTR TO DEF PROC NAME IN BSTK
35E5 23                   INC HL
35E6 73                   LD (HL),E
35E7 23                   INC HL
35E8 72                   LD (HL),D
35E9 ED5BBC5B             LD DE,(STRLOCN)   ;TLBYTE OF PROC NAME JUST FOUND
35ED 23                   INC HL
35EE 73                   LD (HL),E
35EF 23                   INC HL
35F0 72                   LD (HL),D
35F1 DBFB                 IN A,(URPORT)
35F3 F680                 OR 80H            ;ENSURE NOT ZERO
35F5 23                   INC HL
35F6 77                   LD (HL),A
35F7 22725B               LD (RNSTKE),HL
35FA
35FA 0600        PAS3:    LD B,0            ;KEEP ORIG TLBYTE
35FC CDED36               CALL PPSUB        ;RECORD ORIGINAL PROC VAR NAME SO IT CAN
35FF                                        ;BE RESTORED AT END PROC TIME.
35FF                                        ;NAME "UNDER" IT MUST BE DEF PROC NAME
35FF 2AC45B               LD HL,(BSTKEND)
3602 2B                   DEC HL
3603 36FF                 LD (HL),0FFH      ;PRECEDE NAME WITH FF
3605 22C45B               LD (BSTKEND),HL
3608 1812                 JR PPD1
360A
360A             ;NON-REF STRINGS/ARRAYS
360A
360A 3A3B5C      PPD0:    LD A,(FLAGS)
360D 87                   ADD A,A
360E FA9036               JP M,PARERR       ;ERROR IF NUMERIC ARRAY
3611
3611 216051               LD HL,TLBYTE+33
3614 CBB6                 RES 6,(HL)        ;ENSURE THAT THE TYPE IS NOT ARRAY EVEN IF DP NAME
3616                                        ;IS E.G. A$().
3616 CDA13A               CALL EXPTSTR
3619 CD7B2C               CALL ASNST        ;CREATE STRING AT END OF SAVARS
361C
361C CDD73F      PPD1:    CALL SELCHADP
361F DF                   RST 18H
3620
3620 22A95A      PPD2:    LD (PRPTR),HL     ;UPDATE PROC PTR
3623 FE29                 CP ")"
3625 2004                 JR NZ,PPD3
3627
3627 E7                   RST 20H           ;SKIP CLOSING BRACKET OF E.G. "NUM()"
3628 22A95A               LD (PRPTR),HL
362B
362B FE2C        PPD3:    CP ","
362D 2004                 JR NZ,PPD35       ;JR IF NO MORE PROC PARAMS ** BUG FIX
362F
362F E7                   RST 20H           ;SKIP "," IN PROC LIST
3630 22A95A               LD (PRPTR),HL
3633
3633 CDB437      PPD35:   CALL PTTODP
3636 FE29                 CP ")"
3638 2001                 JR NZ,PPD4
363A
363A E7                   RST 20H           ;SKIP CLOSING BRACKET OF E.G. "NUM()"
363B
363B FE2C        PPD4:    CP ","
363D 201A                 JR NZ,PPM2        ;EXIT IF NO MORE DEF PROC VARS **
363F
363F E7                   RST 20H           ;SKIP "," IN DEF PROC LIST
3640 C3AD34               JP PPML           ;LOOP UNTIL PROC PARAMS FINISHED
3643
3643             ;  PPD5:    LD A,(REFFLG)
3643               ;         AND A
3643                ;        JR NZ,PARERR      ;ERROR IF NO VALUE AND "REF" - VAR NAME REQUIRED
3643
3643                  ;      RST 20H           ;SKIP COMMA, GET HL=CHAD
3643                   ;     JR PPD2
3643
3643 F5          PPMIL:   PUSH AF
3644 CDDF3F               CALL SELURPG
3647 F1                   POP AF            ;011xxxxx IF STRING/ARRAY, NO ADDR TO REVEAL
3648                                        ;111XXXXX IF NUMBER, ADDR TO REVEAL
3648                                        ;001XXXXX IF NUMBER, ADDR TO MAKE "USED"
3648 87                   ADD A,A           ;CY IF NUMERIC WITH DISP TO STORE
3649                                        ;P IF NEED TO MAKE USED AND NO NEED TO MAKE
3649                                        ;INVISIBLE
3649 E1                   POP HL
364A CBBE                 RES 7,(HL)        ;VISIBLE**
364C F25736               JP P,PPM3
364F
```

```
364F CBFE                SET 7,(HL)         ;EXISTING NUM/STR VAR MADE "INVISIBLE"
3651 0680                LD B,80H
3653 DCCF36              CALL C,RUAHL       ;RECORD DISP FROM NVARS OF A "NUMBER TO REVEAL"
3656 21                  DB 21H             ;"JR+2"
3657
3657 CBAE       PPM3:    RES 5,(HL)         ;"USED"
3659
3659 F1         PPM2:    POP AF
365A A7                  AND A
365B 20E6                JR NZ,PPMIL        ;LOOP TILL STACKED ADDRS TO MARK "INVIS" ALL DONE
365D
365D 2A725B     RNMLP:   LD HL,(RNSTKE)
3660 7E                  LD A,(HL)
3661 A7                  AND A
3662 2828                JR Z,RNMF
3664
3664 CDDF3F              CALL SELURPG
3667 2B                  DEC HL
3668 56                  LD D,(HL)
3669 2B                  DEC HL
366A 5E                  LD E,(HL)
366B D5                  PUSH DE            ;VARS PTR
366C 2B                  DEC HL
366D 56                  LD D,(HL)
366E 2B                  DEC HL
366F 5E                  LD E,(HL)
3670 2B                  DEC HL
3671 22725B              LD (RNSTKE),HL
3674 EB                  EX DE,HL
3675 D1                  POP DE             ;HL=BSTK PTR TO DP NAME, DE=VARS PTR
3676 7E                  LD A,(HL)
3677 E60F                AND 0FH
3679 4F                  LD C,A             ;C=LEN (OF DP NAME)
367A 1A                  LD A,(DE)
367B E670                AND 70H            ;"VISIBLE"
367D B1                  OR C               ;UPPER BITS (TYPE) FROM VARS, LEN FROM D.P. NAME
367E 47                  LD B,A             ;NEW TLYBTE
367F 1A                  LD A,(DE)
3680 17                  RLA
3681 DC9236              CALL C,NEGVTR      ;IF VAR HAD BEEN MADE INVISIBLE, REVERSE "VAR TO
3684                                        ;REVEAL" BIT ON D.P. NAME IN BSTK. (HAPPENS IF
3684                                        ;ANY PROC VAR PASSED BY REFERENCE HAS SAME NAME
3684                                        ;AS AS A DEF PROC VAR. VARS THAT ARE RENAMED NEED
3684                                        ;NOT AND SHOULD NOT BE INVIS)
3684 78                  LD A,B
3685 0600                LD B,0
3687 CD0437              CALL ILDISR        ;OVERWRITE NAME OF PROC VAR WITH NAME OF DEF PROC
368A 18D1                JR RNMLP
368C
368C CDBC37     RNMF:    CALL PTTOPR
368F C8                  RET Z              ;RET IF PROC LIST ENDED
3690
3690 CF         PARERR:  RST 08H
3691 1A                  DB 26              ;"Parameter error"
3692
3692            ;LOOK FOR A PARTICULAR BSTK ENTRY SPECIFYING "VAR TO REVEAL" AND RESET BIT
3692            ;ENTRY: DE PTS TO TLBYTE OF VAR NAME IN VARS THAT IS ABOUT TO BE RENAMED
3692
3692 C5         NEGVTR:  PUSH BC            ;B=NEW TLBYTE, C=LEN OF DP NAME
3693 D5                  PUSH DE            ;VARS PTR
3694 E5                  PUSH HL
3695 2AC45B              LD HL,(BSTKEND)
3698
3698 7E         FDPNL:   LD A,(HL)
3699 A7                  AND A
369A 28F4                JR Z,PARERR        ;ERROR IF NOT FOUND
369C
369C CB67                BIT 4,A
369E 2004                JR NZ,FDPN2        ;JR IF STRING, ELSE SKIP NUMERIC DISP
36A0
36A0 0E03                LD C,3
36A2 1810                JR FDPN4
36A4
36A4 FEFF       FDPN2:   CP 0FFH
36A6 2002                JR NZ,FDPN3        ;JR IF NOT REF NAME
36A8
36A8 23                  INC HL             ;SKIP FF
36A9 7E                  LD A,(HL)
36AA
36AA E60F       FDPN3:   AND 0FH
36AC 4F                  LD C,A
36AD 1A                  LD A,(DE)
36AE AE                  XOR (HL)
36AF 23                  INC HL
36B0 E6AF                AND 0AFH
36B2 2805                JR Z,FDPN5         ;JR IF MATCH ON "VAR TO REVEAL" BIT, TYPE BITS
36B4                                        ;AND LEN BITS - IGNORE BIT 4 MISMATCH (ALWAYS
36B4                                        ;ZERO IN VARS AREA) AND BIT 6 ($/$ ARRAY)
36B4
36B4 0600       FDPN4:   LD B,0
36B6 09                  ADD HL,BC          ;SKIP STRING NAME
36B7 18DF                JR FDPNL
36B9
36B9 E5         FDPN5:   PUSH HL
36BA D5                  PUSH DE
36BB 41                  LD B,C             ;LEN TO MATCH ON
36BC
36BC 13         FDPBL:   INC DE
36BD 1A                  LD A,(DE)
36BE BE                  CP (HL)
36BF 23                  INC HL
36C0 2002                JR NZ,FDPN6
36C2
```

```
36C2 10F8                  DJNZ FDPBL
36C4
36C4 D1          FDPN6:    POP DE              ;DP NAME PTR
36C5 E1                    POP HL              ;BSTK SRCH PTR
36C6 20EC                  JR NZ,FDPN4         ;JR IF MATCH FAILED
36C8
36C8 2B                    DEC HL
36C9 CBBE                  RES 7,(HL)          ;RESET "VAR TO REVEAL" BIT
36CB E1                    POP HL              ;HL=DP NAME PTR
36CC D1                    POP DE              ;VARS PTR.
36CD C1                    POP BC
36CE C9                    RET
36CF
36CF
36CF             ;CALCULATE DISP FROM NVARS TO CURRENT PAGE/HL, STORE IN BSTK IN PAGE/MOD 16K
36CF             ;FORM, WITH BIT 7 OF PAGE SET IF VAR TO MARK UNUSED, RES IF TO MAKE VISIBLE.
36CF             ;(USED LATER BY END PROC)
36CF
36CF 3A875A      RUAHL:    LD A,(NVARSP)
36D2 4F                    LD C,A
36D3 ED5B885A              LD DE,(NVARS)       ;CDE=NVARS
36D7 DBFB                  IN A,(URPORT)       ;AHL=TLBYTE
36D9 CDE71F                CALL SUBAHLCDE
36DC EB                    EX DE,HL            ;ADE=DISP FROM NVARS (PAGEFORM)
36DD 2AC45B                LD HL,(BSTKEND)
36E0 2B                    DEC HL
36E1 72                    LD (HL),D
36E2 2B                    DEC HL
36E3 73                    LD (HL),E
36E4 2B                    DEC HL
36E5 E60F                  AND 0FH             ;BIT 4 LOW SHOWS NUMERIC
36E7 B0                    OR B                ;SHOW "DISP TO VAR TO MARK "UNUSED"" AT END PROC
36E8                                           ;(BIT 7 SET) OR "DISP TO VAR TO REVEAL" (BIT 7 RES)
36E8 77                    LD (HL),A
36E9
36E9 22C45B      BSSET:    LD (BSTKEND),HL
36EC C9                    RET
36ED
36ED             ;RECORD TLBYTE AND NAME ON BSTK, USING B REG TO SET BITS 7 AND 4 OF TLBYTE
36ED             ;EXIT: BC=0
36ED
36ED 113F51      PPSUB:    LD DE,TLBYTE        ;TLBYTE OF VAR NAME LOOKED FOR
36F0 2AC45B                LD HL,(BSTKEND)
36F3 1A                    LD A,(DE)
36F4 E60F                  AND 0FH
36F6 4F                    LD C,A
36F7 1A                    LD A,(DE)
36F8 E66F                  AND 6FH             ;**
36FA
36FA B0                    OR B                ;BIT 4 ALWAYS SET TO SHOW STRING NAME, BIT 7 SET
36FB                                           ;IF "GLOBAL TO REVEAL"
36FB 0600                  LD B,0              ;BC=NAME LEN
36FD ED42                  SBC HL,BC
36FF 2B                    DEC HL              ;ALLOW SPACE FOR TLBYTE (NON-STANDARD FORM)
3700 22C45B                LD (BSTKEND),HL
3703 EB                    EX DE,HL
3704
3704 12          ILDISR:   LD (DE),A
3705 13                    INC DE
3706 23                    INC HL
3707 EDB0                  LDIR                ;COPY NAME TO BSTK
3709 C9                    RET
370A
370A             ;SUBROUTINE USED BY END PROC AND POP (PROC ADDR)
370A             ;ACTION: FOR NUMERICS, MARKS VAR (DISP ON BSTK) "UNUSED" OR "REVEALS" IT.
370A             ;FOR STR/ARRS, DELETE, AND REVEAL LAST INVIS VERSION IF ONE WAS HIDDEN BY PROC.
370A
370A 2AC45B      DELOCAL:  LD HL,(BSTKEND)
370D 7E                    LD A,(HL)           ;0000=TERMINATOR
370E                                           ;FF=       ORIG NAME OF A REF VAR FOLLOWS. LOOK FOR
370E                                           ;          NEW NAME (FOLLOWS ORIG), RENAME. KEEP
370E                                           ;          NEW NAME ON BSTK TO ALLOW DELOCAL.
370E
370E                                           ;BIT 4 SET IF STRING/ARRAY NAME
370E                                           ;          BIT 7 SET IF GLOBAL VERSION TO REVEAL,
370E                                           ;          AS WELL AS LOCAL VERSION TO DELETE.
370E
370E                                           ;BIT 4 RES IF DISP OF NUMBER FROM NVARS
370E                                           ;          BIT 7 SET IF IT IS A GLOBAL VERSION TO
370E                                           ;          REVEAL.
370E                                           ;          BIT 7 RES IF IT IS LOCAL VAR TO MARK
370E                                           ;          "UNUSED".
370E                                           ;             BITS 7/5 SET IF VALUE OF LOCAL VAR
370E                                           ;             IS POINTED TO. COPY TO BUFFER.
370E                                           ;             BIT 7 RES, 5 SET IF BUFFER VALUE
370E                                           ;             TO BE COPIED TO VAR (ADDR=TLBYTE)
370E                                           ;             AND TLBYTE TO BE MARKED "UNUSED"
370E 23                    INC HL
370F 23                    INC HL
3710 A7                    AND A
3711 28D6                  JR Z,BSSET          ;JR IF TERMINATOR OF DATA USED BY END PROC
3713                                           ;NO NEED TO CHECK FOR 2 ZEROS HERE
3713 2B                    DEC HL
3714 CB67                  BIT 4,A
3716 2046                  JR NZ,DLOCS         ;JR IF STRING/ARRAY NAME
3718
3718 F5                    PUSH AF
3719 5E                    LD E,(HL)
371A 23                    INC HL
371B 56                    LD D,(HL)
371C 23                    INC HL
371D 22C45B                LD (BSTKEND),HL
3720 E60F                  AND 0FH
```

```
3722 4F                  LD C,A             ;CDE=DISP (PAGE FORM)
3723 CD1F1F              CALL ADDRNV
3726 CDDE1F              CALL ADDAHLCDE
3729 F1                  POP AF
372A CB6F                BIT 5,A
372C 2827                JR Z,DLOC3         ;JR IF NOT ADDR OF REF VALUE/DP VAR TLBYTE
372E
372E 17                  RLA
372F 3006                JR NC,DLOC2        ;JR IF HL POINTS TO DEF PROC VAR TLBYTE
3731                                        ;(ALWAYS FOLLOWS REF VALUE ADDR)
3731
3731 DBFB                IN A,(URPORT)      ;WE DON"T NEED THIS REF VALUE ADDR YET - SAVE
3733 F5                  PUSH AF            ;PAGE
3734 E5                  PUSH HL            ;AND ADDR.
3735 18D3                JR DELOCAL
3737
3737 CBEE        DLOC2:  SET 5,(HL)         ;VAR THAT WAS LOCAL MARKED "UNUSED"
3739 7E                  LD A,(HL)
373A E61F                AND 1FH
373C C607                ADD A,7
373E 4F                  LD C,A
373F 0600                LD B,0
3741 09                  ADD HL,BC          ;PT TO VALUE OF LOCAL VAR (LAST BYTE)
3742 11CC5A              LD DE,TEMPW1+4     ;USE TEMP AREA AS BUFFER
3745 0E05                LD C,5
3747 EDB8                LDDR               ;COPY LOCAL VAR VALUE TO BUFFER
3749 13                  INC DE
374A EB                  EX DE,HL           ;HL PTS TO TEMPW1
374B D1                  POP DE
374C F1                  POP AF
374D D3FB                OUT (URPORT),A     ;DE PTS TO VALUE AREA OF REF VAR
374F 0E05                LD C,5
3751 EDB0                LDIR
3753 18B5                JR DELOCAL
3755
3755 17          DLOC3:  RLA
3756 CBBE                RES 7,(HL)         ;"VISIBLE" (IN CASE GLOBAL VAR NEEDS IT)
3758 38B0                JR C,DELOCAL       ;JR IF IT WAS
375A
375A CBEE                SET 5,(HL)         ;VAR THAT WAS LOCAL MARKED "UNUSED"
375C 18AC                JR DELOCAL
375E
375E FEFF        DLOCS:  CP 0FFH
3760 202E                JR NZ,DLCS2        ;JR IF NOT A REF STRING/ARRAY
3762
3762 7E                  LD A,(HL)          ;ORIG TLBYTE
3763 E60F                AND 0FH
3765 3C                  INC A
3766 4F                  LD C,A
3767 0600                LD B,0
3769 E5                  PUSH HL
376A 09                  ADD HL,BC          ;PT TO DEF PROC NAME FOLLOWING ORIG NAME.
376B 4E                  LD C,(HL)          ;DEF PROC CODE BYTE (BIT 7 HOLDS EXTRA
376C                                        ;DATA, BIT 4 IS SET)
376C C5                  PUSH BC
376D 79                  LD A,C
376E E66F                AND 6FH            ;NORMAL T/L BYTE FOR DP NAME
3770                                        ;(SIMPLE $ VS ARRAY $ IRREL. TO SEARCH ROUTINE)
3770 23                  INC HL
3771 CD0D14              CALL LKBSV         ;LOOK FOR DEF PROC NAME
3774 CA082D              JP Z,VNFERR
3777
3777 C1                  POP BC             ;B=0, C=CODED BYTE
3778 D1                  POP DE             ;DE PTS TO ORIG NAME
3779 C5                  PUSH BC            ;CODE BYTE IN C
377A 1A                  LD A,(DE)
377B E60F                AND 0FH
377D 4F                  LD C,A
377E 0C                  INC C              ;BC=ORIG NAME LEN, INC T/L BYTE
377F 2ABC5B              LD HL,(STRLOCN)    ;ADDR OF T/L BYTE IN VARS
3782 AE                  XOR (HL)
3783 E60F                AND 0FH
3785 AE                  XOR (HL)           ;TYPE FROM VARS, NAME LEN FROM ORIG
3786 EB                  EX DE,HL
3787 77                  LD (HL),A
3788 EDB0                LDIR               ;COPY ORIG TLBYTE/NAME BACK TO VAR, BUT USE
378A                                        ;ACTUAL VAR'S TYPE BITS. ($ ARRAY VS $)
378A C1                  POP BC
378B 79                  LD A,C
378C E6EF                AND 0EFH           ;RES 4,A **
378E 180A                JR DLCS3           ;THERE"S NEVER A VERSION TO ERASE - RENAMED INSTEAD
3790
3790             ;LOOK FOR STRING/ARRAY AND ERASE IT IF FOUND; ALSO, IF A GLOBAL VERSION WAS
3790             ;HIDDEN, LOOK FOR IT (LAST "INVIS" VERSION OF VAR) AND REVEAL IT.
3790
3790 E6EF        DLCS2:  AND 0EFH           ;RES BIT 4 (VARS FORM ALWAYS HAS BIT 4 RESET)
3792 F5                  PUSH AF
3793 CD0D14              CALL LKBSV         ;LOOK FOR VAR NAMED ON BSTK. NZ=FND
3796
3796 C45B2C              CALL NZ,ASDEL2     ;DELETE STRING/ARRAY WITH PAGE/LEN AT HL IF IT
3799                                        ;WAS FOUND
3799 F1                  POP AF
379A
379A CB7F        DLCS3:  BIT 7,A
379C C43114              CALL NZ,STARYLK2   ;LOOK FOR INVIS VERSION IF THERE WAS A GLOBAL
379F                                        ;VERSION
379F 2810                JR Z,DELCLH        ;JR IF NONE (SHOULD BE AT LEAST ONE...)
37A1
37A1             ;FIND LAST INVISIBLE VERSION
37A1
37A1 D5          DLOCL:  PUSH DE            ;PTR TO T/L BYTE IN VARS
37A2 DBFB                IN A,(URPORT)
37A4 F5                  PUSH AF
```

```
37A5 CD6D14               CALL FLNOMTCH
37A8 C1                   POP BC
37A9 E1                   POP HL            ;BHL=ADDR OF PREVIOUS "FOUND" VAR (T/L BYTE)
37AA 20F5                 JR NZ,DLOCL       ;JR IF WE JUST FOUND ANOTHER
37AC
37AC 78                   LD A,B
37AD D3FB                 OUT (URPORT),A    ;PT TO LAST INVIS VERSION
37AF CBBE                 RES 7,(HL)        ;MAKE VISIBLE
37B1
37B1 C30A37     DELCLH:   JP DELOCAL
37B4
37B4
37B4            ;POINT CHAD TO DEF PROC PARAMETER LIST OR PROC PARAMETER LIST
37B4            ;EXIT: Z IF CHAD POINTS TO CR/COLON. A COMMA WILL BE SKIPPED.
37B4
37B4 2AAC5A     PTTODP:   LD HL,(DPPTR)
37B7 3AAB5A               LD A,(DPPTRP)
37BA 1806                 JR PTTOC
37BC
37BC 2AA95A     PTTOPR:   LD HL,(PRPTR)
37BF 3AA85A               LD A,(PRPTRP)
37C2
37C2 22975A     PTTOC:    LD (CHAD),HL
37C5 32965A               LD (CHADP),A
37C8 CDDF3F               CALL SELURPG
37CB C3793A               JP RCRC
37CE            ;          RST 18H
37CE             ;         CP 0DH
37CE              ;        RET Z
37CE
37CE            ;          CP ":"
37CE             ;         RET
37CE
37CE
37CE
37CE                      INCLUDE MISC2.SAM      ;BOOT, ERRORS, BUFMV AND STUBS FOR BUFF CMDS,
37CE            ;MISC2.SAM  RENUM, GET, AUTO, DELETE, LET,
37CE            ;OPEN, CLOSE, DEF KEYCODE, LABEL, RUN, CLEAR
37CE
37CE            ;FROM RST 08H
37CE
37CE 2A975A     ERROR2:   LD HL,(CHAD)
37D1 22A35A               LD (XPTR),HL
37D4 08                   EX AF,AF'
37D5 3A965A               LD A,(CHADP)
37D8 32A25A               LD (XPTRP),A
37DB D1                   POP DE
37DC 1A                   LD A,(DE)
37DD 13                   INC DE
37DE D5                   PUSH DE
37DF 1B                   DEC DE
37E0 2AEE5A               LD HL,(RST8V)
37E3 24                   INC H
37E4 25                   DEC H
37E5 C40500               CALL NZ,HLJUMP
37E8
37E8 3AC35B               LD A,(DOSCNT)     ;0 IF DOS NOT IN CONTROL, 1 IF DOS IN CONTROL
37EB 0F                   RRCA
37EC 1A                   LD A,(DE)         ;ERROR NUMBER
37ED 380D                 JR C,NORMERR      ;JR IF DOS RUNNING - DON'T RECURSE!
37EF
37EF 3AC25B               LD A,(DOSFLG)
37F2 A7                   AND A
37F3 1A                   LD A,(DE)
37F4 2015                 JR NZ,PTDOS        ;JR IF DOS BOOTED
37F6
37F6 FE80                 CP 128
37F8 3802                 JR C,NORMERR      ;JR IF NOT A DOS HOOK CODE
37FA
37FA 3E35       NODOS:    LD A,53           ;'NO DOS'
37FC
37FC 323A5C     NORMERR:  LD (ERRNR),A
37FF 21C35B               LD HL,DOSCNT
3802 CB86                 RES 0,(HL)        ;'DOS NOT IN CONTROL'
3804 ED7B3D5C             LD SP,(ERRSP)
3808 C3951D               JP SETSTK
380B
380B            ;PASS TO DOS
380B
380B A7         PTDOS:    AND A
380C 28EE                 JR Z,NORMERR
380E
380E 5F                   LD E,A            ;ERROR NUMBER
380F 0EFA                 LD C,250
3811 ED40                 IN B,(C)          ;B=LRPORT
3813 210000               LD HL,0
3816 39                   ADD HL,SP
3817 3AC25B               LD A,(DOSFLG)     ;0 OR DOS PAGE (1-1FH)
381A 3D                   DEC A             ;GET PAGE NO. FOR SECTION A (DOS IN SECTION B)
381B F3                   DI
381C D3FA                 OUT (250),A       ;DOS PAGED IN AT 4000H, ROM0 ON, ROM1 OFF
381E 310080               LD SP,8000H       ;STACK NOW OK
3821 FB                   EI
3822 C5                   PUSH BC           ;B=PREV LRPORT
3823 E5                   PUSH HL           ;PREV STACK PTR
3824 7B                   LD A,E            ;HOOK CODE
3825 FE80                 CP 128
3827 3004                 JR NC,DOSHK       ;JR WITH HOOK CODES
3829
3829 CD0342               CALL 4203H        ;HANDLE ERROR CODE, RATHER THAN HOOK CODE
382C 37                   SCF               ;'COMING FROM ERROR'
382D
382D D40042     DOSHK:    CALL NC,4200H     ;HANDLE HOOK CODE IN A
```

```
3830                                            ;'COMING FROM HOOK' MUST SET NC!
3830
3830 E1         DOSC:       POP HL              ;PREV STACK PTR
3831 C1                     POP BC
3832 F3                     DI
3833 ED41                   OUT (C),B           ;PREV LRPORT RESTORED
3835 F9                     LD SP,HL            ;PREV STACK
3836 FB                     EI
3837
3837 2AC05B                 LD HL,(DOSER)
383A 24                     INC H
383B 25                     DEC H
383C 2018                   JR NZ,DHLJ          ;JP TO VECTOR IF WANTED
383E
383E 3006                   JR NC,DOSNC         ;NC/CY FROM 4200/4203 - JR IF WAS HOOK CODE
3840
3840 2A3D5C                 LD HL,(ERRSP)
3843 2B                     DEC HL
3844 2B                     DEC HL
3845 F9                     LD SP,HL            ;CLEAR STACK IF WAS ERROR ENTRY
3846
3846 A7         DOSNC:      AND A
3847 20B3                   JR NZ,NORMERR       ;JR IF ERROR (NORMAL ERROR, OR DOS ERROR (81-127D)
3849
3849 1D                     DEC E
384A CAFDE1                 JP Z,LDFL           ;JP IF E WAS 1 - LOAD MAIN BODY OF FILE
384D
384D 1D                     DEC E
384E CADBE4                 JP Z,SVFL           ;JP IF E WAS 2 - SAVE ENTIRE FILE
3851
3851 1D                     DEC E
3852 CAFAE1                 JP Z,LKTH           ;JP IF E WAS 3 - LOAD ENTIRE FILE ** NEW JUMP
3855
3855 C9                     RET                 ;RET IF NO ERRORS - TO NEXT STAT IF VIA ERROR ENTRY
3856
3856 E9         DHLJ:       JP (HL)
3857
3857            ;STUBS OF ROUTINES THAT COPY FROM ROM1 TO BUFFER FOR EXECUTION:
3857
3857            MEPROG:
3857
3857 21D4C5      LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN+INPLN+DKLN+RDLN+DFNLN+TOKLN
385A DD                     DB 0DDH             ;'JR+3'
385B
385B 211DC3     INPUT:      LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN
385E DD                     DB 0DDH             ;'JR+3'
385F
385F 2100C0     RENUM:      LD HL,0C000H        ;PART 2 OF RENUM
3862 010002                 LD BC,RENLN
3865 1830                   JR BUFMV
3867
3867            ;KEYIN A$
3867
3867 2192C2     KEYIN:      LD HL,0C000H+RENLN+GETLN+DELLN
386A 11004B                 LD DE,HDR           ;SO OTHER CMDS DON'T CORRUPT KEYIN, EXCEPT
386D                                            ;LOAD/SAVE EXEC. WILL CORRUPT FIRST PART
386D                                            ;(HARMLESSLY)
386D 015100                 LD BC,KEYLN
3870 1828                   JR BUFMV2
3872
3872 CDC204     TOKMAIN:    CALL SETDE          ;GET DE=WKSP OR ELINE START
3875
3875            ;ENTRY FROM VAL FUNCTION
3875
3875 D9         TOKDE:      EXX
3876 2133C5                 LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN+INPLN+DKLN+RDLN+DFNLN
3879 11804D                 LD DE,CDBUFF+80H    ; (4D80-4E24)
387C 01A100                 LD BC,TOKLN
387F 1819                   JR BUFMV2
3881
3881            ;GET X OR GET X$
3881
3881 2100C2     GET:        LD HL,0C000H+RENLN
3884 DD                     DB 0DDH             ;'JR+3'
3885            ;POP <X>
3885
3885            ;DELETE N TO M - DELETE PROGRAM LINES
3885
3885 2130C2     DELETE:     LD HL,0C000H+RENLN+GETLN
3888 DD                     DB 0DDH             ;'JR+3'
3889
3889 21E3C2     POP:        LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN
388C DD                     DB 0DDH             ;'JR+3'
388D
388D 216EC4     DEFKEY:     LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN+INPLN
3890 DD                     DB 0DDH             ;'JR+3'
3891
3891 21EFC4     DEFFN:      LD HL,0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN+INPLN+DKLN+RDLN
3894 019100                 LD BC,91H           ;(DEF KEY LEN =90H)
3897
3897 11004F     BUFMV:      LD DE,INSTBUF
389A
389A D5         BUFMV2:     PUSH DE
389B F5                     PUSH AF
389C 3E5F                   LD A,5FH
389E D3FA                   OUT (250),A         ;ROM1 ON
38A0 EDB0                   LDIR
38A2 3E1F                   LD A,1FH
38A4 D3FA                   OUT (250),A         ;ROM1 OFF
38A6 F1                     POP AF
38A7 C9                     RET                 ;TO INSTBUF
38A8
38A8 3EFF       DEFAULT:    LD A,0FFH
```

```
38AA FE                     DB 0FEH           ;'JR+1'
38AB
38AB AF          LET:       XOR A
38AC
38AC 32755B                 LD (LTDFF),A
38AF FE                     DB 0FEH           ;'JR+1'
38B0
38B0 E7          LETLP:     RST 20H
38B1
38B1 CDF72C                 CALL SYNTAX1      ;CHECK VALID VAR
38B4 DF                     RST 18H
38B5 FE3D                   CP "="
38B7 C2290D                 JP NZ,NONSENSE
38BA
38BA E7                     RST 20H
38BB 3A755B                 LD A,(LTDFF)
38BE 3C                     INC A
38BF 2019                   JR NZ,LET3        ;JR IF 'LET', CONTINUE WITH 'DEFAULT'
38C1
38C1 CDC63A                 CALL RUNFLG       ;EVEN IF 'DEFAULT',
38C4 3014                   JR NC,LET3        ;ALWAYS CHECK ASSIGNED VALUE IF SYNTAX TIME
38C6
38C6 3A715C                 LD A,(FLAGX)
38C9 1F                     RRA
38CA 380E                   JR C,LET3         ;JR IF VAR DOESN'T EXIST - CREATE IT, LIKE 'LET'
38CC
38CC             ;          LD HL,FLAGS
38CC             ;          BIT 6,(HL)
38CC             ;          JR Z,LET2         ;JR IF STRING
38CC
38CC 3AB15A                 LD A,(DFTFB)
38CF A7                     AND A
38D0 2808                   JR Z,LET3
38D2
38D2 CD083B      LET2:      CALL EXPTEXPR     ;EVAL ASSIGNED VALUE IN ORDER TO SKIP IT. (SLOW,
38D5                                          ;BUT SAVES USE OF SPECIAL SKIPEXPR ROUTINE)
38D5 CD121D                 CALL FDELETE
38D8 1803                   JR LET4
38DA
38DA CDFA2A      LET3:      CALL VALFET1
38DD
38DD DF          LET4:      RST 18H
38DE FE2C                   CP ","
38E0 28CE                   JR Z,LETLP
38E2
38E2 C9                     RET
38E3
38E3             ;E.G. LABEL heaven:
38E3
38E3 CDC63A      LABEL:     CALL RUNFLG
38E6 DAF210                 JP C,SKIPCSTAT    ;JUST SKIP STATEMENT IF RUNNING
38E9
38E9 FE                     DB 0FEH           ;'JR+1'
38EA
38EA             ;SKIP, CHECK FOR VALID NUMERIC VARIABLE
38EA             ;USED BY COMPILE, LABEL
38EA
38EA E7          SVNUMV:    RST 20H
38EB
38EB             ;FROM POP, LABEL
38EB
38EB CDF72C      VNUMV:     CALL SYNTAX1      ;VALID VAR
38EE 213B5C                 LD HL,FLAGS
38F1 CB76                   BIT 6,(HL)
38F3 C0                     RET NZ            ;RET IF NUMERIC
38F4
38F4 CF                     RST 08H
38F5 1D                     DB 29             ;'SYNTAX ERROR'
38F6
38F6
38F6             ;TCLR.SAM - CLEAR, RUN
38F6
38F6 CD503A      RUN:       CALL SYNTAX3
38F9 CD9219                 CALL GOTO2        ;SET NEWPPC AND NSPPC FOR JUMP
38FC CDA234                 CALL RESTOREZ     ;DO RESTORE 0. SWITCHES IN PROGP
38FF 180E                   JR CLR1           ;NOW DO CLEAR 0
3901
3901 CD503A      CLEAR:     CALL SYNTAX3      ;NUMBER OR ZERO
3904 CD8C3F                 CALL UNSTLEN
3907 4F                     LD C,A
3908 0D                     DEC C
3909 B4                     OR H
390A B5                     OR L
390B CBFC                   SET 7,H
390D 2007                   JR NZ,CLR3        ;JR IF A CLEAR PARAM USED
390F
390F 3AB15C      CLR1:      LD A,(RAMTOPP)
3912 2AB25C                 LD HL,(RAMTOP)
3915 4F                     LD C,A
3916
3916 C5          CLR3:      PUSH BC
3917 E5                     PUSH HL
3918 CD1F1F                 CALL ADDRNV
391B EB                     EX DE,HL
391C 4F                     LD C,A            ;CDE=NVARS
391D 2A945A                 LD HL,(ELINE)
3920 3A935A                 LD A,(ELINEP)
3923 CDE71F                 CALL SUBAHLCDE    ;GET ELINE-NVARS IN AHL (AT LEAST 025DH)
3926 015D02                 LD BC,025DH
3929 CDD41F                 CALL SUBAHLBC     ;AHL=SPACE TO RECLAIM
392C 44                     LD B,H
392D 4D                     LD C,L
392E 2A885A                 LD HL,(NVARS)
```

```
3931 CD531E               CALL RECL2BIG     ;RECLAIM ABC AT HL
3934 CD6B39               CALL CLRSR
3937 CDDB33               CALL DOCOMP       ;COMPILE
393A CD9806               CALL MCLS         ;CLEAR ENTIRE SCREEN
393D 2A8E5A               LD HL,(WKEND)
3940 3A8D5A               LD A,(WKENDP)
3943 01B400               LD BC,180         ;**
3946 CDCC1F               CALL ADDAHLBC     ;AHL=WKEND+180
3949 D1                   POP DE
394A C1                   POP BC            ;CDE=CLEAR PARAM, OR RAMTOP
394B CDE71F               CALL SUBAHLCDE
394E 3006                 JR NC,RTERR       ;JR IF RAMTOP WILL BE TOO CLOSE TO WKEND
3950
3950 3AB05C               LD A,(LASTPAGE)
3953 B9                   CP C
3954 3002                 JR NC,CLR4        ;OK IF RAMTOP PAGE <= LAST ALLOCATED PAGE
3956
3956 CF          RTERR:   RST 08H
3957 30                   DB 48             ;'Invalid CLEAR address'
3958
3958             ;CLEAR MACHINE STACK (SHOULDN'T NEED IT!)
3958
3958 79          CLR4:    LD A,C
3959 32B15C               LD (RAMTOPP),A
395C ED53B25C             LD (RAMTOP),DE
3960 E1                   POP HL            ;NEXT STAT
3961 C1                   POP BC            ;ERR HANDLER
3962 31004F               LD SP,ISPVAL
3965 C5                   PUSH BC
3966 ED733D5C             LD (ERRSP),SP
396A E9                   JP (HL)
396B
396B             ;CLEAR FPCS, BASIC STACK, NUMERIC AND STRING VARS, TURNS 'RECORD' OFF
396B             ;SOUND CHIP OFF, ON ERROR OFF
396B
396B CDD439      CLRSR:   CALL CLSND        ;CLEAR SOUND CHIP, A=0
396E 32815B               LD (GRARF),A      ;GRAPHICS RECORDING OFF
3971 32455B               LD (ONERRFLG),A   ;ON ERROR OFF
3974 2AC65B               LD HL,(BASSTK)
3977 36FF                 LD (HL),0FFH
3979 22C45B               LD (BSTKEND),HL   ;CLEAR DO/GOSUB/ETC STACK
397C CD1F1F               CALL ADDRNV
397F 062E                 LD B,46
3981
3981 36FF        CLNVP:   LD (HL),0FFH
3983 23                   INC HL
3984 10FB                 DJNZ CLNVP        ;INIT 23 LETTER PTRS
3986
3986 EB                   EX DE,HL
3987 21E339               LD HL,PSVTAB
398A 0E1A                 LD C,26
398C EDB0                 LDIR              ;COPY 3 PTRS AND YOS/YRG
398E 21E939               LD HL,PSVT2
3991 0E14                 LD C,20
3993 EDB0                 LDIR              ;COPY YOS/YRG AGAIN
3995 EB                   EX DE,HL
3996 CDB639               CALL SETNE
3999 24                   INC H
399A 24                   INC H
399B CDB039               CALL SETSAV
399E 36FF                 LD (HL),0FFH      ;FF TERMINATOR OF SAVARS
39A0 25                   DEC H
39A1 25                   DEC H
39A2 2B                   DEC HL
39A3 2B                   DEC HL
39A4 2B                   DEC HL
39A5 70                   LD (HL),B         ;CHANGE 192 YRG TO 0
39A6 23                   INC HL
39A7 3601                 LD (HL),1         ;NOW 256
39A9 3A4D5A               LD A,(THFATT)
39AC A7                   AND A
39AD C0                   RET NZ
39AE
39AE 34                   INC (HL)          ;IF MODE 2 THIN PIX, MAKE XRG=512
39AF C9                   RET
39B0
39B0 FD21815A    SETSAV:  LD IY,SAVARSP
39B4 1804                 JR SETSYS
39B6
39B6 FD21845A    SETNE:   LD IY,NUMENDP
39BA
39BA             ;SET SYS VAR TO CURRENT PAGE AND ADDR IN HL. ADJUST IF NEEDED TO 8000-BFFF, BUT
39BA             ;KEEP HL THE SAME. ONLY F AND IY ALTERED
39BA
39BA F5          SETSYS:  PUSH AF
39BB DBFB                 IN A,(251)
39BD E61F                 AND 1FH
39BF FD7700               LD (IY+0),A
39C2 FD7501               LD (IY+1),L
39C5 FD7402               LD (IY+2),H
39C8 F1                   POP AF
39C9 CB74                 BIT 6,H
39CB C8                   RET Z
39CC
39CC FDCB02B6             RES 6,(IY+2)
39D0 FD3400               INC (IY+0)
39D3 C9                   RET
39D4
39D4
39D4
39D4 3E20        CLSND:   LD A,32           ;CLEAR 32 REGISTERS
39D6
39D6 01FF01      CSRL:    LD BC,SNDPORT+0100H ;SOUND REG ADDR
```

```
39D9 3D                     DEC A
39DA ED79                   OUT (C),A
39DC 05                     DEC B               ;BC=DATA PORT
39DD ED41                   OUT (C),B
39DF A7                     AND A
39E0 20F4                   JR NZ,CSRL
39E2
39E2 C9                     RET
39E3
39E3 1900         PSVTAB:   DW 0019H            ;X VARS
39E5 0300                   DW 0003H            ;Y VARS
39E7 FFFF                   DW 0FFFFH           ;Z VARS
39E9
39E9 02           PSVT2:    DB 2
39EA 0800                   DW 8
39EC 6F73                   DB "os"
39EE 0000000000             DB 0,0,0,0,0        ;YOS
39F3
39F3 02                     DB 2
39F4 FFFF                   DW 0FFFFH
39F6 7267                   DB "rg"
39F8 0000C00000             DB 0,0,192,0,0      ;YRG
39FD
39FD
39FD              ;STREAM 16 IS CONVERTED TO STREAM -4 INTERNALLY
39FD              ;CLOSE #16 AND OPEN #16 NOT ALLOWED
39FD              ;E.G. RECORD TO A$: LET A$="": PRINT #16;"TESTING" ADDS "TESTING (CR)" TO A$
39FD              ;O/P TO STREAM 16 LOOKS FOR STRING NAME IN (STRM16NM), AND 'PRINTS' CHARS
39FD              ;TO THE END OF THE STRING.
39FD              ;ALLOWS 'SERIAL FILES', TOKEN EXPANSION, CAT TO A STRING, RECORDING OF GRAPHICS
39FD              ;COMMANDS, ETC. CALLED WITH ROM1 OFF
39FD
39FD
39FD 47           S16OSR:   LD B,A              ;CHAR TO O/P
39FE DBFB                   IN A,(URPORT)       ;SAVE URPORT STATUS AND RESTORE AT END
3A00 F5                     PUSH AF
3A01 C5                     PUSH BC
3A02 21765B                 LD HL,STRM16NM      ;STORED TLBYTE AND NAME OF STRING
3A05 113F51                 LD DE,TLBYTE
3A08 010B00                 LD BC,11
3A0B 7E                     LD A,(HL)
3A0C EDB0                   LDIR
3A0E 4F                     LD C,A
3A0F CD3114                 CALL STARYLK2
3A12 CA082D                 JP Z,VNFERR
3A15
3A15 C1                     POP BC              ;B=DATA
3A16 DBFB                   IN A,(URPORT)
3A18 F5                     PUSH AF
3A19 E5                     PUSH HL             ;SAVE ADDR OF $ LEN DATA
3A1A C5                     PUSH BC
3A1B 7E                     LD A,(HL)
3A1C 23                     INC HL
3A1D 4E                     LD C,(HL)
3A1E 23                     INC HL
3A1F 0F                     RRCA
3A20 0F                     RRCA
3A21 B6                     OR (HL)
3A22 47                     LD B,A              ;BC=LEN
3A23 03                     INC BC              ;NEW LEN
3A24 78                     LD A,B
3A25 3C                     INC A
3A26 2002                   JR NZ,S16OK         ;OK UNLESS STRING LEN >FEFF
3A28
3A28 CF           STLERR:   RST 08H
3A29 2A                     DB 42               ;'String too long'
3A2A
3A2A C5           S16OK:    PUSH BC
3A2B 09                     ADD HL,BC           ;HL PTS TO LAST BYTE
3A2C DC9D1F                 CALL C,PGOVERF
3A2F CDEF3F                 CALL CHKHL
3A32 CD181E                 CALL MKRM1          ;OPEN 1 SPACE AT END OF STRING
3A35 C1                     POP BC              ;NEW LEN
3A36 F1                     POP AF              ;A=DATA
3A37 77                     LD (HL),A           ;ADD CHAR TO END
3A38 D1                     POP DE
3A39 F1                     POP AF
3A3A D3FB                   OUT (URPORT),A      ;PAGE IN STRING HEADER AT DE
3A3C CD433A                 CALL MBC
3A3F F1                     POP AF
3A40 D3FB                   OUT (URPORT),A
3A42 C9                     RET
3A43
3A43 CD791F       MBC:      CALL SPLITBC
3A46 21835B                 LD HL,PAGCOUNT
3A49 010300                 LD BC,3
3A4C EDB0                   LDIR                ;COPY PAGE/LEN TO STRING
3A4E C9                     RET
3A4F
3A4F E7           SSYNTAX3: RST 20H
3A50
3A50              ;SYNTAX 3 - NUMBER, OR USE 0
3A50
3A50 DF           SYNTAX3:  RST 18H
3A51 CDF73A                 CALL FETCHNUM
3A54 D8                     RET C               ;RET IF RUNNING
3A55
3A55 F1                     POP AF              ;JUNK RET ADDR
3A56 C9                     RET
3A57
3A57 E7           SSYNTAX6: RST 20H
3A58
3A58              ;SYNTAX 6 - INSIST ON NUMBER
```

```
3A58
3A58 CDE43A     SYNTAX6:   CALL EXPT1NUM
3A5B D8                    RET C
3A5C
3A5C F1                    POP AF
3A5D C9                    RET
3A5E
3A5E E7         SSYNTAX8:  RST 20H
3A5F
3A5F            ;SYNTAX 8 - INSIST ON NUMBER,NUMBER
3A5F
3A5F CDDC3A     SYNTAX8:   CALL EXPT2NUMS
3A62 D8                    RET C
3A63
3A63 F1                    POP AF
3A64 C9                    RET
3A65
3A65 E7         SSYNTAXA:  RST 20H
3A66
3A66            ;SYNTAX A - INSIST ON A STRING
3A66
3A66 CDA13A     SYNTAXA:   CALL EXPTSTR
3A69 D8                    RET C
3A6A
3A6A F1                    POP AF
3A6B C9                    RET
3A6C
3A6C
3A6C            ;COMMA/SEMICOLON RETURN Z
3A6C
3A6C FE2C       COMMASC:   CP ","
3A6E C8                    RET Z
3A6F
3A6F FE3B                  CP ";"
3A71 C9                    RET
3A72
3A72
3A72 FE8E       COCRCOTO:  CP TOTOK
3A74 C8                    RET Z
3A75
3A75            ;COMMA/CR/COLON RETURN Z
3A75
3A75 FE2C       COMCRCO:   CP ","
3A77 C8                    RET Z
3A78
3A78 FE                    DB 0FEH          ;'JR+1'
3A79
3A79 DF         RCRC:      RST 18H
3A7A
3A7A            ;CR/COLON RETURN Z
3A7A
3A7A FE3A       CRCOLON:   CP ":"
3A7C C8                    RET Z
3A7D
3A7D FE0D                  CP 0DH
3A7F C9                    RET
3A80
3A80 DF         RICSC:     RST 18H
3A81
3A81            ;INSIST ON A COMMA OR SEMI-COLON, SKIP
3A81
3A81 FE3B       INSISCSC:     CP ";"
3A83 2804                     JR Z,INSCOMN
3A85
3A85 FE2C       INSISCOMA:    CP ","
3A87 201E                     JR NZ,SYNONS
3A89
3A89 E7         INSCOMN:      RST 20H
3A8A C9                       RET
3A8B
3A8B            ;SKIP, INSIST ON OPENING BRACKET
3A8B
3A8B E7         SINSISOBRK:   RST 20H
3A8C
3A8C            ;INSIST ON AN OPENING BRACKET
3A8C
3A8C FE28       INSISOBRK:    CP "("
3A8E 2017                     JR NZ,SYNONS
3A90
3A90 E7                       RST 20H
3A91 C9                       RET
3A92
3A92            ;EXPECT NUMBER, THEN CLOSING BRACKET
3A92
3A92 CDE43A     EX1NUMCB:     CALL EXPT1NUM
3A95
3A95            ;INSIST ON A CLOSING BRACKET
3A95
3A95 FE29       INSISCBRK:    CP ")"
3A97 200E                     JR NZ,SYNONS
3A99
3A99 E7                       RST 20H
3A9A C9                       RET
3A9B
3A9B            ;EXPECT COMMA, THEN STRING
3A9B
3A9B DF         EXPTCSTR:     RST 18H
3A9C FE2C                     CP ","
3A9E 2007                     JR NZ,SYNONS
3AA0
3AA0            ;SKIP, EXPECT STRING
3AA0
3AA0 E7         SEXPTSTR:     RST 20H
```

```
3AA1
3AA1                ;EXPECT STRING
3AA1
3AA1 CDE314     EXPTSTR:        CALL SCANNING
3AA4 87                         ADD A,A             ;CY IF RUNNING
3AA5 79                         LD A,C              ;CURRENT CHAR
3AA6 F0                         RET P               ;RET IF STRING
3AA7
3AA7 CF         SYNONS:         RST 08H
3AA8 1D                         DB 29               ;'SYNTAX ERROR'
3AA9
3AA9                ;EXPECT COMMA, STRING, CLOSING BRACKET
3AA9
3AA9 CD9B3A     EXPTCSTRB:      CALL EXPTCSTR
3AAC 1815                       JR EXCBRF           ;CHECK ')', RUN FLAG
3AAE
3AAE                ;EXPECT '(N,N)'. SET CY IF RUNNING
3AAE
3AAE CD8B3A     EXB2NUMB:       CALL SINSISOBRK     ;SKIP, '('
3AB1 CDDC3A                     CALL EXPT2NUMS      ;N,N
3AB4 180D                       JR EXCBRF           ;CHECK ')', RUN FLAG
3AB6
3AB6                ;EXPECT Bracket, String, Comma, Number, Bracket. '(a$,n)'. CY IF RUNNING
3AB6
3AB6 CD8B3A     EXBSCNB:        CALL SINSISOBRK     ;'('
3AB9 CDA13A                     CALL EXPTSTR        ;A$
3ABC FE2C                       CP ","
3ABE 20E7                       JR NZ,SYNONS
3AC0
3AC0 CDE33A     SEX1NUMCB:      CALL SEXPT1NUM      ;SKIP, GET N. CY IF RUNNING
3AC3
3AC3 CD953A     EXCBRF:         CALL INSISCBRK      ;')' . CY IF RUNNING
3AC6
3AC6 3A3B5C     RUNFLG:         LD A,(FLAGS)
3AC9 87                         ADD A,A             ;C IF RUNNING, P IF STRING, M IF NUMBER
3ACA C9                         RET
3ACB
3ACB                ;EXPECT Bracket, Number, Comma, String, Bracket. '(n,a$)'. CY IF RUNNING
3ACB
3ACB CD8B3A     EXBNCSB:        CALL SINSISOBRK     ;'('
3ACE CDE43A                     CALL EXPT1NUM       ;N
3AD1 18D6                       JR EXPTCSTRB        ;',A$)'
3AD3
3AD3                ;SKIP, EXPECT N,N,N,N
3AD3
3AD3 E7         SEXPT4NUMS:     RST 20H
3AD4
3AD4 CDDC3A     EXPT4NUMS:      CALL EXPT2NUMS
3AD7 FE2C                       CP ","
3AD9 20CC                       JR NZ,SYNONS
3ADB
3ADB                ;SKIP, EXPECT N,N
3ADB
3ADB E7         SEXPT2NUMS:     RST 20H
3ADC
3ADC                ;EXPECT N,N
3ADC
3ADC CDE43A     EXPT2NUMS:      CALL EXPT1NUM
3ADF
3ADF                ;EXPECT ,N
3ADF
3ADF FE2C       EXPTCNUM:       CP ","
3AE1 20C4                       JR NZ,SYNONS
3AE3
3AE3
3AE3                ;SKIP, EXPECT NUMBER
3AE3
3AE3 E7         SEXPT1NUM:      RST 20H
3AE4
3AE4                ;EXPECT NUMBER
3AE4
3AE4 CDE314     EXPT1NUM:       CALL SCANNING
3AE7 87                         ADD A,A         ;CY IF RUNNING
3AE8 79                         LD A,C
3AE9 F8                         RET M           ;RET IF NUMERIC
3AEA
3AEA 18BB                       JR SYNONS
3AEC
3AEC                ;INSIST A=LETTER
3AEC
3AEC CD1E3B     GETALPH:    CALL ALPHA
3AEF D8                     RET C
3AF0
3AF0 18B5                   JR SYNONS
3AF2
3AF2                ;SYNTAX9 - DEAL WITH COLOUR ITEMS, COORDS
3AF2                ; EG INK 3,PAPER 1;X,Y. USED BY PLOT, CIRCLE, FILL
3AF2
3AF2 CD4513     SYNTAX9:    CALL SYNT9SR
3AF5 18E5                   JR EXPT2NUMS
3AF7
3AF7                ;EXPECT N OR CR/COLON. RETURN WITH CY IF RUNNING
3AF7
3AF7 CD7A3A     FETCHNUM:   CALL CRCOLON
3AFA 20E8                   JR NZ,EXPT1NUM      ;EXITS WITH CY IF RUNNING
3AFC
3AFC 3A3B5C     CONDSTK0:   LD A,(FLAGS)
3AFF 17                     RLA
3B00 D0                     RET NC
3B01
3B01 AF                     XOR A
3B02 CDDA1C                 CALL STACKA
3B05 37                     SCF             ;'RUNNING'
```

```
3B06 C9                    RET
3B07
3B07 E7          SEXPTEXPR:    RST 20H
3B08
3B08             ;EXPECT AN EXPRESSION, SET Z IF STRING, NZ IF NUMERIC
3B08
3B08 CDE314      EXPTEXPR:     CALL SCANNING
3B0B 17                        RLA              ;CY IF RUNNING
3B0C CB7F                      BIT 7,A
3B0E 79                        LD A,C
3B0F C9                        RET
3B10
3B10             ;USED BY INPUT,..
3B10
3B10 CDD73F      CHKENDCP:  CALL SELCHADP
3B13 FE                     DB 0FEH           ;"JR+1"
3B14
3B14             ;SKIP, ABORT IF SYNTAX TIME. USED BY E.G. PI, HIMEM, FREE.
3B14
3B14 E7          SABORTER:  RST 20H
3B15
3B15             ;JUST RETURN IF RUNNING, ELSE JUNK A RET ADDR, RET TO NEXT LEVEL.
3B15
3B15             CHKEND:
3B15 4F          ABORTER:   LD C,A
3B16 3A3B5C                 LD A,(FLAGS)
3B19 17                     RLA
3B1A 79                     LD A,C
3B1B D8                     RET C
3B1C
3B1C F1                     POP AF
3B1D C9                     RET
3B1E
3B1E             ;CY IF A=LETTER, ELSE NC
3B1E
3B1E FE41        ALPHA:     CP "A"
3B20 3F                     CCF
3B21 D0                     RET NC            ;RET IF TOO LOW
3B22
3B22 FE7B                   CP "z"+1
3B24 D0                     RET NC            ;RET IF TOO HI
3B25
3B25 FE5B                   CP "Z"+1
3B27 D8                     RET C             ;RET IF UC
3B28
3B28 FE61                   CP "a"
3B2A 3F                     CCF
3B2B C9                     RET
3B2C
3B2C             ;CY IF A=LETTER OR DIGIT
3B2C
3B2C CD1E3B      ALPHANUM:  CALL ALPHA
3B2F D8                     RET C
3B30
3B30             ;CY IF A DIGIT
3B30
3B30 FE3A        NUMERIC:   CP "9"+1          ;NC IF TOO HIGH
3B32 D0                     RET NC
3B33
3B33 FE30                   CP "0"
3B35 3F                     CCF
3B36 C9                     RET
3B37
3B37
3B37             ;CY IF LETTER OR UNDERLINE OR '$'
3B37             ;(SO E.G. printer, print_out, print$ ARE NOT TOKENISED, BUT print1, print: ARE)
3B37
3B37 CD1E3B      ALDU:      CALL ALPHA
3B3A D8                     RET C
3B3B
3B3B FE24                   CP "$"
3B3D 37                     SCF
3B3E C8                     RET Z
3B3F
3B3F 1804                   JR CKUND
3B41
3B41
3B41             ;CY IF LETTER, DIGIT OR UNDERLINE
3B41
3B41 CD2C3B      ALNUMUND:  CALL ALPHANUM
3B44 D8                     RET C
3B45
3B45 FE5F        CKUND:     CP "_"
3B47 37                     SCF
3B48 C8                     RET Z
3B49
3B49 A7                     AND A
3B4A C9                     RET
3B4B
3B4B             ;EVAL. BRACKETLESS SLICER E.G. 10 TO 30, TO 100, 100 TO, TO
3B4B             ;ENTRY: CHAD PTS TO POSSIBLE SLICER, A=(CHAD)
3B4B             ;EXIT: SYS VARS SET UP, WITH VALUE, IF GIVEN, OR DEFAULT. CHAD PTS. TO NON-ALPHA
3B4B             ;NUMERIC CHAR, NOT 'TO'. IF RUNNING, CY=OUT OF RANGE. A=0 IF SLICER IS 1 NUMBER
3B4B
3B4B 210100      BRKLSSL:   LD HL,1           ;MIN
3B4E 11FFFE                 LD DE,0FEFFH      ;MAX (LINE NUMBERS)
3B51 227D5A                 LD (FIRST),HL
3B54 ED537F5A               LD (LAST),DE
3B58 FE8E                   CP TOTOK
3B5A 281A                   JR Z,BRL2
3B5C
3B5C CD2C3B                 CALL ALPHANUM
3B5F D0                     RET NC            ;NC='IN RANGE' IF EG LIST
```

```
3B60
3B60 CD8E3B                CALL GIR2          ;GET INT IN BC AND HL IF RUNNING, NC IF RUNNING
3B63 380B                  JR C,BRL1          ;JR IF NOT RUNNING
3B65
3B65 ED4B7D5A              LD BC,(FIRST)      ;MIN
3B69 ED42                  SBC HL,BC
3B6B 09                    ADD HL,BC
3B6C 227D5A                LD (FIRST),HL
3B6F D8                    RET C              ;RET IF MIN>VALUE
3B70
3B70 FE8E          BRL1:   CP TOTOK
3B72 3E00                  LD A,0             ;CANNOT USE XOR A!
3B74 2012                  JR NZ,BRL3         ;JR WITH A=0 (FLAG) IF SLICER = 1 NUMBER
3B76                                          ;LAST=FIRST
3B76
3B76 E7            BRL2:   RST 20H
3B77 CD2C3B                CALL ALPHANUM
3B7A D0                    RET NC             ;'IN RANGE' IF EG LIST 10 TO
3B7B
3B7B CD8E3B                CALL GIR2
3B7E 3F                    CCF
3B7F D0                    RET NC             ;END WITH NC IF SYNTAX TIME AND EG LIST 10 TO 20
3B80
3B80 2A7F5A                LD HL,(LAST)       ;MAX
3B83 ED42                  SBC HL,BC
3B85 D8                    RET C              ;RET IF VALUE>MAX
3B86
3B86 60                    LD H,B
3B87 69                    LD L,C
3B88
3B88 227F5A        BRL3:   LD (LAST),HL
3B8B A7                    AND A
3B8C C9                    RET
3B8D
3B8D               ;GET IF RUNNING - AUTO, DELETE, BRACKETLESS SLICER SR.
3B8D               ;ENTRY: CHAD PTS TO EXPR (GIR2) OR IS BEFORE EXPR (GIR).
3B8D               ;EXIT: CHAD PTS PAST EXPR, A=(CHAD), BC AND HL=INT (IF RUNNING)
3B8D               ;CY IF RUNNING
3B8D
3B8D E7            GIR:    RST 20H
3B8E
3B8E CDE43A        GIR2:   CALL EXPT1NUM
3B91 3F                    CCF
3B92 D8                    RET C              ;RET IF NOT RUNNING
3B93
3B93 CD2E1D                CALL GETINT
3B96 DF                    RST 18H
3B97 60                    LD H,B
3B98 69                    LD L,C
3B99 A7                    AND A
3B9A C9                    RET                ;INT IN BC AND HL, NC
3B9B
3B9B
3B9B
3B9B               ;DISPLAY (SCREEN)
3B9B
3B9B CD503A        DISPLAY:CALL SYNTAX3
3B9E
3B9E CD331D                CALL GETBYTE       ;IN A AND C
3BA1
3BA1 32775A        SETDISP:LD (CURDISP),A
3BA4 A7                    AND A
3BA5 2807                  JR Z,DEFDISP       ;'DISPLAY 0' MEANS DISPLAY CURRENT SCREEN
3BA7
3BA7 F7                    RST 30H
3BA8 82D2                  DW SCRNTLK2        ;GET MODE/PAGE FOR SCREEN C, Z IF UNUSED
3BAA 2005                  JR NZ,VIDSEL
3BAC
3BAC CF            ISCRERR:RST 08H
3BAD 2B                    DB 43              ;'INVALID SCREEN NUMBER' (ERRORS SET CURDISP TO 0)
3BAE
3BAE               ;DEFAULT DISPLAY (USED BY 'DISPLAY' AND REPORTS)
3BAE
3BAE 3A785A        DEFDISP:LD A,(CUSCRNP)     ;PAGE BEING USED FOR PRINT, PLOT, ETC.(CURRENT)
3BB1                                          ;CONTAINS MODE TOO.
3BB1
3BB1 F7            VIDSEL: RST 30H
3BB2 2CE5                  DW CUS2            ;SEE IF DISPLAYED=CURRENT
3BB4
3BB4 C8                    RET Z              ;RET IF NO CHANGE IN DISPLAYED SCREEN.
3BB5
3BB5 D5                    PUSH DE            ;NEW DISPLAY PAGE IN D
3BB6 A7                    AND A              ;NC
3BB7 CDBE3B                CALL SDISR         ;SWITCH PREV DISPLAYED SCREEN IN AT 8000H
3BBA                                          ;AND COPY WORKING PALTAB TO IT FOR LATER USE IF
3BBA                                          ;IT IS RE-DISPLAYED
3BBA F1                    POP AF
3BBB D3FC                  OUT (VIDPORT),A    ;MODE/PAGE SENT TO HARDWARE - NEW DISPLAY
3BBD 37                    SCF
3BBE
3BBE 08            SDISR:  EX AF,AF'
3BBF DBFB                  IN A,(251)
3BC1 F5                    PUSH AF
3BC2 08                    EX AF,AF'
3BC3 F5                    PUSH AF
3BC4 3C                    INC A              ;SO SECOND PAGE OF SCREEN AT 8000H
3BC5 CDDF3F                CALL TSURPG
3BC8 21D8BF                LD HL,PALBUF-4000H          ;END OF SECOND SCREEN PAGE
3BCB 11D855                LD DE,PALTAB
3BCE F1                    POP AF
3BCF 3801                  JR C,SDIS2
3BD1
```

```
3BD1 EB                    EX DE,HL
3BD2
3BD2 012800      SDIS2:    LD BC,28H
3BD5 EDB0                  LDIR              ;COPY PALETTE TO/FROM DISPLAYED SCREEN
3BD7                                         ;AND PALTAB IN SYS VARS
3BD7 C3453C                JP PPORT          ;POP AF, OUT 251,RET
3BDA
3BDA 32785A      PRSVARS:  LD (CUSCRNP),A
3BDD
3BDD             ;RESTORE SCREEN VARS FROM SCREEN PAGE
3BDD
3BDD 37          RSVARS:   SCF
3BDE 26                    DB 26H            ;'JR+1'
3BDF
3BDF             ;SAVE SCREEN VARS TO SCREEN PAGE
3BDF
3BDF A7          SSVARS:   AND A
3BE0
3BE0 21345A                LD HL,BGFLG
3BE3 11B0FE                LD DE,PVBUFF      ;POSN, INK ETC. ATTACHED TO SCREEN
3BE6 3001                  JR NC,SSVRC
3BE8
3BE8 EB                    EX DE,HL
3BE9
3BE9 CDA83F      SSVRC:    CALL SPSSR
3BEC 013C00                LD BC,PRPOSN-BGFLG
3BEF EDB0                  LDIR
3BF1 0E90                  LD C,CEXTAB-PRPOSN
3BF3 09                    ADD HL,BC
3BF4 EB                    EX DE,HL
3BF5 09                    ADD HL,BC
3BF6 EB                    EX DE,HL          ;SRC AND DEST ADVANCED TO COPY CEXTAB
3BF7 0E40                  LD C,40H
3BF9 EDB0                  LDIR
3BFB C3BF3F                JP RCURPR
3BFE
3BFE             ;ROM 0 INSTRING
3BFE
3BFE C5          R0INST:       PUSH BC              ;START POSN
3BFF 42                        LD B,D
3C00 4B                        LD C,E               ;BC=BYTES TO CHECK
3C01 11004F                    LD DE,INSTBUF        ;HL=START-OF-SEARCH PTR, (SP)=START POSN
3C04                                                ;DE=T$ START, A=T$ LEN
3C04 C5                        PUSH BC              ;BYTES TO CHECK
3C05
3C05 F5          LOOKLP:       PUSH AF              ;TARGET$ LEN
3C06 08                        EX   AF,AF'          ;A'=TARGET$ LEN COUNTER
3C07 1A                        LD   A,(DE)          ;GET FIRST TARGET$ CHARACTER
3C08 EDB1                      CPIR                 ;LOOK FOR IT IN SEARCH$ USING HL AS PTR
3C0A E2313C                    JP PO,NOTFND0        ;JR IF NOT FOUND - BC=0000
3C0D
3C0D E5                        PUSH HL              ;SEARCH$ PTR
3C0E 3E                        DB 3EH               ;'JR +1' - HL IS AT SECOND CHAR ALREADY
3C0F
3C0F 23          CHKNXTC:      INC HL
3C10
3C10 08                        EX   AF,AF'          ;GET TARGET$ LEN COUNTER
3C11 3D                        DEC  A
3C12 2812                      JR   Z,FOUND         ;FOUND IF ALL CHARS MATCHED
3C14
3C14 08                        EX   AF,AF'
3C15 13                        INC  DE
3C16 1A                        LD   A,(DE)          ;NEXT TARGET$ CHAR
3C17 BE                        CP   (HL)            ;CP NEXT SEARCH$ CHAR
3C18 28F5                      JR   Z,CHKNXTC
3C1A
3C1A D9                        EXX
3C1B B9                        CP C                 ;C' IS USUALLY '#'
3C1C D9                        EXX
3C1D
3C1D 28F0                      JR   Z,CHKNXTC       ;HASH ALWAYS MATCHES
3C1F
3C1F 11004F                    LD DE,INSTBUF        ;MATCH FAILS - RESTORE T$ PTR
3C22 E1                        POP  HL              ;S$ PTR
3C23 F1                        POP  AF              ;REFRESH TARGET$ LEN COUNTER
3C24 18DF                      JR   LOOKLP
3C26
3C26
3C26 E1          FOUND:        POP  HL              ;SEARCH$ PTR
3C27 E1                        POP  HL              ;T$ LEN
3C28 E1                        POP  HL              ;BYTES TO CHECK
3C29 A7                        AND A
3C2A ED42                      SBC  HL,BC           ;SBC BYTES TO CHECK, BYTES LEFT TO CHECK
3C2C                                                ;TO GET BYTES CHECKED
3C2C C1                        POP  BC              ;START POSN
3C2D 09                        ADD  HL,BC
3C2E 44                        LD   B,H
3C2F 4D                        LD   C,L
3C30 C9                        RET                  ;BC=TARGET$ POSN IN SEARCH$. NC
3C31
3C31
3C31             ;ENTRY IF 1ST CHAR NEVER FOUND - HL IS PAST LAST POSN LOOKED AT
3C31
3C31 F1          NOTFND0:      POP  AF              ;T$ LEN
3C32             ;               EX AF,AF'
3C32 F1                        POP  AF              ;JUNK BYTES TO CHECK
3C33
3C33 D1          NOTFND2:      POP  DE              ;START POSN
3C34
3C34 010000      NOTFND3:      LD   BC,0000H
3C37 37                        SCF
3C38 C9                        RET
3C39
```

```
3C39
3C39                                            ;LET/DEFAULT, LABEL, RUN, CLEAR, S16OP, SYNTAX
3C39                                            ;SRS, DISPLAY, SCREEN, SETSV
3C39                    INCLUDE ENDPRINT.SAM    ;END PRINT, ANY DE ADDR, PIXADDR, POATTR, NET
3C39            ;ENDPRINT.SAM - FINAL SCREEN OUTPUT ROUTINES FOR MODES 0-3 PRINTABLE CHARS.
3C39            ;ENTRY: DE=SCREEN ROW/COL, HL PTS TO CHAR DATA, B=OVER AND C=INVERSE MASK
3C39
3C39            ;CALLED FROM ROM1
3C39
3C39 CD493C     EPSUB:      CALL R1OSR
3C3C CDB63F                 CALL SELSCRN
3C3F CD7B3C                 CALL EPSSR
3C42
3C42 F1         POPOUT:     POP AF
3C43 D3FA                   OUT (250),A
3C45
3C45 F1         PPORT:      POP AF
3C46 D3FB                   OUT (251),A
3C48 C9                     RET
3C49
3C49 FDE1       R1OSR:      POP IY
3C4B DBFB                   IN A,(251)
3C4D F5                     PUSH AF
3C4E DBFA                   IN A,(250)
3C50 F5                     PUSH AF
3C51 E6BF                   AND 0BFH
3C53 D3FA                   OUT (250),A       ;ROM1 OFF
3C55 FDE9                   JP (IY)
3C57
3C57            ;USED BY RENUM
3C57
3C57 EB         GTRLNN:     EX DE,HL
3C58 23                     INC HL
3C59 DBFB                   IN A,(251)
3C5B F5                     PUSH AF
3C5C CD4D1A                 CALL FNDLINE
3C5F 56                     LD   D,(HL)
3C60 23                     INC  HL
3C61 5E                     LD   E,(HL)       ;GET LINE NUMBER
3C62 2A7D5A                 LD HL,(FIRST)
3C65 18DE                   JR PPORT
3C67
3C67            ;COPY STRING FROM COMMON MEMORY TO WORKSPACE, STACK PARAMS ON FPCS.
3C67            ;MOVE BC BYTES FROM (HL) TO WKSPACE. USED BY E.G. HEX$, CHR$, STR$.
3C67            ;FOR STRINGS LEN 1-16K. EXIT: DE=STKEND, PAGING UNCHANGED.
3C67
3C67 CD493C     CWKSTK:     CALL R1OSR
3C6A E5                     PUSH HL
3C6B CD801E                 CALL WKROOM       ;DE=DEST, BC UNCHANGED
3C6E E1                     POP HL
3C6F D5                     PUSH DE
3C70 C5                     PUSH BC
3C71 EDB0                   LDIR
3C73 C1                     POP BC
3C74 D1                     POP DE
3C75 CDE71C                 CALL STKSTOREP
3C78 EB                     EX DE,HL          ;DE=STKEND
3C79 18C7                   JR POPOUT
3C7B
3C7B
3C7B 3A365A     EPSSR:      LD A,(CSIZE)      ;HEIGHT **
3C7E E5                     PUSH HL
3C7F D9                     EXX
3C80 D1                     POP DE            ;DE'=DATA PTR IF MODE 3/4
3C81 FE08                   CP 8
3C83 3802                   JR C,HPL2
3C85
3C85 3E08                   LD A,8
3C87
3C87 47         HPL2:       LD B,A
3C88 D9                     EXX               ;** BUG FIX - LIMIT O/P SCANS TO 8
3C89 3A405A                 LD A,(MODE)
3C8C FE02                   CP 2
3C8E 284C                   JR Z,M2PRINT
3C90
3C90 D2513D                 JP NC,M3PRINT
3C93
3C93 3D                     DEC A
3C94 281F                   JR Z,M1PRINT
3C96
3C96            ;*****************************************************************************
3C96            ;MODE 0 PRINT. ENTRY: DE=SCREEN ROW/COL, HL=CHAR DATA, B=OVER AND C=INVERSE MASK
3C96
3C96 CDE43D     M0PRINT:    CALL M0DEADDR     ;GET DE=SCREEN ADDR
3C99 D5                     PUSH DE
3C9A D9                     EXX
3C9B
3C9B D9         M0PRLP:     EXX               ;SCAN COUNTER IN B'
3C9C 1A                     LD A,(DE)         ;SCREEN DATA
3C9D A0                     AND B             ;OVER MASK
3C9E AE                     XOR (HL)          ;XOR SCRN
3C9F A9                     XOR C             ;XOR INVERSE MASK
3CA0 12                     LD (DE),A
3CA1 23                     INC HL
3CA2 14                     INC D
3CA3 7A                     LD A,D
3CA4 E607                   AND 07H
3CA6 2005                   JR NZ,M0PRNT2      ;JR IF NOT CROSSING CHAR BOUNDARY
3CA8
3CA8 EB                     EX DE,HL
3CA9 CD090D                 CALL NXTDOWN1
3CAC EB                     EX DE,HL
3CAD
```

```
3CAD D9          M0PRNT2:   EXX
3CAE 10EB                   DJNZ M0PRLP
3CB0
3CB0 E1                     POP HL
3CB1 BF                     CP A
3CB2 C3733E                 JP POATTR0
3CB5
3CB5             ;*******************************************************************************
3CB5             ;MODE 1 PRINT. ENTRY: DE=SCREEN ROW/COL, HL=CHAR DATA, B=OVER AND C=INVERSE MASK
3CB5
3CB5 CDD03D      M1PRINT:   CALL M1DEADDR     ;GET ADDR OF ROW/COL DE IN DE
3CB8 D9                     EXX
3CB9
3CB9 D9          M1PRLP:    EXX
3CBA 1A                     LD A,(DE)         ;FETCH SCREEN DATA
3CBB A0                     AND B             ;AND 0 IF OVER 0, AND FF IF OVER 1 OR 2
3CBC AE                     XOR (HL)          ;XOR CHAR DATA
3CBD A9                     XOR C             ;INVERSE MASK USED
3CBE 12                     LD (DE),A         ;PLACE ON SCREEN
3CBF 23                     INC HL            ;NEXT CHAR DATUM
3CC0 7B                     LD A,E
3CC1 C620                   ADD A,32
3CC3 5F                     LD E,A            ;DROP DE BY 1 SCAN.
3CC4
3CC4 3001                   JR NC,M1PRNC
3CC6
3CC6 14                     INC D
3CC7
3CC7 D9          M1PRNC:    EXX
3CC8 10EF                   DJNZ M1PRLP       ;LOOP FOR 8 SCANS
3CCA
3CCA D9                     EXX
3CCB 21001F                 LD HL,1F00H       ;DISP TO ATTR AREA (TOP ROW)
3CCE 19                     ADD HL,DE
3CCF CD773E                 CALL SETATTR      ;USE ATTRT TO CHANGE ATTR AT (HL). HL PRESERVED, A=
3CD2                                          ;NEW ATTR
3CD2 0607                   LD B,7            ;7 MORE TO CHANGE
3CD4 112000                 LD DE,32
3CD7
3CD7 19          M1PRATTR:  ADD HL,DE         ;DROP TO ATTR FOR NEXT CHAR ROW
3CD8 77                     LD (HL),A
3CD9 10FC                   DJNZ M1PRATTR
3CDB
3CDB C9                     RET
3CDC
3CDC             ;*******************************************************************************
3CDC             ;MODE 2 PRINT. ENTRY: DE=SCREEN ROW/COL, HL=CHAR DATA, B=OVER AND C=INVERSE MASK
3CDC
3CDC CDF53D      M2PRINT:   CALL M2DEADDR     ;CY SET IF 6-PIX CHARS, Z/NZ=EVEN/ODD
3CDF 265B                   LD H,CEXTAB/256   ;H IS EXPANSION TABLE MSB
3CE1 79                     LD A,C
3CE2 D9                     EXX
3CE3 4F                     LD C,A            ;C"=INVERSE MASK
3CE4 3813                   JR C,PR80COL
3CE6
3CE6 212151                 LD HL,MEMVAL      ;IF 64-COL, MUST ROTATE CHAR DATA FOR CORRECT
3CE9                                          ;ALIGNMENT WHEN USING 85-COL O/P ROUTINE
3CE9 C5                     PUSH BC           ;**
3CEA
3CEA 1A          P64AL:     LD A,(DE)
3CEB 0F                     RRCA
3CEC 77                     LD (HL),A
3CED 23                     INC HL
3CEE 13                     INC DE
3CEF 10F9                   DJNZ P64AL
3CF1
3CF1 C1                     POP BC            ;**
3CF2 112151                 LD DE,MEMVAL      ;NEW CHAR LOCN.
3CF5 D9                     EXX
3CF6 48                     LD C,B            ;MAKE RHS OVER MASK=LHS OVER MASK
3CF7 1807                   JR PR80EVEN
3CF9
3CF9             ;85-COLUMN PRINT
3CF9             ;ENTRY: NZ IF ODD COLUMN, Z IF EVEN COLUMN.
3CF9
3CF9 D9          PR80COL:   EXX
3CFA 78                     LD A,B            ;OVER MASK
3CFB 202A                   JR NZ,PR80ODD
3CFD
3CFD F60F                   OR 0FH
3CFF 4F                     LD C,A            ;RHS OVER MASK FOR EVEN COLS=0F OR FF
3D00
3D00 D9          PR80EVEN:  EXX               ;LHS OVER MASK FOR EVEN COLS=00 OR FF
3D01
3D01 1A          M2PREVLP:  LD A,(DE)         ;GET CHAR DATA FROM (DE")
3D02 A9                     XOR C             ;INVERSE MASK
3D03 13                     INC DE
3D04 D9                     EXX
3D05
3D05 F5                     PUSH AF
3D06 0F                     RRCA              ;01234560->00123456
3D07 0F                     RRCA              ;         ->60012345
3D08 0F                     RRCA              ;         ->56001234
3D09 E60F                   AND 0FH           ;GET VALUE OF HIGHER CHAR NIBBLE
3D0B 6F                     LD L,A            ;HL PTS TO ENTRY IN 16-BYTE TABLE CONTAINING
3D0C                                          ;COLOURED EXPANDED DATA
3D0C 1A                     LD A,(DE)         ;SCREEN DATA
3D0D AE                     XOR (HL)
3D0E A0                     AND B             ;LHS OVER MASK
3D0F AE                     XOR (HL)          ;EXPANDED CHAR DATA
3D10 12                     LD (DE),A
3D11
3D11 1C                     INC E             ;NEXT SCREEN COLUMN
```

```
3D12 F1                    POP AF             ;01234560
3D13 07                    RLCA               ;12345600
3D14 E60F                  AND 0FH            ;GET VALUE OF LOWER CHAR NIBBLE.
3D16 6F                    LD L,A             ;HL PTS TO ENTRY IN 16-BYTE TABLE CONTAINING
3D17                                          ;COLOURED EXPANDED DATA
3D17 1A                    LD A,(DE)
3D18 AE                    XOR (HL)
3D19 A1                    AND C              ;RHS OVER MASK
3D1A AE                    XOR (HL)
3D1B 12                    LD (DE),A
3D1C 7B                    LD A,E
3D1D C67F                  ADD A,127
3D1F 5F                    LD E,A             ;DROP DE TO NEXT SCAN, BACK UP 1 BYTE
3D20 3001                  JR NC,M2PRNCE
3D22
3D22 14                    INC D
3D23
3D23 D9          M2PRNCE:  EXX
3D24 10DB                  DJNZ M2PREVLP
3D26
3D26 C9                    RET
3D27
3D27 F6F0        PR80ODD:  OR 0F0H
3D29 4F                    LD C,A             ;LHS OVER MASK FOR ODD COLS=F0 OR FF
3D2A D9                    EXX                ;RHS OVER MASK FOR ODD COLS=00 OR FF
3D2B
3D2B 1A          M2PRODLP: LD A,(DE)          ;GET CHAR DATA; 01234560
3D2C A9                    XOR C              ;INVERSE MASK
3D2D 13                    INC DE
3D2E D9                    EXX
3D2F
3D2F F5                    PUSH AF
3D30 07                    RLCA               ;12345600
3D31 07                    RLCA               ;23456001
3D32 07                    RLCA               ;34560012
3D33 E60F                  AND 0FH            ;GET VALUE OF HIGHER CHAR NIBBLE.
3D35 6F                    LD L,A             ;HL PTS TO ENTRY IN 16-BYTE TABLE CONTAINING
3D36                                          ;COLOURED EXPANDED DATA
3D36 1A                    LD A,(DE)          ;SCREEN DATA
3D37 AE                    XOR (HL)
3D38 A1                    AND C              ;LHS OVER MASK
3D39 AE                    XOR (HL)           ;EXPANDED CHAR DATA
3D3A 12                    LD (DE),A
3D3B
3D3B 1C                    INC E              ;NEXT SCREEN COLUMN
3D3C F1                    POP AF             ;GET ORIG CHAR DATUM AGAIN
3D3D 0F                    RRCA               ;00123456
3D3E E60F                  AND 0FH            ;GET VALUE OF LOWER CHAR NIBBLE.
3D40 6F                    LD L,A             ;HL PTS TO ENTRY IN 16-BYTE TABLE CONTAINING
3D41                                          ;COLOURED EXPANDED DATA
3D41 1A                    LD A,(DE)
3D42 AE                    XOR (HL)
3D43 A0                    AND B              ;RHS OVER MASK
3D44 AE                    XOR (HL)
3D45 12                    LD (DE),A
3D46 7B                    LD A,E
3D47 C67F                  ADD A,127
3D49 5F                    LD E,A             ;DROP DE TO NEXT SCAN, BACK UP 1 BYTE
3D4A 3001                  JR NC,M2PRNCO
3D4C
3D4C 14                    INC D
3D4D
3D4D D9          M2PRNCO:  EXX
3D4E 10DB                  DJNZ M2PRODLP
3D50
3D50 C9                    RET
3D51
3D51             ;*************************************************************************
3D51             ;MODE 3 PRINT. ENTRY: DE=SCREEN ROW/COL, HL=CHAR DATA, B=OVER AND C=INVERSE MASK
3D51             ;USES HL,DE,BC',DE",BC"
3D51             ;TAKES ABOUT 294*8 TS vs. POSSIBLE 218*8 FOR OVER 0 ONLY ROUTINE USING LDI
3D51
3D51 CDC33D      M3PRINT:  CALL M3DEADDR
3D54 265B                  LD H,CEXTAB/256    ;H IS EXPANSION TABLE MSB
3D56 79                    LD A,C
3D57
3D57 D9                    EXX
3D58 4F                    LD C,A             ;C"=INVERSE MASK
3D59
3D59 1A          M3PRLP:   LD A,(DE)          ;GET CHAR DATA
3D5A A9                    XOR C              ;INVERSE MASK
3D5B 13                    INC DE
3D5C D9                    EXX
3D5D
3D5D 4F                    LD C,A
3D5E 1F                    RRA
3D5F 1F                    RRA
3D60 1F                    RRA
3D61 E61E                  AND 1EH            ;GET VALUE OF HIGHER CHAR NIBBLE,*2
3D63 6F                    LD L,A             ;HL PTS TO ENTRY IN 16-WORD TABLE CONTAINING
3D64                                          ;COLOURED EXPANDED DATA
3D64 1A                    LD A,(DE)          ;SCREEN DATA
3D65 A0                    AND B              ;OVER MASK
3D66 AE                    XOR (HL)           ;EXPANDED CHAR DATA
3D67 12                    LD (DE),A
3D68 2C                    INC L              ;NEXT EXPANDED DATUM
3D69 1C                    INC E              ;NEXT SCREEN COLUMN
3D6A
3D6A 1A                    LD A,(DE)
3D6B A0                    AND B              ;OVER MASK
3D6C AE                    XOR (HL)
3D6D 12                    LD (DE),A
3D6E 1C                    INC E              ;NEXT SCREEN COLUMN
```

```
3D6F
3D6F 79                     LD A,C            ;GET ORIG CHAR DATUM AGAIN
3D70
3D70 17                     RLA
3D71 E61E                   AND 1EH           ;GET VALUE OF LOWER CHAR NIBBLE *2
3D73 6F                     LD L,A            ;HL PTS TO ENTRY IN 16-WORD TABLE CONTAINING
3D74                                          ;COLOURED EXPANDED DATA
3D74 1A                     LD A,(DE)
3D75 A0                     AND B             ;OVER MASK
3D76 AE                     XOR (HL)
3D77 12                     LD (DE),A
3D78 2C                     INC L             ;NEXT EXPANDED DATUM
3D79 1C                     INC E             ;NEXT SCREEN COLUMN
3D7A
3D7A 1A                     LD A,(DE)
3D7B A0                     AND B             ;OVER MASK
3D7C AE                     XOR (HL)
3D7D 12                     LD (DE),A
3D7E 7B                     LD A,E
3D7F C67D                   ADD A,125
3D81 5F                     LD E,A            ;DROP DE TO NEXT SCAN, BACK UP 3 BYTES
3D82 3001                   JR NC,M3PRNC
3D84
3D84 14                     INC D
3D85
3D85 D9           M3PRNC:   EXX
3D86 10D1                   DJNZ M3PRLP
3D88
3D88 C9                     RET
3D89
3D89
3D89              ;POFETCH -  GET D=ROW, E=COL, A=RHS LIMIT, CY IF PRINTER OR OTHER
3D89
3D89 3A735A       POFETCH:  LD A,(DEVICE)     ;0=UPPER SCREEN, 1=LOWER, 2=PRINTER OR OTHER
3D8C A7                     AND A             ;NC, TEST FOR ZERO
3D8D 2810                   JR Z,POF2         ;JR IF UPPER SCREEN
3D8F
3D8F ED5B6E5A               LD DE,(SPOSNL)
3D93 3D                     DEC A
3D94 280D                   JR Z,POF3         ;JR IF LOWER SCREEN
3D96
3D96 ED5B705A               LD DE,(PRPOSN)
3D9A 3A0E5A                 LD A,(PRRHS)      ;RHS LIMIT (LINE LEN-1)
3D9D 37                     SCF               ;"PRINTER"
3D9E C9                     RET
3D9F
3D9F ED5B6C5A     POF2:     LD DE,(SPOSNU)
3DA3
3DA3 3A565A       POF3:     LD A,(WINDRHS)
3DA6 C9                     RET
3DA7
3DA7 CD7204       CCRESTOP: CALL RESTOP
3DAA F7                     RST 30H
3DAB F35E                   DW CCRP2-8000H    ;JP TO ROM1 TO DEAL WITH TAB, AT, PAPER, ETC.
3DAD
3DAD F7           POSTFF:   RST 30H
3DAE EE5C                   DW PSTFF2-8000H   ;JP TO ROM1 TO DEAL WITH FN NAME
3DB0
3DB0              ;UTILITY MESSAGES. ENTRY WITH A=NUMBER
3DB0
3DB0 ED5BD65B     UTMSG:    LD DE,(UMSGS)
3DB4
3DB4              ;PRINT MSG "A" FROM LIST AT DE
3DB4
3DB4 F7           POMSG:    RST 30H
3DB5 1B5D                   DW POMSPX-8000H
3DB7
3DB7              ;*************************************************************************
3DB7              ;ANYDEADDR - GET IN DE ADDR OF ROW D, COL E FOR ANY MODE
3DB7              ;EXIT WITH CY IF MODE 2 6-PIX CHARS, Z/NZ =EVEN/ODD
3DB7
3DB7 3A405A       ANYDEADDR: LD A,(MODE)
3DBA A7                     AND A
3DBB 2827                   JR Z,M0DEADDR
3DBD
3DBD 3D                     DEC A
3DBE 2810                   JR Z,M1DEADDR
3DC0
3DC0 3D                     DEC A
3DC1 2832                   JR Z,M2DEADDR
3DC3
3DC3              ;GET IN DE ADDR OF ROW D, COL E FOR MODE 3 (8000H+ROW*CSIZEH*80H+COL*4)
3DC3
3DC3 CD123E       M3DEADDR: CALL CLCPO
3DC6 7B                     LD A,E
3DC7 87                     ADD A,A
3DC8 87                     ADD A,A
3DC9 87                     ADD A,A           ;0-248
3DCA 37                     SCF
3DCB CB1A                   RR D
3DCD 1F                     RRA
3DCE 5F                     LD E,A
3DCF C9                     RET
3DD0
3DD0              ;GET IN DE ADDR OF ROW D, COL E FOR MODE 1 (8000H+ROW*CSIZEH*20H+COL)
3DD0
3DD0 CD123E       M1DEADDR: CALL CLCPO        ;SAY A=256*PIX
3DD3 7A                     LD A,D
3DD4 0F                     RRCA              ;128*
3DD5 0F                     RRCA              ;64*
3DD6 0F                     RRCA              ;32*
3DD7 57                     LD D,A
3DD8 AB                     XOR E
```

```
3DD9 E6E0                AND 0E0H          ;COMBINE LOWER 3 BITS OF SCAN LINE AND COL
3DDB AB                  XOR E
3DDC 5F                  LD E,A
3DDD 7A                  LD A,D
3DDE E61F                AND 1FH           ;UPPER 5 BITS OF SCAN LINE
3DE0 F680                OR 80H
3DE2 57                  LD D,A
3DE3 C9                  RET
3DE4
3DE4            ;GET IN DE ADDR OF ROW D, COL E FOR MODE 0
3DE4
3DE4 CD123E     M0DEADDR:  CALL CLCPO
3DE7 C5                  PUSH BC
3DE8 42                  LD B,D
3DE9 7B                  LD A,E
3DEA 87                  ADD A,A
3DEB 87                  ADD A,A
3DEC 87                  ADD A,A
3DED 4F                  LD C,A            ;BC=PIX COORDS
3DEE EB                  EX DE,HL
3DEF CD3B3E              CALL M0PIXAD
3DF2 EB                  EX DE,HL
3DF3 C1                  POP BC
3DF4 C9                  RET
3DF5
3DF5            ;GET IN DE ADDR OF ROW D, COL E FOR MODE 2 (8000H+ROW*CSIZEH*80H+COL*2)
3DF5            ;                                      OR  (8000H+ROW*CSIZEH*80H+COL*3/2)
3DF5            ;EXIT WITH NC IF 8-PIX CHARS, OR CY AND Z/NZ=EVEN/ODD COL
3DF5
3DF5 CD123E     M2DEADDR:  CALL CLCPO
3DF8 3A355A              LD A,(FL6OR8)
3DFB A7                  AND A
3DFC 7B                  LD A,E
3DFD 2808                JR Z,M2DEADDR2    ;JR IF 6-PIX CHARS
3DFF
3DFF 87                  ADD A,A           ;0-126
3E00 87                  ADD A,A           ;0-252
3E01 37                  SCF
3E02 CB1A                RR D
3E04 1F                  RRA               ;NC=8-PIX CHARS
3E05 5F                  LD E,A
3E06 C9                  RET
3E07
3E07 87         M2DEADDR2: ADD A,A
3E08 83                  ADD A,E           ;A=ORIG COL*3 (0-252)
3E09 37                  SCF
3E0A CB1A                RR D
3E0C 1F                  RRA               ;(0-126 - OFFSET FROM LHS)
3E0D CB43                BIT 0,E           ;Z IF ORIG COL=EVEN
3E0F 5F                  LD E,A
3E10 37                  SCF               ;SIGNAL 6-PIX, Z/NZ=EVEN/ODD
3E11 C9                  RET
3E12
3E12            ;CALC PIX IN D ROWS, ADD OFFSET IF LOWER SCREEN
3E12
3E12 CDD00C     CLCPO:     CALL CALCPIXD
3E15 57                  LD D,A
3E16 3A735A              LD A,(DEVICE)
3E19 A7                  AND A
3E1A C8                  RET Z             ;RET IF UPPER SCREEN
3E1B
3E1B 3A5D5A              LD A,(LSOFF)
3E1E 82                  ADD A,D
3E1F 57                  LD D,A
3E20 C9                  RET
3E21
3E21            ;*************************************************************************
3E21            ;ANY MODE PIXEL ADDRESS FOR PT. B,C (OR POINT B,HL IF THIN PIX)
3E21            ;EXIT: HL=ADDR (8000+) A=X MOD 8
3E21            ;IF MODE 3, CY IF ODD PIXEL
3E21
3E21 3A4D5A     ANYPIXAD:  LD A,(THFATT)
3E24 A7                  AND A
3E25 2006                JR NZ,NTTHINPIX
3E27
3E27 4D                  LD C,L            ;SAVE ORIG X LSB
3E28 CB1C                RR H
3E2A CB1D                RR L              ;HALVE X IF THIN PIX
3E2C FE                  DB 0FEH           ;"JR+1"
3E2D
3E2D 69         NTTHINPIX: LD L,C
3E2E
3E2E 60                  LD H,B
3E2F 3A405A              LD A,(MODE)
3E32 A7                  AND A
3E33 2806                JR Z,M0PIXAD
3E35
3E35 3D                  DEC A
3E36 2820                JR Z,M1PIXAD
3E38
3E38 79                  LD A,C
3E39 1828                JR M1PIXAD2
3E3B
3E3B            ;GET PIXEL ADDR OF POINT C,B IN HL. MODE 0. (Y AXIS HAS ZERO AT TOP)
3E3B            ;ALTERS HL AND A. ADDR=8000-97FF. A=PIX OFFSET
3E3B
3E3B 68         M0PIXAD:   LD L,B
3E3C 78                  LD A,B
3E3D B7                  OR A
3E3E 1F                  RRA
3E3F 1F                  RRA
3E40 37                  SCF
3E41 1F                  RRA
```

```
3E42 E69F                  AND 9FH
3E44 AD                    XOR L
3E45 E6F8                  AND 0F8H
3E47 AD                    XOR L
3E48 67                    LD H,A
3E49 79                    LD A,C
3E4A 07                    RLCA
3E4B 07                    RLCA
3E4C 07                    RLCA
3E4D AD                    XOR L
3E4E E6C7                  AND 0C7H
3E50 AD                    XOR L
3E51 07                    RLCA
3E52 07                    RLCA
3E53 6F                    LD L,A
3E54 79                    LD A,C
3E55 E607                  AND 07H
3E57 C9                    RET
3E58
3E58             ;GET MODE 1 PIXEL ADDR. ENTRY: L=X, H=Y
3E58             ;EXIT: HL=ADDR, B=PIXEL OFFSET (0-7), A=B
3E58
3E58 7D          M1PIXAD:  LD A,L
3E59 A7                    AND A
3E5A CB1C                  RR H              ;NC ROTATED IN
3E5C CB1D                  RR L
3E5E A7                    AND A
3E5F CB1C                  RR H
3E61 CB1D                  RR L
3E63
3E63
3E63 E607        M1PIXAD2: AND 07H
3E65 47                    LD B,A
3E66 37                    SCF
3E67 CB1C                  RR H
3E69 CB1D                  RR L              ;HL=Y/8+X/8+8000H=ADDRESS
3E6B C9                    RET
3E6C
3E6C
3E6C             ;POATTR.SAM
3E6C             ;*************************************************************************
3E6C             ;POATTR01. SET ATTR OF PATTERN DATA AT (HL) FOR MODE 0 OR 1
3E6C             ;ENTRY: HL=SCREEN ADDR. USES HL, BC AND AF
3E6C
3E6C 3A405A      POATTR01: LD A,(MODE)
3E6F A7                    AND A
3E70 7C                    LD A,H
3E71 CBEF                  SET 5,A           ;ADD 2000H FOR USE IF MODE 1
3E73
3E73             ;THIS ENTRY CAN BE USED DIRECTLY BY MODE 0
3E73
3E73 CC490C      POATTR0:  CALL Z,CTAA
3E76
3E76 67                    LD H,A
3E77
3E77             ;ENTRY POINT IF HL ALREADY PTS TO ATTR
3E77
3E77 ED4B4E5A    SETATTR:  LD BC,(ATTRT)
3E7B 7E                    LD A,(HL)
3E7C A9                    XOR C
3E7D A0                    AND B
3E7E A9                    XOR C
3E7F ED4B505A              LD BC,(PFLAGT)
3E83 CB61                  BIT 4,C
3E85 2808                  JR Z,POATTR1        ;JR IF NOT INK 9
3E87
3E87 F607                  OR 07H
3E89 CB6F                  BIT 5,A
3E8B 2802                  JR Z,POATTR1
3E8D
3E8D EE07                  XOR 7
3E8F
3E8F CB71        POATTR1:  BIT 6,C
3E91 2808                  JR Z,POATTR2        ;JR IF NOT PAPER 9
3E93
3E93 F638                  OR 38H
3E95 CB57                  BIT 2,A
3E97 2802                  JR Z,POATTR2
3E99
3E99 EE38                  XOR 38H
3E9B
3E9B 77          POATTR2:  LD (HL),A
3E9C C9                    RET
3E9D
3E9D             ;COMPARE TWO STRINGS FROM STACK (OVER <16K). PAGING UNALTERED ON RETURN.
3E9D             ;EXIT: Z IF STRINGS MATCH, CY IF S1<S2, NZ,NC IF S1>S2. HL=S1 PTR
3E9D             ;USES HL,BC,AF, HL" DE", BC" AF"
3E9D
3E9D CD493C      STRCOMP:  CALL R1OSR
3EA0 E5                    PUSH HL           ;S1 PTR
3EA1 CD073F                CALL UNSTKPRT     ;BC=S2 LEN, DE=S2 ST, A=PORT VALUE
3EA4 D5                    PUSH DE           ;S2 ST
3EA5 F5                    PUSH AF           ;S2 PORT VALUE
3EA6 C5                    PUSH BC           ;S2 LEN
3EA7 CD073F                CALL UNSTKPRT     ;BC=S1 LEN, DE=S1 ST, A=PORT
3EAA E1                    POP HL            ;S2 LEN
3EAB A7                    AND A
3EAC ED42                  SBC HL,BC         ;S2 LEN-S1 LEN
3EAE 09                    ADD HL,BC
3EAF 3002                  JR NC,STRCOMP2    ;JR IF BC<=HL
3EB1
3EB1 44                    LD B,H
```

```
3EB2 4D                     LD C,L            ;MAKE BC=SHORTEST LEN
3EB3
3EB3 6F          STRCOMP2:  LD L,A            ;L=S1 PORT
3EB4 08                     EX AF,AF'         ;SAVE Z IF LENS EQUAL, CY IF S1 LEN GRTR.
3EB5 78                     LD A,B
3EB6 FE40                   CP 40H
3EB8 D2283A                 JP NC,STLERR      ;ONLY DEAL WITH STRINGS<16K
3EBB
3EBB F1                     POP AF
3EBC 67                     LD H,A            ;H=S2 PORT
3EBD C5                     PUSH BC
3EBE 0EFB                   LD C,251
3EC0
3EC0 D9                     EXX
3EC1 C1                     POP BC            ;BC"=SHORTEST LEN
3EC2 E1                     POP HL            ;HL"=S2 ST
3EC3 180D                   JR SCOMPBG
3EC5
3EC5 D9          SCOMPLP:   EXX
3EC6 ED69                   OUT (C),L         ;S1 PAGE SEL
3EC8 1A                     LD A,(DE)         ;S1 CHAR
3EC9 13                     INC DE
3ECA ED61                   OUT (C),H         ;S2 PAGE SEL
3ECC
3ECC D9                     EXX
3ECD BE                     CP (HL)           ;S2 CHAR
3ECE 200A                   JR NZ,SCOMPEX     ;JR WITH CY IF S2 GRTR; NZ,NC IF S1 GRTR
3ED0
3ED0 23                     INC HL
3ED1 0B                     DEC BC
3ED2
3ED2 78          SCOMPBG:   LD A,B
3ED3 B1                     OR C
3ED4 20EF                   JR NZ,SCOMPLP     ;LOOP UNTIL STRINGS MATCH OVER BC CHARS
3ED6
3ED6 08                     EX AF,AF'         ;Z IF LENS EQUAL - STRINGS MATCH
3ED7 2801                   JR Z,SCOMPEX      ;** COMPARISON BUG FIX
3ED9
3ED9 3F                     CCF               ;CY IF S2 STRING LONGER - IE GREATER
3EDA                                          ;NZ,NC IF S1 STRING LONGER - IE GREATER
3EDA
3EDA E1          SCOMPEX:   POP HL            ;S1 PTR
3EDB
3EDB 08          SCOMPC:    EX AF,AF'
3EDC F1                     POP AF
3EDD D3FA                   OUT (250),A
3EDF F1                     POP AF
3EE0 D3FB                   OUT (251),A
3EE2 08                     EX AF,AF'
3EE3 C9                     RET
3EE4
3EE4
3EE4             ;STRING BUFFER FETCH. COPY STRING ON FPCS TO "INSTBUF" IN COMMON MEM. ERROR IF
3EE4             ;LEN >255 OR 0. ON EXIT, BC AND A=LEN, DE=START, PAGING UNALTERED.
3EE4
3EE4 CDEA3E      SBUFFET:   CALL SBFSR
3EE7 C0                     RET NZ
3EE8
3EE8 CF          INVARG:    RST 08H
3EE9 1B                     DB 27
3EEA
3EEA             ;AS SBUFFET, BUT LEN ZERO GIVES Z FLAG, NOT AN ERROR MSG
3EEA
3EEA 3EFF        SBFSR:     LD A,0FFH
3EEC
3EEC 08          SBFSR2:    EX AF,AF'
3EED CD493C                 CALL R1OSR
3EF0 CDDC3F                 CALL GETSTRING    ;AND SELECT PAGE
3EF3 08                     EX AF,AF'
3EF4 80                     ADD A,B           ;ADD FF OR FE, LEN MSB
3EF5 38F1                   JR C,INVARG       ;ERROR IF T$ LEN >FF  OR >01FF
3EF7
3EF7 78                     LD A,B
3EF8 B1                     OR C
3EF9 28E0                   JR Z,SCOMPC       ;RET IF LEN=0
3EFB                                          ;A=LEN.
3EFB EB                     EX DE,HL
3EFC 11004F                 LD DE,INSTBUF     ;256 BYTES IN PAGE 0
3EFF C5                     PUSH BC
3F00 D5                     PUSH DE
3F01 EDB0                   LDIR              ;COPY T$ TO PAGE 0
3F03 D1                     POP DE
3F04 C1                     POP BC            ;BC=LEN, A=C. NC
3F05 18D4                   JR SCOMPC         ;NZ HERE FROM 'OR C' - SCOMPC PRESERVES IT TOO
3F07
3F07             ;UNSTACK STRING AND GET PORT VALUE NEEDED TO SWITCH IT IN
3F07
3F07 CD011D      UNSTKPRT:  CALL STKFETCH     ;A=PAGE, DE=START, BC=LEN
3F0A 67                     LD H,A
3F0B DBFB                   IN A,(251)
3F0D AC                     XOR H
3F0E E6E0                   AND 0E0H          ;USE UPPER 3 BITS FROM PORT
3F10 AC                     XOR H
3F11 C9                     RET               ;A=PORT VALUE
3F12
3F12             ;ADDRESS OF IDERR GOES INTO SOME CHANNELS
3F12
3F12 CF          IDERR:     RST 08H
3F13 14                     DB 20             ;"Invalid device"
3F14
3F14             ;CHECK IF OK TO USE ABC BYTES (PAGE FORM). EXITS WITH ABC, DE CHANGED.
3F14
3F14 F5          TSTRMBIG:      PUSH AF
```

```
3F15 C5                     PUSH BC
3F16 CD1C3F                 CALL TSTRMABC
3F19 C1                     POP BC
3F1A F1                     POP AF
3F1B C9                     RET
3F1C
3F1C                ;CHECK IF OK TO USE ABC BYTES (PAGE FORM). EXITS WITH DE CHANGED. NC. AHL=
3F1C                ;PAGE FORM OF NEW WKEND
3F1C
3F1C 60             TSTRMABC:   LD H,B
3F1D 69                     LD L,C
3F1E
3F1E CD2120         TSTRMAHL:   CALL AHLNORM
3F21 44                     LD B,H
3F22 4D                     LD C,L          ;ABC=19-BIT SPACE
3F23 FE                     DB 0FEH         ;"JR+1"
3F24
3F24                ;CHECK IF OK TO USE BC BYTES (0-FFFF). ERROR IF NOT. EXITS WITH BC UNCHANGED, NC
3F24                ;AHL=PAGE FORM OF NEW WKEND, DE=FREE, OR A HIGH VALUE IF FREE>=64K
3F24
3F24 AF             TESTROOM:   XOR A
3F25
3F25 C5                     PUSH BC
3F26 57                     LD D,A          ;DBC=SPACE
3F27 CD1620                 CALL WENORMAL   ;GET AHL=WKEND (19 BIT)
3F2A 09                     ADD HL,BC
3F2B 8A                     ADC A,D         ;AHL=NEW VALUE AFTER BC USED
3F2C CDF21F                 CALL PAGEFORM   ;AHL=PAGE FORM OF NEW WKEND
3F2F F5                     PUSH AF
3F30 E5                     PUSH HL
3F31 EB                     EX DE,HL
3F32 4F                     LD C,A
3F33 3AB15C                 LD A,(RAMTOPP)
3F36 2AB25C                 LD HL,(RAMTOP)  ;AHL=RAMTOP
3F39 CDE71F                 CALL SUBAHLCDE  ;AHL=ROOM
3F3C 380D                   JR C,OOMERR     ;ERROR IF NEW WKEND PAGE WOULD BE ABOVE RAMTOP
3F3E
3F3E CD2120                 CALL AHLNORM    ;AHL=19 BIT ROOM
3F41 EB                     EX DE,HL
3F42 A7                     AND A
3F43 2802                   JR Z,TRM2
3F45
3F45 CBFA                   SET 7,D         ;IF >64K FREE, MAKE DE A HI VALUE
3F47
3F47 E1             TRM2:       POP HL
3F48 F1                     POP AF          ;AHL=NEW WKEND
3F49 C1                     POP BC          ;SPACE
3F4A C9                     RET
3F4B
3F4B CF             OOMERR:     RST 08H
3F4C 01                     DB 1            ;"OUT OF MEMORY"
3F4D
3F4D
3F4D                ;COPY STRING FROM (DE), LEN BC, PORT VALUE A, TO WKSPACE. SOURCE CAN BE ANYWHERE
3F4D                ;AND NEED NOT BE SWITCHED IN ON ENTRY. LEN TRUNCATED TO <=255 WITHOUT ANY MSG.
3F4D                ;LEN 0 WILL CRASH! ROM1 MUST BE SWITCHED OFF ON ENTRY!
3F4D                ;EXIT: HL=PAST ROOM END, DE=ROOM START, DE"=PAST STRING END, B"=0, C"=URPORT,
3F4D                ;H"=SRC PORT VALUE, L"=WKSPACE PORT. WORKSPACE IS SWITCHED IN.
3F4D                ;NOTE: REASONABLY FAST FOR SHORT STRINGS
3F4D                ;USED BY READ AND VAL ONLY
3F4D
3F4D                ;SCOPYWK:   PUSH DE         ;SRC
3F4D                 ;          INC B
3F4D                  ;         DEC B
3F4D                   ;        JR Z,SCOPYWK2   ;JR IF LEN<=256
3F4D
3F4D                ;           LD BC,0FFH      ;ELSE TRUNCATE - STRMOVE ONLY ALLOWS SHORT STRINGS
3F4D
3F4D                ;SCOPYWK2:  CALL WKROOM     ;DE=ROOM, HL=END BYTE, BC AND A UNCHANGED
3F4D                 ;          LD B,C          ;LEN COUNTER
3F4D                  ;         LD H,A          ;H=STRING PORT VALUE
3F4D                   ;        IN A,(251)
3F4D                    ;       LD L,A          ;L=WKSPACE PORT VALUE
3F4D                     ;      PUSH DE         ;WKSPACE
3F4D
3F4D                ;           EXX
3F4D                 ;          POP HL          ;DEST TO HL"
3F4D                  ;         LD D,H
3F4D                   ;        LD E,L          ;KEEP A COPY OF THE START IN DE
3F4D                    ;       EXX
3F4D
3F4D                ;           POP DE          ;SRC
3F4D
3F4D                ;;MOVE B BYTES FROM (DE) PORT VALUE H TO (HL") PORT VALUE L. EXIT WITH ALT REGS.
3F4D
3F4D                ;           LD C,251
3F4D
3F4D                ;STRMVLP:   OUT (C),H       ;SRC PAGE
3F4D                 ;          LD A,(DE)       ;SRC BYTE
3F4D                  ;         INC DE
3F4D                   ;        OUT (C),L       ;DEST PAGE
3F4D
3F4D                ;           EXX
3F4D                 ;          LD (HL),A       ;TO WKSPACE AT HL"
3F4D                  ;         INC HL
3F4D                   ;        EXX
3F4D                    ;
3F4D                     ;      DJNZ STRMVLP    ;ABOUT 80 Ts/BYTE
3F4D
3F4D                ;           EXX             ;RET WITH REGS SWAPPED
3F4D
3F4D                ;COPY BC BYTES FROM DE TO WKSPACE
3F4D
```

```
3F4D 04          SCOPYWK:   INC B
3F4E 05                     DEC B
3F4F 2803                   JR Z,SCOPYWK2     ;JR IF LEN<=256
3F51
3F51 01FF00                 LD BC,0FFH        ;ELSE TRUNCATE
3F54
3F54 CD493C      SCOPYWK2:  CALL R1OSR
3F57 EB                     EX DE,HL          ;SRC TO HL
3F58 11004F                 LD DE,INSTBUF
3F5B D5                     PUSH DE
3F5C C5                     PUSH BC
3F5D EDB0                   LDIR
3F5F C1                     POP BC
3F60 C5                     PUSH BC
3F61 CD801E                 CALL WKROOM
3F64 C1                     POP BC
3F65 E1                     POP HL
3F66 D5                     PUSH DE
3F67 EDB0                   LDIR
3F69 EB                     EX DE,HL          ;HL=PAST END
3F6A D1                     POP DE            ;START
3F6B 2B                     DEC HL
3F6C 360D                   LD (HL),0DH
3F6E F1                     POP AF
3F6F D3FA                   OUT (250),A
3F71 F1                     POP AF
3F72 C9                     RET
3F73
3F73             ;CALLED BY LENGTH FN AND TAPEMN
3F73
3F73 CDAA13      LENGSR:    CALL LOOKVARS     ;HL POINTS TO NUMERIC VALUE, OR STRING LEN DATA
3F76 F5                     PUSH AF           ;FOUND/NOT FOUND FLAG
3F77 C5                     PUSH BC
3F78 D5                     PUSH DE
3F79 222151                 LD (MEMVAL),HL
3F7C DBFB                   IN A,(URPORT)
3F7E 112351                 LD DE,MEMVAL+2
3F81 12                     LD (DE),A
3F82 13                     INC DE
3F83 0E07                   LD C,7
3F85 CD4C2D                 CALL SCOPN2       ;LDIR 7 BYTES, SELCHADP
3F88 D1                     POP DE
3F89 C1                     POP BC            ;TYPE
3F8A F1                     POP AF            ;FLAGS
3F8B C9                     RET
3F8C
3F8C                        DS 3F8CH-$,0
3F8C
3F8C             ;FIXED ROUTINES
3F8C             ;*************************************************************************
3F8C             ;UNSTACK A 5-BYTE NUMBER TO AN ADDRESS IN A (PAGE) AND HL (8000-BFFF)
3F8C             ;(RES 7,H IF 5-BYTE IS A LENGTH, TO GIVE PAGE, +0000-3FFFH). GIVES IOOR ERROR
3F8C             ; IF NUMBER IS NEGATIVE OR >07FFFF
3F8C
3F8C EF          UNSTLEN:   DB CALC           ;N
3F8D E2                     DB STK16K         ;N,16K
3F8E 08                     DB MOD            ;N MOD 16K
3F8F DB                     DB RCL3           ;N MOD 16K,INT(N/16K) (PLACED BY MOD)
3F90 33                     DB EXIT
3F91
3F91 CD331D                 CALL GETBYTE
3F94 FE21                   CP 21H
3F96 D2391D                 JP NC,IOORERR     ;PAGE MUST BE 00-20H (0=ROM, 1-20=RAM)
3F99
3F99 F5                     PUSH AF
3F9A CD2E1D                 CALL GETINT       ;TO HL AND BC
3F9D F1                     POP AF
3F9E C9                     RET               ;A=PAGE
3F9F
3F9F             ;NEW POKE/DPOKE SR
3F9F
3F9F CD6F12      NPDPS:     CALL PDPSUBR
3FA2 7C                     LD A,H
3FA3 FEC0                   CP 0C0H
3FA5 D8                     RET C
3FA6
3FA6 184A                   JR INCURPAGE
3FA8
3FA8             ;SELECT SCREEN, ROM1 OFF
3FA8
3FA8 DBFA        SPSSR:     IN A,(250)
3FAA 327A5A                 LD (CLRP),A
3FAD E6BF                   AND 0BFH
3FAF D3FA                   OUT (250),A       ;ROM1 OFF
3FB1
3FB1             ;STORE PAGE SCREEN SELECT
3FB1
3FB1 DBFB        SPSS:      IN A,(251)
3FB3 32795A                 LD (CURP),A
3FB6
3FB6 3A785A      SELSCRN:   LD A,(CUSCRNP)    ;SCREEN PAGE
3FB9 1824                   JR SELURPG
3FBB
3FBB             ;READ BYTE FROM SCREEN AT HL, FORCING SCREEN ON, ROM1 OFF
3FBB
3FBB CDA83F      SREAD:     CALL SPSSR        ;SELECT SCREEN, ROM1 OFF
3FBE 7E                     LD A,(HL)
3FBF
3FBF 08          RCURPR:    EX AF,AF'
3FC0 3A7A5A                 LD A,(CLRP)
3FC3 D3FA                   OUT (250),A
3FC5 3E                     DB 3EH            ;'JR+1'
3FC6
```

```
3FC6                TRCURP:
3FC6 08             RCURP:      EX AF,AF'
3FC7
3FC7 3A795A         RCUR2:      LD A,(CURP)
3FCA D3FB                       OUT (251),A
3FCC 08                         EX AF,AF'
3FCD C9                         RET
3FCE
3FCE                ;SET CHADP VAR AND SWITCH IT IN
3FCE
3FCE 32965A         SETCHADP:       LD (CHADP),A
3FD1
3FD1                ;ROM1 OFF, SELCHADP
3FD1
3FD1 DBFA           R1OCHP:         IN A,(250)
3FD3 E6BF                           AND 0BFH
3FD5 D3FA                           OUT (250),A         ;ROM1 OFF
3FD7
3FD7                ;SWITCH IN CHADP
3FD7
3FD7 3A965A         SELCHADP:       LD A,(CHADP)        ;CHAD PAGE
3FDA 1803                           JR TSURPG
3FDC
3FDC                ;UNSTACK A STRING AND SELECT IT"S PAGE. DE=START (8000-BFFF), BC=LEN
3FDC
3FDC                TGTSTR:
3FDC CD011D         GETSTRING:      CALL STKFETCH       ;A=PAGE, DE=START, BC=LEN
3FDF
3FDF                ;SELECT UPPER RAM PAGE
3FDF
3FDF                SELURPG:
3FDF E5             TSURPG:     PUSH HL
3FE0 67                         LD H,A
3FE1 DBFB                       IN A,(251)
3FE3 AC                         XOR H
3FE4 E6E0                       AND 0E0H            ;KEEP TOP 3 BITS FROM PORT
3FE6 AC                         XOR H
3FE7 D3FB                       OUT (251),A
3FE9 E1                         POP HL
3FEA C9                         RET
3FEB
3FEB                ;INCREMENT UPPER RAM PAGE, ENSURE DE IS NOT IN C000-FFFF AREA
3FEB                ;USES A, ALTERS D
3FEB
3FEB CBB2           INCURPDE:       RES 6,D
3FED 1805                           JR INCURCOM
3FEF
3FEF                ;INC PAGE AND ADJUST HL POINTER IF NEEDED
3FEF
3FEF CB74           CHKHL:          BIT 6,H
3FF1 C8                             RET Z
3FF2
3FF2                ;INCREMENT UPPER RAM PAGE, ENSURE HL IS NO IN C000-FFFF AREA
3FF2                ;USES A, ALTERS H
3FF2
3FF2 CBB4           INCURPAGE:      RES 6,H
3FF4
3FF4 DBFB           INCURCOM:       IN A,(251)
3FF6 3C                             INC A
3FF7 18E6                           JR SELURPG
3FF9
3FF9 CBF4           DECURPAGE:      SET 6,H
3FFB DBFB                           IN A,(251)
3FFD 3D                             DEC A
3FFE 18DF                           JR SELURPG
4000
4000                            INCLUDE TMISCX1.SAM   ;BUFFER CODE: RENUM, GET, DELETE, KEYIN, POP,
4000                            ORG 0C000H
C000                ;MISCX1.SAM - START OF UPPER ROM. FIRST PART IS SECTIONS THAT ARE COPIED
C000                ;TO A RAM BUFFER FOR EXECUTION SO THE UPPER ROM CAN BE PAGED OUT AND
C000                ;THE BASIC PROGRAM GOT AT.
C000
C000                            ORG INSTBUF
4F00
4F00                ;MOVED TO INSTBUF FOR EXECUTION
4F00
4F00 210A00         RNMP2:      LD HL,10
4F03 22425A                     LD   (RLINE),HL
4F06 22895B                     LD   (RSTEP),HL
4F09 CD4B3B                     CALL BRKLSSL        ;ASSESS <N> TO <M>
4F0C DF                         RST  18H
4F0D 21425A                     LD   HL,RLINE
4F10 FE8C                       CP   LINETOK
4F12 CCB550                     CALL Z,REVAL
4F15 21895B                     LD   HL,RSTEP
4F18 FE8F                       CP   STEPTOK
4F1A CCB550                     CALL Z,REVAL
4F1D CD153B                     CALL CHKEND
4F20
4F20 0618                       LD B,24
4F22 CD243F                     CALL TESTROOM       ;ABORT IF <6K FREE
4F25 CDB13F                     CALL SPSS
4F28
4F28 2A425A         MAKETABLE:  LD   HL,(RLINE)
4F2B ED4B895B                   LD   BC,(RSTEP)
4F2F 1102C0                     LD   DE,SBN+2
4F32
4F32 D9                         EXX
4F33 210080                     LD   HL,SBO
4F36 E5                         PUSH HL
4F37 110000                     LD   DE,0000H
4F3A
4F3A E1             MKTBLP:     POP  HL             ;SBO
```

```
4F3B 72                  LD   (HL),D
4F3C 23                  INC  HL
4F3D 73                  LD   (HL),E
4F3E 2B                  DEC  HL
4F3F E5                  PUSH HL             ;SBO
4F40 2A7F5A              LD   HL,(LAST)      ;FEFF IF END OF PROG WANTED
4F43 ED52                SBC  HL,DE
4F45 3821                JR C,RENUM3         ;JR IF LINE NUMBER JUST PLACED IS PAST BLOCK END
4F47                                         ;(COULD BE FF?? PROG TERMINATOR)
4F47 CD573C              CALL GTRLNN
4F4A 37                  SCF
4F4B ED52                SBC  HL,DE
4F4D 30EB                JR   NC,MKTBLP
4F4F
4F4F E1                  POP  HL
4F50 23                  INC  HL
4F51 23                  INC  HL
4F52 E5                  PUSH HL
4F53
4F53 D9                  EXX
4F54 EB                  EX   DE,HL
4F55 72                  LD   (HL),D
4F56 23                  INC  HL
4F57 73                  LD   (HL),E
4F58 23                  INC  HL
4F59 24                  INC H
4F5A 280A                JR Z,NRFLERR
4F5C
4F5C 25                  DEC H
4F5D EB                  EX   DE,HL
4F5E 09                  ADD  HL,BC          ;ADD LINE,STEP TO GET NEXT NEW LINE NO.
4F5F 3805                JR C,NRFLERR
4F61
4F61 7C                  LD A,H
4F62 D9                  EXX
4F63
4F63 3C                  INC  A
4F64 20D4                JR   NZ,MKTBLP      ;LOOP UNLESS LINE NUMBERS TOO HIGH (>FEFF)
4F66
4F66 CF          NRFLERR:  RST 08H
4F67 21                  DB 33               ;'No room for line'
4F68
4F68 C1          RENUM3:   POP BC
4F69 2A0080              LD HL,(SBO)
4F6C 22C85A              LD (TEMPW1),HL      ;SAVE IT SO IT CAN BE READ WHEN SCREEN SWITCHED OUT
4F6F CDC63F              CALL RCURP
4F72 21455C              LD HL,PPC
4F75 CDC550              CALL TRANSHL
4F78 C5                  PUSH BC
4F79 3A475C              LD A,(SUBPPC)
4F7C F5                  PUSH AF
4F7D
4F7D 7A                  LD A,D
4F7E 3C                  INC A
4F7F 2001                JR NZ,REN3          ;JR IF DE IS A REAL LINE NUMBER
4F81
4F81 5F                  LD E,A              ;ELSE DE=FF00, MAX LINE+1
4F82
4F82 AF          REN3:     XOR A              ;NC
4F83 D5                  PUSH DE
4F84 D9                  EXX
4F85 D1                  POP  DE
4F86 ED42                SBC  HL,BC
4F88 ED42                SBC  HL,BC
4F8A ED52                SBC  HL,DE
4F8C 30D8                JR   NC,NRFLERR     ;ERROR IF MOVING LINES ONTO OTHERS
4F8E
4F8E 2AC85A              LD   HL,(TEMPW1)
4F91 55                  LD   D,L
4F92 5C                  LD   E,H
4F93 2A425A              LD   HL,(RLINE)
4F96 ED52                SBC  HL,DE
4F98 38CC                JR C,NRFLERR
4F9A
4F9A CDD64F              CALL CHGREF
4F9D
4F9D             ;CHANGE LINE NUMBERS
4F9D
4F9D 2A7D5A              LD   HL,(FIRST)
4FA0 CD4D1A              CALL FNDLINE
4FA3
4FA3 46          CHGLL:    LD   B,(HL)
4FA4 23                  INC  HL
4FA5 4E                  LD   C,(HL)
4FA6 E5                  PUSH HL
4FA7 CDD250              CALL TRANSFORM
4FAA E1                  POP  HL
4FAB 380F                JR C,REN5           ;JR IF FINISHED BLOCK
4FAD
4FAD 71                  LD   (HL),C
4FAE 2B                  DEC  HL
4FAF 70                  LD   (HL),B         ;ALTER LINE NUMBER
4FB0 23                  INC  HL
4FB1 23                  INC  HL
4FB2 4E                  LD   C,(HL)
4FB3 23                  INC  HL
4FB4 46                  LD   B,(HL)         ;BC=LINE LEN
4FB5 23                  INC  HL
4FB6 09                  ADD  HL,BC          ;PT TO NEXT LINE
4FB7 CDEF3F              CALL CHKHL
4FBA 18E7                JR   CHGLL
4FBC
4FBC 21495C      REN5:     LD HL,EPPC
```

```
4FBF CDC550               CALL TRANSHL
4FC2 2E6C                 LD L,>SDTOP
4FC4 CDC550               CALL TRANSHL
4FC7 CD9806               CALL MCLS          ;CLEAR ENTIRE SCREEN
4FCA C3D033               JP GT4P            ;UNSTACK STAT, PPC, SET UP FOR GOTO NEXT, COMP
4FCD
4FCD            ;CHANGE REFERENCES TO LINE NUMBERS
4FCD
4FCD CD121D     FAILED:   CALL FDELETE
4FD0
4FD0 C1         NBLKL:    POP  BC
4FD1 C1                   POP  BC
4FD2 C1                   POP  BC
4FD3 C39B50               JP CHGR7
4FD6
4FD6 DD21AA50   CHGREF:   LD   IX,RENTAB-1
4FDA
4FDA DD23       RENCL:    INC  IX
4FDC CD321F               CALL ADDRPROG
4FDF 7E                   LD A,(HL)
4FE0 3C                   INC A
4FE1 C8                   RET Z
4FE2
4FE2 22AF5A               LD   (CLA),HL
4FE5 23                   INC  HL
4FE6 23                   INC  HL
4FE7 23                   INC  HL
4FE8 23                   INC  HL
4FE9 22975A               LD   (CHAD),HL
4FEC
4FEC DD7E00     RLOOP:    LD   A,(IX+0)
4FEF A7                   AND  A
4FF0 C8                   RET  Z
4FF1
4FF1 5F                   LD   E,A
4FF2 CD9C1D               CALL SRCHPROG
4FF5 30E3                 JR   NC,RENCL
4FF7
4FF7 32475C               LD   (SUBPPC),A
4FFA DBFB                 IN   A,(251)
4FFC 32965A               LD   (CHADP),A
4FFF DD7E00               LD A,(IX+0)
5002 FE8C                 CP LINETOK
5004 200B                 JR NZ,CHGRY
5006
5006 2B                   DEC HL
5007 2B                   DEC HL
5008 7E                   LD A,(HL)          ;PT TO BYTE BEFORE "LINE"
5009 FE24                 CP "$"
500B 2804                 JR Z,CHGRY         ;RENUM E.G. SAVE ASD$ LINE 10
500D
500D FE22                 CP 22H
500F 20DB                 JR NZ,RLOOP        ;NO RENUM UNLESS E.G. SAVE "NAME" LINE 10
5011
5011 DF         CHGRY:    RST  18H
5012
5012 FE8E       CHGR1:    CP   TOTOK
5014 2001                 JR   NZ,CHGR2
5016
5016 E7                   RST  20H
5017
5017 E5         CHGR2:    PUSH HL
5018 06FF                 LD   B,0FFH
501A FEDE                 CP   0DEH          ;ONTOK
501C 200B                 JR   NZ,CHGR5
501E
501E E7         CHGR3:    RST  20H
501F FE3B                 CP   ":"
5021 20FB                 JR   NZ,CHGR3
5023
5023 E7                   RST  20H
5024 D1                   POP  DE
5025 E5                   PUSH HL
5026 11                   DB 11H             ;'JR+2'
5027
5027 23         CHGR4:    INC HL
5028 7E                   LD A,(HL)
5029
5029 CD303B     CHGR5:    CALL NUMERIC
502C 04                   INC  B
502D 38F8                 JR   C,CHGR4
502F
502F C5                   PUSH BC
5030
5030 7E         CHGR6:    LD A,(HL)
5031 23                   INC HL
5032 FE20                 CP " "
5034 28FA                 JR Z,CHGR6
5036
5036 E5                   PUSH HL
5037 FE0E                 CP   0EH
5039 2095                 JR   NZ,NBLKL
503B
503B CD201D               CALL HLTOFPCS
503E 22975A               LD   (CHAD),HL
5041 CD2E1D               CALL GETINT
5044 DF                   RST  18H
5045 CD723A               CALL COCRCOTO      ;CHECK FOR CR/:/TO/,
5048 2083                 JR NZ,FAILED
504A
504A CDD250               CALL TRANSFORM
504D 3881                 JR   C,NBLKL
504F
```

```
504F CDDD1C               CALL STACKBC      ;EXITS WITH DE=STKEND
5052 21FBFF               LD HL,-5
5055 19                   ADD HL,DE         ;POINT TO TOP ITEM ON FPCS
5056 D1                   POP  DE
5057 CD4B2B               CALL LDI5         ;COPY TO 5-BYTE BUFFER IN LINE
505A CD7E01               CALL JPFSTRS      ;BC=NEW LEN OF NUMBER (DIGITS)
505D F1                   POP  AF           ;ORIG NUMBER OF DIGITS
505E D1                   POP  DE
505F 2AAF5A               LD   HL,(CLA)
5062
5062            ;ADJUST LINE LEN
5062            ;ENTRY: HL PTS TO START OF BASIC LINE, DE PTS TO SPACE TO OPEN OR CLOSE IN THE
5062            ;LINE, BC=NEW LENGTH OF SPACE, A=OLD LENGTH.
5062
5062 91         ADJLINE:  SUB C
5063 282A                 JR Z,ADLNO        ;JR IF LENS THE SAME
5065
5065 23                   INC  HL
5066 23                   INC  HL
5067 C5                   PUSH BC
5068 F5                   PUSH AF
5069 4E                   LD   C,(HL)
506A 23                   INC  HL
506B 46                   LD   B,(HL)       ;CURRENT LINE LEN
506C E5                   PUSH HL
506D ED44                 NEG
506F 6F                   LD   L,A
5070 17                   RLA
5071 9F                   SBC  A,A
5072 67                   LD   H,A
5073 09                   ADD  HL,BC
5074 44                   LD   B,H
5075 4D                   LD   C,L
5076 E1                   POP  HL
5077 70                   LD   (HL),B
5078 2B                   DEC  HL
5079 71                   LD   (HL),C       ;NEW LINE LEN
507A F1                   POP  AF
507B 0600                 LD   B,0
507D 3009                 JR   NC,ADJL2     ;JR IF CLOSEING UP SPACE
507F
507F ED44                 NEG
5081 4F                   LD   C,A
5082 EB                   EX   DE,HL
5083 CD1B1E               CALL MAKEROOM
5086 1805                 JR   ADJL3
5088
5088 4F         ADJL2:    LD   C,A
5089 EB                   EX DE,HL
508A CD521E               CALL RECLAIM2
508D
508D EB         ADJL3:    EX DE,HL
508E C1                   POP BC
508F
508F 21A05B     ADLNO:    LD HL,PRNBUFF
5092 EDB0                 LDIR
5094 EB                   EX DE,HL
5095 0E06                 LD C,6
5097 09                   ADD HL,BC
5098 22975A               LD (CHAD),HL
509B
509B CD793A     CHGR7:    CALL RCRC         ;RST 18, CRCOLON
509E CAEC4F               JP Z,RLOOP
50A1
50A1 CD723A               CALL COCRCOTO     ;Z IF TO/,
50A4 C2EC4F               JP NZ,RLOOP
50A7
50A7 E7                   RST 20H
50A8 C31250               JP CHGR1
50AB
50AB CD         RENTAB:   DB 0CDH           ;DELETETOK
50AC DD                   DB 0DDH           ;ONERRORTOK
50AD 8C                   DB 08CH           ;LINETOK
50AE BE                   DB 0BEH           ;LLISTTOK
50AF BD                   DB 0BDH           ;LISTTOK
50B0 BA                   DB 0BAH           ;RESTORETOK
50B1 B4                   DB 0B4H           ;GOTOTOK
50B2 B5                   DB 0B5H           ;GOSUBTOK
50B3 B0                   DB 0B0H           ;RUNTOK
50B4 00                   DB 0
50B5
50B5 E5         REVAL:    PUSH HL
50B6 CD8D3B               CALL GIR
50B9 E1                   POP HL
50BA D8                   RET C
50BB
50BB 78                   LD A,B
50BC B1                   OR C
50BD CA391D               JP Z,IOORERR
50C0
50C0 71                   LD (HL),C
50C1 23                   INC HL
50C2 70                   LD (HL),B
50C3 DF                   RST 18H
50C4 C9                   RET
50C5
50C5 4E         TRANSHL:  LD C,(HL)
50C6 23                   INC HL
50C7 46                   LD B,(HL)
50C8 E5                   PUSH HL
50C9 CDD250               CALL TRANSFORM
50CC E1                   POP HL
50CD D8                   RET C             ;RET IF NOT IN BLOCK
```

```
50CE
50CE 70                     LD (HL),B
50CF 2B                     DEC HL
50D0 71                     LD (HL),C
50D1 C9                     RET
50D2
50D2 2A7F5A     TRANSFORM:  LD   HL,(LAST)
50D5 A7                     AND  A
50D6 ED42                   SBC  HL,BC
50D8 D8                     RET  C
50D9
50D9 2A7D5A                 LD   HL,(FIRST)
50DC ED42                   SBC  HL,BC
50DE 2802                   JR   Z,TRANS2
50E0
50E0 3F                     CCF
50E1 D8                     RET  C
50E2
50E2 CDB13F     TRANS2:     CALL SPSS
50E5 210280                 LD   HL,SBO+2
50E8
50E8 7E         TRANS3:     LD A,(HL)
50E9 B8                     CP B
50EA 2004                   JR NZ,TRANS4
50EC
50EC 23                     INC HL
50ED 7E                     LD A,(HL)
50EE 2B                     DEC HL
50EF B9                     CP C
50F0
50F0 3004       TRANS4:     JR   NC,TRANS5
50F2
50F2 23                     INC  HL
50F3 23                     INC  HL
50F4 18F2                   JR   TRANS3
50F6
50F6 7C         TRANS5:     LD   A,H
50F7 C640                   ADD  A,0+(SBN-SBO)/256   ;NC
50F9 67                     LD   H,A
50FA 46                     LD   B,(HL)
50FB 23                     INC  HL
50FC 4E                     LD   C,(HL)
50FD C3C63F     TRANSF:     JP   RCURP
5100
5100 0200=      RENLN       EQU TRANSF+3-RNMP2
5100
5100                        ORG INSTBUF
4F00
4F00 CDF72C     GETP2:      CALL SYNTAX1      ;VALID VARIABLE
4F03 CD153B                 CALL CHKEND
4F06
4F06 213B5C                 LD HL,FLAGS
4F09 E5                     PUSH HL
4F0A CBAE                   RES 5,(HL)        ;'NO KEY'
4F0C CDE704                 CALL WKBR         ;GET KEY WITH BREAK OPTION
4F0F E1                     POP HL            ;FLAGS
4F10 CB76                   BIT 6,(HL)
4F12 2008                   JR NZ,GT2         ;JR IF NUMERIC VALUE WANTED
4F14
4F14 CDDA1C                 CALL STACKA
4F17
4F17 EF                     DB CALC
4F18 56                     DB CHRS           ;CONVERT KEY VALUE TO STRING
4F19 33                     DB EXIT
4F1A
4F1A 180E                   JR GT4
4F1C
4F1C CD303B     GT2:        CALL NUMERIC
4F1F 3804                   JR C,GT3          ;JR IF '0' TO '9'
4F21
4F21 E6DF                   AND 0DFH          ;FORCE U.C
4F23 D607                   SUB 7
4F25
4F25 D630       GT3:        SUB 30H           ;'0'->0, 'A'->10D ETC.
4F27 CDDA1C                 CALL STACKA
4F2A
4F2A CD9904     GT4:        CALL NOISE
4F2D C30F2B                 JP ASSIGN         ;ASSSIGN STACKED VALUE TO VARIABLE
4F30
4F30 0030=      GETLN       EQU GT4+6-GETP2
4F30
4F00                        ORG $,INSTBUF
4F00
4F00 210100     DELPT2:     LD HL,1
4F03 CD514F                 CALL DELSR
4F06 DF                     RST 18H           ;CHAR AFTER 'DELETE'
4F07 FE8E                   CP TOTOK
4F09 280C                   JR Z,DL2
4F0B
4F0B CD8E3B                 CALL GIR2         ;EVAL NUM, GET TO HL IF RUNNING. SCF IF SYNTAX
4F0E D4514F                 CALL NC,DELSR     ;CALL IF RUNNING
4F11
4F11 DF                     RST 18H
4F12 FE8E                   CP TOTOK
4F14 C2290D                 JP NZ,NONSENSE
4F17
4F17 E7         DL2:        RST 20H           ;SKIP 'TO'
4F18 CD7A3A                 CALL CRCOLON
4F1B 21FFFE                 LD HL,0FEFFH      ;LAST LEGAL LINE NUMBER
4F1E C48E3B                 CALL NZ,GIR2      ;GET LAST LINE NUM
4F21
4F21 CDC63A                 CALL RUNFLG
4F24 3009                   JR NC,DL4          ;JR IF SYNTAX TIME
```

```
4F26
4F26 CD4D1A               CALL FNDLINE
4F29 2004                 JR NZ,DL4         ;IF NOT USED, HL IS START OF NEXT LINE
4F2B
4F2B CD4E1F               CALL NEXTONE      ;IF LINE IN USE, GET ADDR OF END+1 IN DE
4F2E EB                   EX DE,HL          ; THEN HL
4F2F
4F2F CDEF3F     DL4:      CALL CHKHL
4F32 DBFB                 IN A,(251)
4F34 57                   LD D,A
4F35 CD103B               CALL CHKENDCP
4F38
4F38 3A7F5A               LD A,(LAST)
4F3B 4F                   LD C,A
4F3C CDDF3F               CALL TSURPG       ;SWITCH IN START
4F3F 7A                   LD A,D            ;AHL=END ADDR
4F40 ED5B7D5A             LD DE,(FIRST)     ;CDE=START ADDR
4F44 CDE71F               CALL SUBAHLCDE
4F47 D8                   RET C             ;RET IF START HIGHER THAN END E.G. DELETE 2 TO 1
4F48
4F48 44                   LD B,H
4F49 4D                   LD C,L            ;ABC=LEN TO CLOSE
4F4A EB                   EX DE,HL          ;HL=START
4F4B CD531E               CALL RECL2BIG     ;CLOSE UP ABC AT (HL)
4F4E C3C933               JP GT4R           ;'GOTO' NEXT STATEMENT SO NXTLINE ETC. SET UP AGAIN
4F51
4F51 CD4D1A     DELSR:    CALL FNDLINE      ;DEFAULT LINE IS LINE 1 OR FIRST AFTER IT
4F54 CDEF3F               CALL CHKHL
4F57 227D5A               LD (FIRST),HL
4F5A DBFB                 IN A,(251)
4F5C 327F5A               LD (LAST),A
4F5F C3D73F     DELFIN:   JP SELCHADP
4F62
4F62 0062=      DELLN     EQU DELFIN+3-DELPT2
4F62
4F62                      ORG HDR
4B00
4B00 CD663A     KEYP2:    CALL SYNTAXA
4B03
4B03 3EFE                 LD A,0FEH
4B05 CDEC3E               CALL SBFSR2       ;COPY STRING TO BUFFER. LEN 0-511 OK
4B08 C8                   RET Z             ;RET IF LEN Z
4B09
4B09 2A455C               LD HL,(PPC)
4B0C E5                   PUSH HL
4B0D 3A475C               LD A,(SUBPPC)
4B10 F5                   PUSH AF           ;SAVE LINE/STAT FOR LATER 'GOTO'
4B11 D5                   PUSH DE           ;STRING START
4B12 C5                   PUSH BC           ;STRING LEN
4B13 CDA504               CALL CLEARSP      ;CLEAR ELINE
4B16 C1                   POP BC
4B17 C5                   PUSH BC
4B18 CD351F               CALL ADDRELN
4B1B CD1B1E               CALL MAKEROOM
4B1E EB                   EX DE,HL          ;DE PTS TO ROOM
4B1F C1                   POP BC
4B20 E1                   POP HL            ;STRING START
4B21 EDB0                 LDIR              ;COPY STRING TO ELINE (INDIRECT METHOD MEANS
4B23                                        ;EVEN DIRECT KEYIN OF LITERAL STRING WORKS)
4B23 CD2E20               CALL SETESP       ;ERRORS WILL RETURN TO THIS ROUTINE NOW.
4B26                                        ;OLD ERRSP IS ON STACK
4B26 CD7238               CALL TOKMAIN      ;TOKENIZE ELINE
4B29 CD130D               CALL LINESCAN     ;CHECK SYNTAX
4B2C 3A3A5C               LD A,(ERRNR)
4B2F A7                   AND A
4B30 2015                 JR NZ,KI3         ;JR IF ERROR OCCURRED
4B32
4B32 213B5C               LD HL,FLAGS
4B35 CBFE                 SET 7,(HL)        ;'RUNNING'
4B37 CD7910               CALL EVALLINO     ;CY IF IOOR, Z IF 0
4B3A DA290D               JP C,NONSENSE
4B3D
4B3D 2005                 JR NZ,KI2         ;JR IF THERE WAS A LINE NUMBER
4B3F
4B3F 0C                   INC C             ;STAT 1
4B40 CD2B0D               CALL LOOPEL       ;RUN ELINE
4B43 37                   SCF
4B44
4B44 D4A010     KI2:      CALL NC,INSERTLN
4B47
4B47 E1         KI3:      POP HL            ;OLD ERRSP
4B48 CD6804               CALL RESESP       ;RESET ERRSP, RESPOND TO ANY ERRORS
4B4B CDA504               CALL CLEARSP      ;CLEAR ELINE
4B4E C3D033     KEYFIN:   JP GT4P           ;SET UP SO GOTO HAPPENS. MEANS LINE CAN BE KEYED
4B51                                        ;IN BEFORE LINE BEING EXECUTED.
4B51
4B51 0051=      KEYLN     EQU KEYFIN+3-KEYP2
4B51
4B51                      ORG INSTBUF
4F00
4F00 CD7A3A     POPP2:    CALL CRCOLON
4F03
4F03 C4EB38               CALL NZ,VNUMV        ;EVAL NUM VARIABLE, SET NZ
4F06
4F06 08         POP1:     EX AF,AF'         ;Z IF NO VAR
4F07 CD153B               CALL CHKEND
4F0A
4F0A 08                   EX AF,AF'
4F0B F5                   PUSH AF
4F0C 2AC45B               LD HL,(BSTKEND)
4F0F 7E                   LD A,(HL)
4F10 3C                   INC A
4F11 2002                 JR NZ,POP2        ;JR IF NOT THE STACK TERMINATOR
```

```
4F13
4F13 CF                    RST 08H
4F14 0B                    DB 11             ;'No POP data'
4F15
4F15 CD281A     POP2:      CALL RETLOOP2     ;ACCEPT ANY TYPE OF RET ADDR - DO/GOSUB/PROC
4F18                                         ;HL=LINE, A=TYPE/PAGE
4F18 47                    LD B,A
4F19 F1                    POP AF
4F1A C5                    PUSH BC           ;TYPE/PAGE
4F1B 2814                  JR Z,POP4         ;RET IF NO VARIABLE TO POP INTO
4F1D
4F1D 24                    INC H
4F1E 25                    DEC H
4F1F 54                    LD D,H
4F20 5C                    LD E,H
4F21 2807                  JR Z,POP3         ;JR IF ELINE - USE LINE OF 0
4F23
4F23 78                    LD A,B
4F24 CDDF3F                CALL TSURPG
4F27 56                    LD D,(HL)
4F28 23                    INC HL
4F29 5E                    LD E,(HL)         ;DE=LINE NUMBER
4F2A
4F2A EB         POP3:      EX DE,HL
4F2B CDD61C                CALL STACKHL
4F2E CD0F2B                CALL ASSIGN       ;FPCS TO VAR
4F31
4F31 F1         POP4:      POP AF            ;TYPE/PAGE
4F32 E6E0                  AND 0E0H
4F34 FE40                  CP 40H
4F36 CA0A37                JP Z,DELOCAL    ;IF 'PROC' TYPE, CLEAR LOCAL VARS TOO
4F39
4F39 C9         POP5:      RET
4F3A
4F3A 003A=      POPLN      EQU POP5-POPP2+1
4F3A
4F3A                       ORG INSTBUF
4F00
4F00 CDC63A     INPP2:     CALL RUNFLG
4F03 DCB506                CALL C,CLSLOWER     ;CLS AND SELECT CHANNEL 'K' IF RUNNING
4F06 CD9006                CALL SPACAN       ;NO INDENT **
4F09 3C                    INC A             ;A=1
4F0A 323C5C                LD (TVFLAG),A
4F0D CD264F                CALL INPSL
4F10 AF                    XOR A
4F11 32715C                LD (FLAGX),A      ;'NOT INPUT LINE' SO LISTING OK
4F14 CD103B                CALL CHKENDCP
4F17
4F17 CDB506                CALL CLSLOWER
4F1A 3A6D5A                LD A,(SPOSNU+1)
4F1D 213A5A                LD HL,UWTOP
4F20 96                    SUB (HL)
4F21 3C                    INC A
4F22 328C5C                LD (SCRCT),A      ;EVERYTHING ON SCREEN DOWN TO SPOSN HAS BEEN SEEN,
4F25                                         ;SO WE CAN SCROLL THOSE LINES OFF BEFORE 'SCROLL?'
4F25 C9                    RET
4F26
4F26 DF         INPSL:     RST 18H
4F27 CDED07                CALL PRTERM
4F2A 2809                  JR Z,IP2CR        ;JR IF CR/COLON/CLOSE BRACK.
4F2C
4F2C CDD807                CALL PRSEPR
4F2F C8                    RET Z             ;RET IF TERMINATOR FOUND AFTER ;/,/'
4F30
4F30 DC3D4F                CALL C,IPITEM     ;CALL INPUT ITEM SR IF NON-SEPARATOR FOUND
4F33 18F1                  JR INPSL
4F35
4F35 3A735A     IP2CR:     LD A,(DEVICE)
4F38 A7                    AND A
4F39 CACF07                JP Z,RUNCR        ;CR IF RUNNING IF UPPER SCREEN IN USE
4F3C
4F3C C9                    RET
4F3D
4F3D FE28       IPITEM:    CP "("
4F3F 2008                  JR NZ,INP2        ;JR IF NOT AN EMBEDDED PRINT ITEM
4F41
4F41 E7                    RST 20H           ;SKIP '('
4F42 CDB707                CALL PRINT2       ;HANDLE PRINT ITEM
4F45 DF                    RST 18H
4F46 C3953A                JP INSISCBRK      ;CHECK/SKIP ')'
4F49
4F49 FE8C       INP2:      CP LINETOK
4F4B 2810                  JR Z,INP4
4F4D
4F4D CD1E3B                CALL ALPHA
4F50 D2F807                JP NC,PRITEM      ;JR IF NOT A LETTER
4F53
4F53 CDF72C                CALL SYNTAX1      ;EXITS WITH DE=FLAGX
4F56 21715C                LD HL,FLAGX
4F59 CBBE                  RES 7,(HL)        ;'NOT INPUT LINE'
4F5B 180E                  JR INP5
4F5D
4F5D CDF62C     INP4:      CALL SSYNTAX1     ;SKIP 'LINE', EVAL VAR
4F60 CDC63A                CALL RUNFLG
4F63 FA290D                JP M,NONSENSE     ;ERROR IF NUMERIC VARIABLE
4F66
4F66 21715C                LD HL,FLAGX
4F69 CBFE                  SET 7,(HL)        ;"USING INPUT LINE"
4F6B
4F6B CD153B     INP5:      CALL ABORTER
4F6E
4F6E E5                    PUSH HL
4F6F CD891D                CALL SETWORK      ;CLEAR WORKSPACE
```

```
4F72 E1                  POP HL             ;FLAGX
4F73 CBEE                SET 5,(HL)         ;'INPUT MODE'
4F75 CBF6                SET 6,(HL)         ;'NUMERIC RESULT'
4F77 010100              LD BC,1
4F7A CDC63A              CALL RUNFLG
4F7D FA884F              JP M,INP6          ;JP IF NUMERIC - USE 1 SPACE
4F80
4F80 CBB6                RES 6,(HL)         ;STRING RESULT
4F82 CB7E                BIT 7,(HL)
4F84 2002                JR NZ,INP6         ;JR IF INPUT LINE - USE 1 SPACE
4F86
4F86 0E03                LD C,3
4F88
4F88 CD801E     INP6:    CALL WKROOM
4F8B 360D                LD (HL),0DH        ;LAST LOCN
4F8D 0D                  DEC C
4F8E 2805                JR Z,INP7          ;JR IF JUST ONE SPACE
4F90
4F90 2B                  DEC HL
4F91 3E22                LD A,22H
4F93 77                  LD (HL),A          ;SECOND LOCN
4F94 12                  LD (DE),A          ;FIRST LOCN
4F95
4F95 229A5A     INP7:    LD (KCUR),HL       ;KCUR PTS TO CR, OR 2ND QUOTE
4F98 DBFB                IN A,(URPORT)
4F9A 32995A              LD (KCURP),A
4F9D 3A715C              LD A,(FLAGX)
4FA0 17                  RLA
4FA1 3832                JR C,INP9          ;JR IF INPUT LINE - NO SYNTAX CHECK ON INPUT
4FA3
4FA3 3A965A              LD A,(CHADP)
4FA6 F5                  PUSH AF
4FA7 2A975A              LD HL,(CHAD)
4FAA E5                  PUSH HL
4FAB 2A3D5C              LD HL,(ERRSP)
4FAE E5                  PUSH HL
4FAF
4FAF 21AF4F     INPERR:  LD HL,INPERR
4FB2 E5                  PUSH HL
4FB3 AF                  XOR A
4FB4 323A5C              LD (ERRNR),A       ;'NO ERROR'
4FB7 CDBD05              CALL KSCHK
4FBA 2004                JR NZ,INP8         ;JR IF NOT 'K' OR 'S' CHANNEL
4FBC
4FBC ED733D5C            LD (ERRSP),SP      ;ONLY SET ERRSP FOR K/S
4FC0
4FC0 CD2B1F     INP8:    CALL ADDRWK        ;PT TO WORKSPACE
4FC3 CD6910              CALL REMOVEFP
4FC6 CDA602              CALL EDITOR
4FC9 CD7238              CALL TOKMAIN
4FCC 213B5C              LD HL,FLAGS
4FCF CBBE                RES 7,(HL)         ;SYNTAX TIME
4FD1 CD2E50              CALL INPAS
4FD4 A7                  AND A              ;SKIP NEXT INSTR.
4FD5
4FD5 DC9F02     INP9:    CALL C,EDCX        ;EDITOR
4FD8
4FD8 CDBD05              CALL KSCHK
4FDB 200F                JR NZ,INPA         ;JR IF NOT 'K' OR 'S' CHANNEL
4FDD
4FDD 329B5A              LD (KCUR+1),A      ;NO CURSOR WANTED
4FE0 CD7605              CALL EDPRT         ;PRINT INPUT LINE
4FE3 2A825C              LD HL,(OLDPOS)
4FE6 0162DC              LD BC,POSTORE
4FE9 CDEA01              CALL R1ONCLBC      ;SCREEN POSN=PAST END OF PRINTED INPUT LINE
4FEC
4FEC 21715C     INPA:    LD HL,FLAGX
4FEF 7E                  LD A,(HL)
4FF0 CBBE                RES 7,(HL)         ;'NOT INPUT LINE'
4FF2 CBAE                RES 5,(HL)         ;'NOT INPUT MODE'
4FF4 17                  RLA
4FF5 3015                JR NC,INPC         ;JR IF NOT INPUT LINE
4FF7
4FF7 CD2B1F              CALL ADDRWK
4FFA 54                  LD D,H
4FFB 5D                  LD E,L             ;DE=START
4FFC 01FFFF              LD BC,0FFFFH
4FFF
4FFF 7E          INPBL:  LD A,(HL)
5000 23                  INC HL
5001 03                  INC BC
5002 FE0D                CP 0DH
5004 20F9                JR NZ,INPBL        ;COUNT LEN TILL 0D TERMINATOR
5006
5006 CDE71C              CALL STKSTOREP
5009 C30F2B              JP ASSIGN
500C
500C F1          INPC:   POP AF             ;JUNK INPERR
500D E1                  POP HL
500E 223D5C              LD (ERRSP),HL
5011 E1                  POP HL
5012 22A95A              LD (PRPTR),HL
5015 F1                  POP AF
5016 32A85A              LD (PRPTRP),A      ;ORIG CHAD IN AUTO-ADJUST SYS VAR
5019 213B5C              LD HL,FLAGS
501C CBFE                SET 7,(HL)
501E CD2E50              CALL INPAS
5021 3AA85A              LD A,(PRPTRP)
5024 CDCE3F              CALL SETCHADP
5027 2AA95A              LD HL,(PRPTR)
502A 22975A              LD (CHAD),HL
502D C9                  RET
502E
```

```
502E 2A915A     INPAS:      LD HL,(WORKSP)
5031 22975A                 LD (CHAD),HL
5034 3A905A                 LD A,(WORKSPP)
5037 CDCE3F                 CALL SETCHADP
503A DF                     RST 18H
503B FEB1                   CP 0B1H             ;STOPTOK
503D 2006                   JR NZ,INPA2
503F
503F CDC63A                 CALL RUNFLG
5042 D0                     RET NC
5043
5043 CF                     RST 08H
5044 11                     DB 17               ;'STOP in INPUT'
5045
5045 3A715C     INPA2:      LD A,(FLAGX)
5048 CDFD2A                 CALL VALFET2
504B DF                     RST 18H
504C FE0D                   CP 0DH
504E C8                     RET Z
504F
504F CF                     RST 08H
5050 1D         INPFIN:     DB 29               ;'NONSENSE'
5051
5051 0151=      INPLN       EQU INPFIN+1-INPP2
5051
5051                                            ;INPUT
5051                        INCLUDE TMISCX2.SAM   ;DEF KEYCODE, READ, DEF FN, TOKEN.
5051            ;MISCX2.SAM
5051
5051                        ORG INSTBUF
4F00
4F00 CDE43A     DKP2:       CALL EXPT1NUM       ;NUMBER OF KEY CODE TO DEFINE
4F03                                            ;(KEYBOARD HAS 69 KEYS WITH 3 SHIFTS (CAPS, SYM,
4F03                                            ;CONTROL) GIVING 276 BYTES IN KEY MAP, ANY OF WHICH
4F03                                            ;CAN BE PROGRAMMED TO GIVE ANY KEY CODE. KEY CODES
4F03                                            ;ABOVE 168? CAN BE DEFINED
4F03 CD6C3A                 CALL COMMASC        ;SEE IF COMMA OR SEMI-COLON
4F06 2817                   JR Z,DFK4           ;JR IF STRING DEFINITION SHOULD FOLLOW
4F08
4F08 CDC63A                 CALL RUNFLG
4F0B D26410                 JP NC,DFKNL         ;JP IF SYNTAX TIME - CHECK, STRIP FP FORMS
4F0E
4F0E EB                     EX DE,HL            ;DE=CHAD
4F0F 2A9D5A                 LD HL,(NXTLINE)
4F12 37                     SCF
4F13 ED52                   SBC HL,DE
4F15 44                     LD B,H
4F16 4D                     LD C,L              ;ASSUMING NXTLINE HAS SAME PAGE PART AS CHAD,
4F17                                            ;BC=LEN OF REST OF LINE
4F17 0B                     DEC BC              ;DEC BECAUSE WE DON'T USE ':' SEPARATOR
4F18 13                     INC DE              ;SKIP ':'
4F19 21120E                 LD HL,LINEEND
4F1C E3                     EX (SP),HL          ;JUNK NEXT STAT, STACK LINEEND SO WE DON'T
4F1D 1806                   JR DFK5             ;EXECUTE THE REST OF THIS LINE
4F1F
4F1F CD653A     DFK4:       CALL SSYNTAXA       ;DEFINITION STRING
4F22
4F22 CDDC3F                 CALL GETSTRING      ;BC=LEN, DE=ST, PAGED IN
4F25
4F25 D5         DFK5:       PUSH DE
4F26 C5                     PUSH BC
4F27 CD331D                 CALL GETBYTE        ;KEY CODE
4F2A 3C                     INC A               ;255->0
4F2B FEC1                   CP 193              ;ONLY KEY CODES 192-254 ARE DEFINABLE
4F2D DA391D                 JP C,IOORERR
4F30
4F30 3D                     DEC A
4F31 F5                     PUSH AF
4F32 CDAA05                 CALL FNDKYD         ;FIND ANY EXISTING KEY DATA
4F35 381A                   JR C,DFK55          ;JR IF DOESNT EXIST, ELSE CLOSE IT UP
4F37
4F37            ;DEF KEY CLOSE
4F37            ;CLOSES BC+3 BYTES AT HL-3 (I.E. ENTRY WITH HL PTING TO TEXT, BC=TEXT LEN,
4F37            ;ERASES ENTIRE DEFINITION)
4F37
4F37 03                     INC BC
4F38 03                     INC BC
4F39 03                     INC BC
4F3A E5                     PUSH HL             ;DEF TEXT START
4F3B 09                     ADD HL,BC           ;PT TO PAST END OF DEFINITION,+3
4F3C E5                     PUSH HL
4F3D CDA805                 CALL DKTR           ;GET HL PTING TO TERMINATOR+3
4F40 D1                     POP DE
4F41 A7                     AND A
4F42 ED52                   SBC HL,DE
4F44 23                     INC HL
4F45 44                     LD B,H
4F46 4D                     LD C,L              ;BC=BYTES TO MOVE (PAST END TO TERMINATOR, PLUS 1)
4F47 EB                     EX DE,HL
4F48 2B                     DEC HL
4F49 2B                     DEC HL
4F4A 2B                     DEC HL              ;PT TO PAST DEF END
4F4B D1                     POP DE
4F4C 1B                     DEC DE
4F4D 1B                     DEC DE
4F4E 1B                     DEC DE              ;PT TO DEF START
4F4F EDB0                   LDIR
4F51
4F51 F1         DFK55:      POP AF              ;KEY CODE
4F52 C1                     POP BC              ;DEF$ LEN
4F53 F5                     PUSH AF
4F54 78                     LD A,B
4F55 FEFF                   CP 0FFH
```

```
4F57 2801                  JR Z,DFK6         ;ABORT IF WAS E.G. DEF KEY 1 (CR). LEN IS FFFFH
4F59
4F59 B1                    OR C              ;ALSO ABORT IF LEN 0
4F5A
4F5A CAD304      DFK6:     JP Z,PPRET        ;POP POP RET - JUNK KEY, SRC, ABORT
4F5D
4F5D             ;DEF KEY OPEN
4F5D             ;ENTRY: BC=TEXT LEN (BC+3 IS OPENED TO ALLOW FOR KEY CODE AND LEN)
4F5D             ;EXIT: DE PTS TO SPACE, BC IS UNCHANGED
4F5D
4F5D C5                    PUSH BC
4F5E CDA805                CALL DKTR         ;HL=TERMINATOR ADDR+3
4F61 C1                    POP BC
4F62 E5                    PUSH HL
4F63 09                    ADD HL,BC         ;HL=NEW POSN FOR TERMINATOR. NC
4F64 EB                    EX DE,HL
4F65 2AD05B                LD HL,(DKLIM)
4F68 ED52                  SBC HL,DE
4F6A DA0730                JP C,TMDERR       ;ERROR IF TERMINATOR PAST LIMIT
4F6D                                         ;ALLOWS TERMINATOR TO BE AT LIMIT...
4F6D EB                    EX DE,HL
4F6E 36FF                  LD (HL),0FFH      ;NEW TERMINATOR (OLD WILL BE OVER-WRITTEN)
4F70 D1                    POP DE
4F71 1B                    DEC DE
4F72 1B                    DEC DE
4F73 1B                    DEC DE            ;PT TO OLD TERMINATOR
4F74
4F74 F1                    POP AF            ;KEY CODE
4F75 E1                    POP HL            ;SRC
4F76 12                    LD (DE),A
4F77 13                    INC DE
4F78 79                    LD A,C
4F79 12                    LD (DE),A
4F7A 13                    INC DE
4F7B 78                    LD A,B
4F7C 12                    LD (DE),A         ;LEN IN PLACE
4F7D 13                    INC DE
4F7E EDB0                  LDIR
4F80 C9          DKFIN:    RET
4F81
4F81 0081=       DKLN      EQU DKFIN+1-DKP2
4F81
4F81
4F81 0000=       RDLN      EQU 0   ;RDFIN+3-RDP2
4F81
4F81             ;DEF FN COMMAND (OR MARKER - DOES NOTHING EXCEPT CHECK SYNTAX)
4F81
4F81                       ORG INSTBUF
4F00
4F00 CDC63A      DFNP2:    CALL RUNFLG
4F03 DAF210                JP C,SKIPCSTAT    ;SKIP STATEMENT IF RUNNING
4F06
4F06 DF                    RST 18H
4F07 CD2732                CALL FNNAME
4F0A F5                    PUSH AF           ;A=1 IF STRING DEF FN
4F0B 22975A                LD (CHAD),HL
4F0E DF                    RST 18H
4F0F FE28                  CP "("
4F11 2021                  JR NZ,DFN5
4F13
4F13 E7                    RST 20H           ;SKIP '('
4F14 FE29                  CP ")"
4F16 281B                  JR Z,DFN4
4F18
4F18 CDEC3A      DFNPL:    CALL GETALPH      ;INSIST ON A LETTER
4F1B 23                    INC HL            ;SKIP VAR LETTER
4F1C 7E                    LD A,(HL)
4F1D FE24                  CP "$"
4F1F 2001                  JR NZ,DFN3
4F21
4F21 23                    INC HL            ;SKIP '$'
4F22
4F22 CD6433      DFN3:     CALL MAKESIX
4F25 ED53975A              LD (CHAD),DE      ;PT TO LAST OF 5-BYTES
4F29 E7                    RST 20H
4F2A FE29                  CP ")"
4F2C 2805                  JR Z,DFN4
4F2E
4F2E CD853A                CALL INSISCOMA
4F31 18E5                  JR DFNPL
4F33
4F33 E7          DFN4:     RST 20H
4F34
4F34 FE3D        DFN5:     CP "="
4F36 200A                  JR NZ,DFNNS
4F38
4F38 CD073B                CALL SEXPTEXPR
4F3B C1                    POP BC
4F3C 2802                  JR Z,DFN6         ;JR IF EXPR ASSIGNED TO DEF FN IS STRING
4F3E
4F3E 05                    DEC B
4F3F C0                    RET NZ            ;RET IF B WAS NOT 1
4F40
4F40 05          DFN6:     DEC B             ;ERROR IF B<>1 (NAME<>STRING TYPE)
4F41 C8                    RET Z
4F42
4F42 CF          DFNNS:    RST 08H
4F43 1D                    DM 29             ;'Syntax error'
4F44
4F44 0044=       DFNLN     EQU DFNNS+2-DFNP2
4F44                       ORG CDBUFF+80H
4D80
4D80 D9          TOKPT2:   EXX
```

```
4D81 EB                  EX DE,HL          ;E-LINE START OR WKSPACE START, OR OTHER,  TO HL
4D82
4D82 7E          TOKRST: LD A,(HL)         ;JR TO PT CHAD TO LINE START
4D83 1813                JR TOKQUEN
4D85
4D85 E1          LOOKNA: POP HL
4D86 7E                  LD A,(HL)
4D87 23                  INC HL
4D88 FE41                CP "A"
4D8A 380C                JR C,TOKQUEN      ;JR IF WE JUST CONSIDERED '>' OR '<'
4D8C
4D8C FE                  DB 0FEH           ;'JR+1'
4D8D
4D8D 23          LKNONALP: INC HL
4D8E 7E                  LD A,(HL)
4D8F CD1E3B              CALL ALPHA
4D92 38F9                JR C,LKNONALP     ;LOOP WHILE LETTER
4D94
4D94 FE5F                CP "_"            ;CHECK FOR LETTER, UNDERLINE OR DOLLAR
4D96
4D96 28F5                JR Z,LKNONALP     ;LOOP WHILE UNDERLINE
4D98
4D98 22975A      TOKQUEN: LD (CHAD),HL      ;RESET CHAD
4D9B FE                  DB 0FEH           ;'JR+1'
4D9C
4D9C E7          TOKMLP: RST 20H
4D9D EB                  EX DE,HL
4D9E FE0D                CP 0DH
4DA0 C8                  RET Z             ;RET IF END MARKER FOUND
4DA1
4DA1 CD1E3B              CALL ALPHA
4DA4 3820                JR C,POSFIRST     ;JR IF A POSSIBLE FIRST LETTER FOUND
4DA6
4DA6 FE3C                CP "<"
4DA8 281C                JR Z,POSFIRST     ;COULD BE '<=' OR '<>'
4DAA
4DAA FE3E                CP ">"
4DAC 2818                JR Z,POSFIRST     ;COULD BE '>='
4DAE
4DAE 13                  INC DE
4DAF FEFF                CP 0FFH
4DB1 280E                JR Z,FNTS         ;JR WITH (DE)=FN LETTER IF FN LDR
4DB3
4DB3 1B                  DEC DE
4DB4 FE22                CP 22H            ;QUOTE
4DB6 20E4                JR NZ,TOKMLP
4DB8
4DB8 13          QUOTELP: INC DE
4DB9 1A                  LD A,(DE)
4DBA FE0D                CP 0DH
4DBC C8                  RET Z
4DBD
4DBD FE22                CP 22H
4DBF 20F7                JR NZ,QUOTELP
4DC1
4DC1 13          FNTS:   INC DE
4DC2 EB                  EX DE,HL
4DC3 7E                  LD A,(HL)
4DC4 18D2                JR TOKQUEN
4DC6
4DC6 D5          POSFIRST: PUSH DE
4DC7 EB                  EX DE,HL
4DC8 11234E              LD DE,TOKFIN+3    ;END OF THIS ROUTINE, IN CDBUFF
4DCB 010F00              LD BC,15          ;MAX LEN TO TOKENIZE
4DCE D5                  PUSH DE
4DCF EDB0                LDIR
4DD1 D1                  POP DE
4DD2 21C7F8              LD HL,KEYWTAB-1
4DD5 3EC5                LD A,KEYWNO+1     ;WORDS TO CHECK, PLUS 1
4DD7 CD8A01              CALL JGTTOK       ;GET A=1 TO KEYWNUMBER, IF MATCHED
4DDA 200A                JR NZ,YGOTM
4DDC
4DDC 2AFA5A              LD HL,(MTOKV)
4DDF 24                  INC H
4DE0 25                  DEC H
4DE1 C40500              CALL NZ,HLJUMP
4DE4
4DE4 289F                JR Z,LOOKNA       ;JR IF NO MATCH FOUND, OR MTOKV ZERO
4DE6
4DE6 EB          YGOTM:  EX DE,HL          ;ELSE HL=WORD START, DE=PAST END (IN BUFFER)
4DE7 A7                  AND A
4DE8 ED52                SBC HL,DE         ;HL=LEN
4DEA D1                  POP DE            ;ELINE PTR
4DEB 19                  ADD HL,DE
4DEC EB                  EX DE,HL          ;HL PTS TO START, DE TO PAST END, IN ELINE
4DED
4DED FE4A                CP 4AH
4DEF 3007                JR NC,TOK42       ;JR IF IN RANGE OF CMDS
4DF1
4DF1 C63A                ADD A,3AH         ;ELSE GET FN CODE 3BH-83H
4DF3 36FF                LD (HL),0FFH      ;OVERWRITE FIRST LETTER WITH FN LEADER
4DF5 23                  INC HL
4DF6 1811                JR TOK55
4DF8
4DF8 C63B        TOK42:  ADD A,3BH         ;CONVERT LIST ENTRY TO TOKEN CODE 85H-
4DFA FEFF                CP 0FFH           ;'INK'
4DFC 2002                JR NZ,TOK43
4DFE
4DFE 3EA1                LD A,0A1H         ;PENTOK - ALLOW 'PEN' TO BE ENTERED AS 'INK'
4E00
4E00 2B          TOK43:  DEC HL            ;PT TO BEFORE WORD IN INPUT
4E01 08                  EX AF,AF'         ;SAVE TOKEN
4E02 7E                  LD A,(HL)
```

```
4E03 FE20                  CP " "
4E05 2801                  JR Z,TOK5          ;AVOID INC IF THERE IS A LEADING SPACE - SO IT
4E07                                          ;WILL BE OVER-WRITTEN BY TOKEN
4E07 23                    INC HL             ;PT TO FIRST LETTER
4E08
4E08 08          TOK5:     EX AF,AF'          ;RESTORE TOKEN
4E09
4E09 77          TOK55:    LD (HL),A          ;PLACE AT START OF SPELLED-OUT FORM
4E0A 23                    INC HL             ;HL PTS TO FIRST BYTE TO DELETE
4E0B EB                    EX DE,HL
4E0C 7E                    LD A,(HL)
4E0D FE20                  CP " "
4E0F 2001                  JR NZ,TOK6         ;JR IF NO TRAILING SPACE, ELSE INC HL TO INCLUDE
4E11                                          ;SPACE IN 'CLOSED UP' REGION
4E11 23                    INC HL
4E12
4E12 D5          TOK6:     PUSH DE
4E13 CD4F1E                CALL RECLAIM1      ;CLOSE UP (DE) <- (HL)
4E16 D1                    POP DE             ;KEEP LOOKING FROM HL ONWARDS
4E17 62                    LD H,D
4E18 6B                    LD L,E
4E19 1B                    DEC DE
4E1A 1A                    LD A,(DE)
4E1B FEB7                  CP 0B7H            ;REMTOK. DON'T TOKENIZE REM STATEMENTS
4E1D C2824D                JP NZ,TOKRST       ;KEEP TOKENISING FROM HL ONWARDS
4E20
4E20 C9          TOKFIN:   RET
4E21
4E21 00A1=       TOKLN     EQU TOKFIN+1-TOKPT2
4E21
4E21                       ORG INSTBUF
4F00
4F00 CD891D      MEPRO2:   CALL SETWORK       ;CLEAR WORKSP IN CASE E.G. MERGE A$+B$
4F03 010100                LD BC,1
4F06 CD801E                CALL WKROOM        ;OPEN 1 BYTE IN WORKROOM (NOTE: ENSURES WKEND MOVED
4F09                                          ;BY MKRBIG)
4F09 36FF                  LD (HL),0FFH       ;TERMINATOR
4F0B E5                    PUSH HL
4F0C CD3B20                CALL RDLLEN        ;CDE=FILE LEN TO LOAD
4F0F E1                    POP HL
4F10 C5                    PUSH BC
4F11 D5                    PUSH DE
4F12 79                    LD A,C
4F13 42                    LD B,D
4F14 4B                    LD C,E
4F15 CD1C1E                CALL MKRBIG        ;OPEN ABC BYTES
4F18 D1                    POP DE
4F19 C1                    POP BC
4F1A 37                    SCF                ;'LOAD' NOT 'VERIFY'
4F1B CD7801                CALL JLDVD         ;LOAD CDE TO (HL)
4F1E CD9D4F                CALL MBASLNS
4F21 CDDC4F                CALL MNUMS
4F24 21604B                LD HL,HDL+16
4F27 CD3F20                CALL RDTHREE       ;CDE=LEN OF PROG ALONE
4F2A C5                    PUSH BC
4F2B D5                    PUSH DE
4F2C 2E66                  LD L,>(HDL+22)
4F2E CD3F20                CALL RDTHREE       ;CDE=LEN OF PROG+NVARS+GAP
4F31 79                    LD A,C
4F32 EB                    EX DE,HL           ;AHL=DITTO
4F33 D1                    POP DE
4F34 C1                    POP BC
4F35 CDE71F                CALL SUBAHLCDE     ;AHL=LEN OF NVARS+GAP
4F38 010100                LD BC,1
4F3B CDCC1F                CALL ADDAHLBC      ;ALLOW FOR BASIC PROG TERMINATOR
4F3E F5                    PUSH AF
4F3F E5                    PUSH HL
4F40 CD2B1F                CALL ADDRWK
4F43 C1                    POP BC
4F44 F1                    POP AF
4F45 CD531E                CALL RECL2BIG      ;DELETE LOADED NVARS+GAP
4F48
4F48             ;MERGE STRINGS/ARRAYS (PART OF PROGRAM FILE)
4F48             ;ENTRY: DEST PTS TO LOADED DATA
4F48
4F48 CD2B1F      MSTAR:    CALL ADDRWK        ;PT TO SRC (TLBYTE OR VAR, OR FF)
4F4B 7E                    LD A,(HL)
4F4C 3C                    INC A
4F4D CAC933                JP Z,GT4R          ;END IF ALL DONE - TERMINATOR HIT
4F50
4F50 113F51                LD DE,TLBYTE
4F53 E60F                  AND 0FH
4F55 4F                    LD C,A
4F56 0600                  LD B,0
4F58 7E                    LD A,(HL)
4F59 EDB0                  LDIR               ;COPY TLBYTE AND NAME TO BUFFER
4F5B CD3114                CALL STARYLK2      ;CALL WITH A=DESIRED T/L
4F5E C45B2C                CALL NZ,ASDEL2     ;DELETE STRING/ARRAY AT (STRLOCN) IF IT WAS FOUND
4F61
4F61 CD2B1F                CALL ADDRWK        ;SRC
4F64 010B00                LD BC,11
4F67 09                    ADD HL,BC          ;PT TO LEN DATA
4F68 7E                    LD A,(HL)
4F69 23                    INC HL
4F6A 5E                    LD E,(HL)
4F6B 23                    INC HL
4F6C 56                    LD D,(HL)
4F6D EB                    EX DE,HL
4F6E 0E0E                  LD C,14
4F70 CDCC1F                CALL ADDAHLBC      ;AHL=TOTAL LEN OF STR/ARRAY (PAGEFORM)
4F73 F5                    PUSH AF
4F74 E5                    PUSH HL
4F75 CDB61F                CALL ADDRELND
```

```
4F78 C1                    POP BC
4F79 F1                    POP AF
4F7A F5                    PUSH AF
4F7B C5                    PUSH BC
4F7C CD1C1E                CALL MKRBIG       ;MAKE ABC BYTES AT HL FOR STR/ARRAY
4F7F EB                    EX DE,HL
4F80 DBFB                  IN A,(251)
4F82 4F                    LD C,A            ;ROOM IS AT CDE
4F83 E1                    POP HL
4F84 CBBC                  RES 7,H
4F86 22845B                LD (MODCOUNT),HL
4F89 F1                    POP AF
4F8A 32835B                LD (PAGCOUNT),A
4F8D CD2B1F                CALL ADDRWK       ;PT TO SRC
4F90 DBFB                  IN A,(251)        ;AHL=SRC, CDE=DEST
4F92 CD5E2A                CALL FARLDIR      ;COPY TO DEST
4F95 CD2B1F                CALL ADDRWK
4F98 CD5E2C                CALL ASDEL3       ;DELETE SRC
4F9B 18AB                  JR MSTAR
4F9D
4F9D             ;MERGE BASIC LINES
4F9D
4F9D CD2B1F      MBASLNS:  CALL ADDRWK       ;PT TO LOADED BLOCK (SRC)
4FA0
4FA0 46          MPRG1:    LD B,(HL)
4FA1 23                    INC HL
4FA2 78                    LD A,B
4FA3 3C                    INC A
4FA4 C8                    RET Z             ;RET IF ALL LINES DONE
4FA5
4FA5 4E                    LD C,(HL)
4FA6 CD4F1A                CALL FNDLP        ;FIND LINE BC, STARTING FROM PROG
4FA9 E5                    PUSH HL
4FAA DBFB                  IN A,(251)
4FAC F5                    PUSH AF           ;DESTINATION VARS
4FAD CC4A1E                CALL Z,NORECL     ;RECLAIM IF IT EXISTS
4FB0
4FB0 CD2B1F                CALL ADDRWK       ;PT TO SRC
4FB3 23                    INC HL
4FB4 23                    INC HL
4FB5 4E                    LD C,(HL)
4FB6 23                    INC HL
4FB7 46                    LD B,(HL)
4FB8 03                    INC BC
4FB9 03                    INC BC
4FBA 03                    INC BC
4FBB 03                    INC BC            ;BC=TOTAL SRC LINE LEN
4FBC F1                    POP AF
4FBD E1                    POP HL
4FBE E5                    PUSH HL
4FBF F5                    PUSH AF
4FC0 C5                    PUSH BC
4FC1 CDDF3F                CALL TSURPG
4FC4 CD1B1E                CALL MAKEROOM     ;MAKE ROOM FOR LINE
4FC7 CD2B1F                CALL ADDRWK       ;AHL=SRC
4FCA C1                    POP BC            ;LEN
4FCB CD791F                CALL SPLITBC
4FCE C1                    POP BC
4FCF 48                    LD C,B
4FD0 D1                    POP DE            ;CDE=DEST
4FD1 CD5E2A                CALL FARLDIR      ;COPY TO PROGRAM
4FD4 CD2B1F                CALL ADDRWK       ;PT TO SRC
4FD7 CD4A1E                CALL NORECL       ;DELETE LINE FROM WORKSP
4FDA 18C4                  JR MPRG1
4FDC
4FDC             ;MERGE NUMBERS - WORKS WITH 16K OR LESS OF NUMERIC VARS.
4FDC
4FDC 3E61        MNUMS:    LD A,"a"
4FDE
4FDE 324051      MNUML:    LD (FIRLET),A
4FE1 5E                    LD E,(HL)
4FE2 23                    INC HL
4FE3 56                    LD D,(HL)
4FE4 14                    INC D
4FE5 C4F24F                CALL NZ,MNUMSR    ;CALL IF LETTER LIST HAS BEEN USED
4FE8
4FE8 23                    INC HL
4FE9 3A4051                LD A,(FIRLET)
4FEC 3C                    INC A
4FED FE7B                  CP "z"+1
4FEF 38ED                  JR C,MNUML
4FF1
4FF1 C9                    RET
4FF2
4FF2 E5          MNUMSR:   PUSH HL           ;SRC PTRS LIST POSN
4FF3
4FF3 15          MNSRL:    DEC D
4FF4 19                    ADD HL,DE
4FF5 7E                    LD A,(HL)
4FF6 E6A0                  AND 0A0H          ;MASK ALL EXCEPT 'HIDDEN' AND 'UNUSED' BITS
4FF8 2024                  JR NZ,MNSR3       ;DO NOT DEAL WITH SUCH VARS
4FFA
4FFA DBFB                  IN A,(251)
4FFC F5                    PUSH AF
4FFD E5                    PUSH HL
4FFE 7E                    LD A,(HL)
4FFF E61F                  AND 1FH
5001 4F                    LD C,A
5002 7E                    LD A,(HL)
5003 CBB7                  RES 6,A           ;'NOT A FOR-TYPE VARIABLE'**
5005 323F51                LD (TLBYTE),A
5008 23                    INC HL
5009 23                    INC HL
```

```
500A 23                    INC HL             ;PT TO NAME SECOND LET, OR VALUE
500B 2807                  JR Z,MNSR2         ;JR IF 1-LET NUMBER
500D
500D 0600                  LD B,0
500F 114151                LD DE,FIRLET+1
5012 EDB0                  LDIR               ;COPY REST OF NAME TO BUFFER
5014
5014 CD201D      MNSR2:    CALL HLTOFPCS
5017 CD2B2B                CALL CRTVAR4       ;CREATE NUMERIC VAR WITH VALUE ON FPCS, NAME IN
501A                                          ;BUFFER, TYPE IN A
501A E1                    POP HL             ;TLBYTE OF SRC VAR JUST DEALT WITH
501B F1                    POP AF
501C D3FB                  OUT (251),A
501E
501E 23          MNSR3:    INC HL
501F 5E                    LD E,(HL)
5020 23                    INC HL
5021 56                    LD D,(HL)
5022 14                    INC D
5023 20CE                  JR NZ,MNSRL        ;JR IF MORE VARS IN LIST
5025
5025 E1                    POP HL             ;LIST POSN
5026 C9          MEEND:    RET
5027
5027 0127=       MELN      EQU MEEND+1-MEPRO2
5027
5027              ORG 0C000H+RENLN+GETLN+DELLN+KEYLN+POPLN+INPLN+DKLN+RDLN+DFNLN+TOKLN+MELN
C6FB
C6FB                                          ;MERGE, EOF, PTR, PATH$, FN STUBS, LINK
C6FB
C6FB                       INCLUDE FPCMAIN.SAM    ;CALCVARS, FPCMAIN
C6FB             ;FPCMAIN.SAM
C6FB             ;FLOATING POINT CALCULATOR CODES AND ADDRESSES
C6FB
C6FB             ;BINARY
C6FB 9FCB        FPATAB:       DW FPMULT      ;00 MULT
C6FD 3ECD                      DW FPADDN      ;01 ADDN (NUMBERS)
C6FF 8DD0                      DW FPCONCAT    ;02 ADDN (STRINGS)
C701 37CD                      DW FPSUBN      ;03 SUBN
C703 49CA                      DW FPPOWER     ;04 TO-POWER-OF
C705 C2CC                      DW FPDIVN      ;05 DIVN
C707
C707 9718                      DW FPSWOP      ;06 SWOP
C709 58C8                      DW FPDROP      ;07 DROP
C70B
C70B 5DC9                      DW FPMOD       ;08 MOD
C70D 69C9                      DW FPIDIV      ;09 IDIV
C70F 6DC9                      DW FPBOR       ;0A BOR
C711 290D                      DW NONSENSE    ;0B BXOR
C713 76C9                      DW FPBAND      ;0C BAND
C715
C715 02C9                      DW FPOR        ;0D OR
C717
C717 0000=       MULT        EQU 00H
C717 0001=       ADDN        EQU 01H
C717 0002=       CONCAT      EQU 02H
C717 0003=       SUBN        EQU 03H
C717 0004=       POWER       EQU 04H
C717 0005=       DIVN        EQU 05H
C717 0006=       SWOP        EQU 06H
C717 0007=       DROP        EQU 07H
C717 0008=       MOD         EQU 08H
C717 0009=       IDIV        EQU 09H
C717 000D=       NUOR        EQU 0DH
C717
C717 FCC8                      DW FPAND       ;0E N AND N
C719 4CC9                      DW FPNNOTE     ;0F N<>N
C71B 44C9                      DW FPNLESE     ;10 N<=N
C71D 48C9                      DW FPNGRTE     ;11 N>=N
C71F 3CC9                      DW FPNLESS     ;12 N<N
C721 38C9                      DW FPNEQUAL    ;13 N=N
C723 40C9                      DW FPNGRTR     ;14 N>N
C725
C725 08C9                      DW FPSAND      ;15 $ AND N
C727 31C9                      DW FPSNOTE     ;16 $<>$
C729 24C9                      DW FPSLESE     ;17 $<=$
C72B 2BC9                      DW FPSGRTE     ;18 $>=$
C72D 18C9                      DW FPSLESS     ;19 $<$
C72F 13C9                      DW FPSEQUAL    ;1A $=$
C731 1DC9                      DW FPSGRTR     ;1B $>$
C733
C733             ;BINARY INTERNAL OPERATIONS (ALSO DROP, SWOP ABOVE)
C733
C733 7F18                      DW FPSWOP13    ;1C SWOP13
C735 8318                      DW FPSWOP23    ;1D SWOP23
C737 77C8                      DW FPJPTR      ;1E JPTRUE
C739 72C8                      DW FPJPFL      ;1F JPFALSE
C73B
C73B 000E=       NUAND       EQU 0EH
C73B 000F=       NNOTE       EQU 0FH
C73B 0010=       NLESE       EQU 10H
C73B 0011=       NGRTE       EQU 11H
C73B 0012=       NLESS       EQU 12H
C73B 0013=       NEQUAL      EQU 13H
C73B 0014=       NGRTR       EQU 14H
C73B
C73B 0015=       SAND        EQU 15H
C73B 0016=       SNOTE       EQU 16H
C73B 0017=       SLESE       EQU 17H
C73B 0018=       SGRTE       EQU 18H
C73B 0019=       SLESS       EQU 19H
C73B 001A=       SEQUAL      EQU 1AH
C73B 001B=       SGRTR       EQU 1BH
```

```
C73B
C73B 001C=      SWOP13     EQU 1CH
C73B 001D=      SWOP23     EQU 1DH
C73B 001E=      JPTRUE     EQU 1EH
C73B 001F=      JPFALSE    EQU 1FH
C73B
C73B            ;UNARY INTERNAL OPERATIONS
C73B
C73B 68C8                     DW FPJUMP      ;20 JUMP
C73D 59C8                     DW FPLDBREG    ;21 LDBREG
C73F 5FC8                     DW FPDECB      ;22 DECB
C741 AEC8                     DW FPSTKBR     ;23 STKBREG
C743 05C8                     DW FPUSEB      ;24 USEB
C745 53C8                     DW FPDUP       ;25 DUP
C747 B4C8                     DW FP1LIT      ;26 ONELIT
C749 87C8                     DW FP5LIT      ;27 FIVELIT
C74B 81C8                     DW FPSOMELIT   ;28 SOMELIT
C74D 93C8                     DW FPLKADDRB   ;29 LKADDRB
C74F 98C8                     DW FPLKADDRW   ;2A LKADDRW
C751 C1C9                     DW FPREDARG    ;2B REDARG
C753 E2C8                     DW FPLESS0     ;2C LESS0
C755 DCC8                     DW FPLESE0     ;2D LESE0
C757 C1C8                     DW FPGRTR0     ;2E GRTR0
C759 BBC8                     DW FPGRTE0     ;2F GRTE0
C75B D6D1                     DW FPTRUNCT    ;30 TRUNC
C75D 3ECE                     DW FPFORM      ;31 RESTACK
C75F EEC9                     DW FPPOWR2     ;32 POWR2
C761 49C8                     DW FPEXIT      ;33 EXIT
C763 4BC8                     DW FPEXIT2     ;34 EXIT2
C765 4BC8                     DW FPEXIT2     ;35 SPARE
C767 4BC8                     DW FPEXIT2     ;36 SPARE
C769 4BC8                     DW FPEXIT2     ;37 SPARE
C76B 4BC8                     DW FPEXIT2     ;38 SPARE
C76D
C76D 0020=      JUMP       EQU 20H
C76D 0021=      LDBREG     EQU 21H
C76D 0022=      DECB       EQU 22H
C76D 0023=      STKBREG    EQU 23H
C76D 0024=      USEB       EQU 24H
C76D 0025=      DUP        EQU 25H
C76D 0026=      ONELIT     EQU 26H
C76D 0027=      FIVELIT    EQU 27H
C76D 0028=      SOMELIT    EQU 28H
C76D 0029=      LKADDRB    EQU 29H
C76D 002A=      LKADDRW    EQU 2AH
C76D 002B=      REDARG     EQU 2BH
C76D 002C=      LESS0      EQU 2CH
C76D 002D=      LESE0      EQU 2DH
C76D 002E=      GRTR0      EQU 2EH
C76D 002F=      GRTE0      EQU 2FH
C76D 0030=      TRUNC      EQU 30H
C76D 0031=      RESTACK    EQU 31H
C76D 0032=      POWR2      EQU 32H
C76D 0033=      EXIT       EQU 33H
C76D 0034=      EXIT2      EQU 34H
C76D
C76D
C76D            ;OPERATIONS USED BY BASIC.
C76D
C76D            ;PRIORITY 16,NUMERIC ARG, NUMERIC RESULT
C76D
C76D 8AC9                     DW FPSIN       ;39 SIN
C76F B5C9                     DW FPCOS       ;3A COS
C771 BAC9                     DW FPTAN       ;3B TAN
C773 5FCB                     DW FPARCSIN    ;3C ASN
C775 6ECB                     DW FPARCCOS    ;3D ACS
C777 EACA                     DW FPARCTAN    ;3E ATN
C779 6BCA                     DW FPLOGN      ;3F LOGN
C77B E3C9                     DW FPEXP       ;40 EXP
C77D
C77D 0039=      SIN        EQU 39H
C77D 003A=      COS        EQU 3AH
C77D 003B=      TAN        EQU 3BH
C77D 003C=      ASN        EQU 3CH
C77D 003D=      ACS        EQU 3DH
C77D 003E=      ATN        EQU 3EH
C77D 003F=      LOGN       EQU 3FH
C77D 0040=      EXP        EQU 40H
C77D
C77D 82D1                     DW FPABS       ;41 ABS
C77F B0D1                     DW FPSGN       ;42 SGN
C781 62D1                     DW FPSQR       ;43 SQR
C783 C2D1                     DW FPINT       ;44 INT
C785
C785 11D1                     DW FPUSR       ;45 USR
C787 7718                     DW FPIN        ;46 IN
C789 16D1                     DW FPPEEK      ;47 PEEK
C78B 1DD1                     DW FPDPEEK     ;48 DPEEK
C78D E5CE                     DW FPDVAR      ;49 DVAR
C78F ECCF                     DW FPSVAR      ;4A SVAR
C791 D4CF                     DW FPBUTTON    ;4B BUTTON
C793 E8CE                     DW FPEOF       ;4C EOF
C795 EBCE                     DW FPPTR       ;4D PTR - DISC USE
C797 290D                     DW NONSENSE    ;4E UNUSED
C799
C799 0041=      ABS        EQU 41H
C799 0042=      SGN        EQU 42H
C799 0043=      SQR        EQU 43H
C799 0044=      INT        EQU 44H
C799 0046=      INP        EQU 46H
C799 0047=      PEEK       EQU 47H
C799 004C=      EOF        EQU 4CH
C799
```

```
C799             ;PRIORITY 15, STRING ARG, NUMERIC RESULT
C799
C799 28D2                        DW FPUDG       ;4F UDG ADDRESS
C79B 290D                        DW NONSENSE    ;50 NUMBER
C79D 5DD1                        DW FPLEN       ;51 LEN
C79F 4CD1                        DW FPCODE      ;52 CODE
C7A1
C7A1             ;PRIORITY 15, STRING ARG, STRING RESULT
C7A1
C7A1 85CE                        DW FPVALS      ;53 VAL$
C7A3
C7A3             ;PRIORITY 15, STRING ARG, NUMERIC RESULT
C7A3
C7A3 84CE                        DW FPVAL       ;54 VAL
C7A5
C7A5             ;PRIORITY 15, STRING ARG, STRING RESULT
C7A5
C7A5 34D1                        DW FPTRUSTR    ;55 TRUNC$
C7A7
C7A7 004F=       UDGA        EQU 4FH
C7A7 0051=       LEN         EQU 51H
C7A7 0052=       CODE        EQU 52H
C7A7 0053=       VALS        EQU 53H
C7A7 0054=       VAL         EQU 54H
C7A7
C7A7             ;PRIORITY 16, NUMERIC ARG, STRING RESULT
C7A7
C7A7 F2CF                        DW FPCHRS      ;56 CHR$
C7A9 45D1                        DW FPSTRS      ;57 STR$
C7AB 02D0                        DW FPBINS      ;58 BIN$
C7AD 25D0                        DW FPHEXS      ;59 HEX$
C7AF 0CD1                        DW FPUSRS      ;5A USR$
C7B1 B8CF                        DW FPINKEY     ;5B INKEY$
C7B3
C7B3 0056=       CHRS        EQU 56H
C7B3 0057=       STRS        EQU 57H
C7B3 005B=       INKEY       EQU 5BH
C7B3
C7B3             ;ODD PRIORITY, N ARG, N RESULT
C7B3
C7B3 D4C8                        DW FPNOT       ;5C NOT     PRIORITY 4
C7B5 8FD1                        DW FPNEGAT     ;5D NEGATE  PRIORITY 9
C7B7
C7B7 005C=       NOT         EQU 5CH
C7B7 005D=       NEGATE      EQU 5DH
C7B7
C7B7 00EF=       CALC        EQU 0EFH
C7B7 00C8=       STOD0       EQU 0C8H
C7B7 00C9=       STOD1       EQU 0C9H
C7B7 00CA=       STOD2       EQU 0CAH
C7B7 00CB=       STOD3       EQU 0CBH
C7B7 00CC=       STOD4       EQU 0CCH  ;USED BY DRCURVE
C7B7
C7B7 00D0=       STO0        EQU 0D0H
C7B7 00D1=       STO1        EQU 0D1H
C7B7 00D2=       STO2        EQU 0D2H
C7B7 00D3=       STO3        EQU 0D3H
C7B7 00D4=       STO4        EQU 0D4H  ;USED BY DRCURVE
C7B7 00D5=       STO5        EQU 0D5H  ;USED BY DRCURVE
C7B7
C7B7 00D8=       RCL0        EQU 0D8H
C7B7 00D9=       RCL1        EQU 0D9H
C7B7 00DA=       RCL2        EQU 0DAH
C7B7 00DB=       RCL3        EQU 0DBH
C7B7 00DC=       RCL4        EQU 0DCH  ;USED BY DRCURVE
C7B7 00DD=       RCL5        EQU 0DDH  ;USED BY DRCURVE
C7B7
C7B7 00E0=       STKHALF     EQU 0E0H
C7B7 00E1=       STKZERO     EQU 0E1H
C7B7 00E2=       STK16K      EQU 0E2H
C7B7 00E6=       STKFONE     EQU 0E6H
C7B7 00E9=       STKONE      EQU 0E9H
C7B7 00EC=       STKTEN      EQU 0ECH
C7B7 00F0=       STKHALFPI   EQU 0F0H
C7B7
C7B7
C7B7             ;FPCMAIN.SAM - FLOATING POINT CALCULATOR CONTROL
C7B7             ;ENTRY FROM RST 28H. B=JUNK OR VALUE FOR LOOP COUNTER OR CALC CODE FOR USEB
C7B7
C7B7 ED5B655C    FPCMAIN:   LD DE,(STKEND)
C7BB
C7BB 01BBC7      FPCLP:     LD BC,FPCLP
C7BE C5                     PUSH BC             ;'RET' WILL LOOP BACK HERE AT END OF A ROUTINE
C7BF ED53655C               LD (STKEND),DE      ;RESET STKEND IN CASE A BINARY OPERATION HAS
C7C3                                            ;COMBINED TWO VALUES INTO ONE, OR VALUE HAS BEEN
C7C3                                            ;STACKED
C7C3
C7C3 DD7E00                 LD A,(IX+0)         ;GET A PARAMETER
C7C6 DD23                   INC IX              ;PT TO NEXT ONE
C7C8
C7C8             ;ENTRY POINT IF 'A' REGISTER ALREADY HAS CALC CODE.
C7C8
C7C8 2AF05A      BREGEN:    LD HL,(RST28V)
C7CB 24                     INC H
C7CC 25                     DEC H
C7CD C40500                 CALL NZ,HLJUMP      ;CALL WITH A=CODE, IX PAST CODE, DE=STKEND
C7D0                                            ;ALLOWS DIRECT TRANSLATION OF ALL ZX FPC CODES
C7D0                                            ;EXCEPT 34H (STK LITERAL) USING A LOOK-UP TABLE.
C7D0                                            ;SEE FILES FOR 34H METHOD.
C7D0
C7D0 0600                   LD B,0              ;USED LATER..
C7D2 FE20                   CP 20H
C7D4 381A                   JR C,FPCBIN         ;JR IF BINARY OPERATOR (00-1FH)
```

```
C7D6
C7D6 FE60                   CP 60H
C7D8 381B                   JR C,FPCUNA       ;JR IF UNARY OPERATOR (20-5FH)
C7DA
C7DA C620                   ADD A,20H         ;E0-FF ->00-1F
C7DC 384C                   JR C,FPCONST
C7DE
C7DE C608                   ADD A,8           ;D8-DF ->00-07
C7E0 3834                   JR C,FPRCL
C7E2
C7E2 21FBFF                 LD HL,-5
C7E5 19                     ADD HL,DE         ;HL=STKEND-5 (SRC FOR STORED VALUE)
C7E6 C608                   ADD A,8           ;D0-D7 ->00-07
C7E8 3822                   JR C,FPSTO
C7EA
C7EA C608                   ADD A,8           ;C8-CF ->00-07
C7EC 381C                   JR C,FPSTOD
C7EE
C7EE CF                     RST 08H
C7EF 33                     DB 51             ;'FPC error'
C7F0
C7F0            ;HANDLE BINARY FPC OPERATION
C7F0
C7F0 21FBFF     FPCBIN:     LD HL,-5
C7F3 19                     ADD HL,DE
C7F4 EB                     EX DE,HL          ;BACK DE UP TO PT TO STKEND-5
C7F5
C7F5            ;HANDLE UNARY FPC OPERATION
C7F5
C7F5 4F         FPCUNA:     LD C,A
C7F6 CB01                   RLC C             ;BC=PARAM*2 - WORD INDEX
C7F8 21FBC6                 LD HL,FPATAB
C7FB 09                     ADD HL,BC
C7FC 4E                     LD C,(HL)
C7FD 23                     INC HL
C7FE 46                     LD B,(HL)
C7FF C5                     PUSH BC           ;ROUTINE ADDR
C800 21FBFF                 LD HL,-5
C803 19                     ADD HL,DE         ;IF UNARY  OP, HL=STKEND-5, DE=STKEND
C804                                          ;IF BINARY OP, HL=STKEND-10, DE=STKEND-5
C804 C9                     RET               ;TO ROUTINE. CY SET
C805
C805            ;USE (BCREG) SYS VAR AS SOURCE OF CALCULATOR CODE. USED BY EXPR. EVALUATOR
C805            ;UNARY
C805
C805 3A875B     FPUSEB:     LD A,(BCREG+1)
C808 18BE                   JR BREGEN
C80A
C80A            ;STORE-WITH-DELETE USED ABOUT 27 TIMES, SO COMBINE
C80A            ;STOD0, STOD1, ETC
C80A
C80A E5         FPSTOD:     PUSH HL
C80B FE                     DB 0FEH           ;'JR+1'
C80C
C80C            ;STORE TO A MEMORY (NO DELETE)
C80C
C80C D5         FPSTO:      PUSH DE           ;STKEND
C80D
C80D EB                     EX DE,HL
C80E CD1CC8                 CALL LOCMEM1
C811 EB                     EX DE,HL          ;CALC MEM ADDR TO DE, BC=5
C812
C812 EDB0                   LDIR              ;COPY TOP STACK ENTRY TO CALC MEMORY
C814 D1                     POP DE            ;STKEND OR STKEND-5 IF DELETE WANTED
C815 C9                     RET
C816
C816            ;RECALL A MEMORY
C816
C816 CD1CC8     FPRCL:      CALL LOCMEM1      ;FIND CALC MEM ADDR, GET BC=5
C819 EDB0                   LDIR              ;COPY MEMORY TO STKEND
C81B C9                     RET
C81C
C81C 2A685C     LOCMEM1:    LD HL,(MEM)
C81F
C81F 4F         LOCMEM2:    LD C,A
C820 87                     ADD A,A
C821 87                     ADD A,A
C822 81                     ADD A,C           ;*5
C823 4F                     LD C,A
C824 0600                   LD B,0
C826 09                     ADD HL,BC
C827 0E05                   LD C,5
C829 C9                     RET
C82A
C82A 4F         FPCONST:    LD C,A            ;C=00-1F. MANY VALUES OF C MAKE NO SENSE.
C82B 2134C8                 LD HL,FPCTAB
C82E 09                     ADD HL,BC         ;PT TO A CONSTANT. DE=STKEND
C82F 0E05                   LD C,5
C831 EDB0                   LDIR
C833 C9                     RET               ;TO FPCLP WITH DE=NEW STKEND
C834
C834 80         FPCTAB:     DB 80H            ;DISP 0 - 0.5
C835 00                     DB 0              ;DISP 1 - ZERO
C836 00                     DB 0              ;DISP 2 - 16384
C837 00                     DB 0
C838 00                     DB 0
C839 40                     DB 40H
C83A 81                     DB 81H            ;DISP 6 - FP ONE
C83B 00                     DB 0
C83C 00                     DB 0
C83D 00                     DB 0              ;DISP 9 - INTEGER ONE
C83E 00                     DB 0
C83F 01                     DB 1
```

```
C840 00                   DB 0              ;DISP 0C - TEN
C841 00                   DB 0
C842 0A                   DB 10
C843 00                   DB 0
C844
C844 81490FDAA2           DB 81H,49H,0FH,0DAH,0A2H  ;DISP 10H - PI/2
C849
C849            ;TERMINATE FPC PARAM LIST
C849
C849 C1         FPEXIT:   POP BC            ;JUNK FPC LOOPING ADDRESS
C84A C9                   RET               ;RET TO ROM0'S RST 28H ROUTINE
C84B
C84B            ;TERMINATE FPC PARAM LIST AND JUNK A RET ADDR (REPLACES DB EXIT: RET)
C84B
C84B C1         FPEXIT2:  POP BC            ;FPC LOOPING ADDR
C84C C1                   POP BC            ;ROM0 RST 28H RET ADDR
C84D F1                   POP AF            ;PORT STATUS
C84E DDE1                 POP IX            ;ORIG IX
C850 C3F900               JP LRPOUT
C853
C853            ;DUPLICATE LAST ENTRY. UNARY OP
C853
C853 010500     FPDUP:    LD BC,5
C856 EDB0                 LDIR
C858
C858 C9         FPDROP:   RET               ;EXIT IS A UNARY OP SO HL=STKEND-5, DE=STKEND
C859                                        ;DELETE (DROP) IS BINARY SO DE=STKEND-5 - DROP.
C859 DD7E00     FPLDBREG: LD A,(IX+0)
C85C BF                   CP A              ;SET ZERO
C85D 1804                 JR FPLDDCC
C85F
C85F            ;USE (BCREG) SYS VAR AS A COUNTER. DEC AND JR NZ. UNARY
C85F
C85F 3A875B     FPDECB:   LD A,(BCREG+1)
C862 3D                   DEC A
C863
C863 32875B     FPLDDCC:  LD (BCREG+1),A
C866 2816                 JR Z,FPPSKIP
C868
C868            ;UNCONDITIONAL JUMP OF PARAM PTR BY (IX) BYTES. UNARY
C868
C868 DD7E00     FPJUMP:   LD A,(IX+0)
C86B 4F                   LD C,A
C86C 17                   RLA
C86D 9F                   SBC A,A
C86E 47                   LD B,A            ;B=FF OR 00 ACCORDING TO C
C86F DD09                 ADD IX,BC         ;ADJUST IX UP OR DOWN
C871 C9                   RET
C872
C872            ;JUMP IF ZERO. BINARY OPERATOR. DE=STKEND-5 SO 'LAST VALUE' DROPPED
C872
C872 CD51C9     FPJPFL:   CALL TSTZERO      ;Z IF VALUE AT (DE) (IE N2) IS ZERO
C875 1805                 JR FPJPTF
C877
C877            ;JUMP IF TRUE. BINARY OPERATOR. DE=STKEND-5
C877
C877 62         FPJPTR:   LD H,D
C878 6B                   LD L,E
C879 23                   INC HL
C87A 23                   INC HL            ;PT TO 1/0 ON FPCS
C87B 35                   DEC (HL)          ;SET ZERO IF TRUE (1)
C87C
C87C 28EA       FPJPTF:   JR Z,FPJUMP
C87E
C87E DD23       FPPSKIP:  INC IX            ;SKIP BYTES TO JUMP BY
C880 C9                   RET
C881
C881            ;STACK SOME LITERALS FROM PARAM LIST
C881
C881 DD4600     FPSOMELIT: LD B,(IX+0)      ;FETCH NUMBER OF LITERAL BYTES (5,10,15,20 ETC)
C884 DD23                 INC IX
C886 21                   DB 21H            ;"JR+2"
C887
C887            ;STACK 5-BYTE LITERAL FROM PARAM LIST
C887
C887 0605       FP5LIT:   LD B,5
C889
C889 DD7E00     FIVLTLP:  LD A,(IX+0)       ;SRC=PARAM LIST
C88C 12                   LD (DE),A
C88D DD23                 INC IX
C88F 13                   INC DE
C890 10F7                 DJNZ FIVLTLP
C892
C892 C9                   RET               ;DE=NEW STKEND
C893
C893            ;'PEEK' USING ADDRESS IN PARAM LIST
C893
C893 CD9EC8     FPLKADDRB: CALL LKADDRSR
C896 1857                 JR STACKC
C898
C898            ;'DPEEK' USING ADDRESS IN PARAM LIST
C898
C898 CD9EC8     FPLKADDRW: CALL LKADDRSR
C89B C3DD1C               JP STACKBC
C89E
C89E EB         LKADDRSR: EX DE,HL
C89F DD5E00               LD E,(IX+0)
C8A2 DD23                 INC IX
C8A4 DD5600               LD D,(IX+0)
C8A7 DD23                 INC IX
C8A9 EB                   EX DE,HL     ;STKEND TO DE AGAIN
C8AA 4E                   LD C,(HL)
C8AB 23                   INC HL
```

```
C8AC 46                  LD B,(HL)
C8AD C9                  RET
C8AE
C8AE            ;STACK BCREG+1 SYSTEM VAR (HOLDS B REGISTER ENTRY VALUE)
C8AE
C8AE 3A875B     FPSTKBR: LD A,(BCREG+1)
C8B1 4F                  LD C,A
C8B2 183B                JR STACKC
C8B4
C8B4            ;STACK 1-BYTE LITERAL FROM PARAM LIST. ENTRY: DE=STKEND
C8B4
C8B4 DD4E00     FP1LIT:  LD C,(IX+0)
C8B7 DD23                INC IX
C8B9 1834                JR STACKC         ;STACK 00 00 C 00
C8BB
C8BB            ;SET LAST VALUE TO TRUE (1) IF IT IS >=0 (+VE). UNARY
C8BB
C8BB CD52C9     FPGRTE0: CALL TSTZERO2     ;SEE IF (HL) NUMBER IS ZERO
C8BE EB                  EX DE,HL
C8BF 2827                JR Z,SETTRUE
C8C1
C8C1            ;SET LAST VALUE TO TRUE (1) IF IT IS >0 (+VE AND NON-ZERO). UNARY
C8C1
C8C1 23         FPGRTR0: INC HL
C8C2 7E                  LD A,(HL)
C8C3 17                  RLA               ;CY IF -VE
C8C4 2B                  DEC HL
C8C5 3824                JR C,SETFALSE
C8C7
C8C7 23                  INC HL
C8C8 23                  INC HL
C8C9 23                  INC HL
C8CA 7E                  LD A,(HL)         ;MSB
C8CB 2B                  DEC HL
C8CC B6                  OR (HL)           ;LSB
C8CD 2B                  DEC HL
C8CE 2B                  DEC HL
C8CF B6                  OR (HL)           ;EXP
C8D0 2819                JR Z,SETFALSE
C8D2
C8D2 1814                JR SETTRUE
C8D4
C8D4            ;SET LAST VALUE TO TRUE (1) IF IT IS 0. UNARY
C8D4
C8D4 CD52C9     FPNOT:   CALL TSTZERO2
C8D7 EB                  EX DE,HL          ;REVERSE EX AT END OF TSTZERO
C8D8
C8D8            ;SET TRUE IF ZERO, FLASE IF NZ
C8D8
C8D8 280E       ZTRUE:   JR Z,SETTRUE
C8DA
C8DA 180F                JR SETFALSE
C8DC
C8DC            ;SET LAST VALUE TO TRUE (1) IF IT IS <=0. UNARY
C8DC
C8DC CD52C9     FPLESE0: CALL TSTZERO2
C8DF EB                  EX DE,HL
C8E0 2806                JR Z,SETTRUE
C8E2
C8E2            ;SET LAST VALUE TO TRUE (1) IF IT IS <0 (-VE). UNARY
C8E2
C8E2 23         FPLESS0: INC HL
C8E3 7E                  LD A,(HL)
C8E4 2B                  DEC HL
C8E5 17                  RLA               ;CY IF -VE
C8E6
C8E6            ;SET TRUE IF CY, FALSE IF NC
C8E6
C8E6 3003       CYTRUE:  JR NC,SETFALSE
C8E8
C8E8            ;SET LAST VALUE TO 'TRUE' (1). ENTRY: HL PTS TO LAST VALUE
C8E8
C8E8 0E01       SETTRUE: LD C,1
C8EA FE                  DB 0FEH           ;'JR+1'
C8EB
C8EB            ;SET LAST VALUE TO 'FALSE' (0). ENTRY: HL PTS TO LAST VALUE.
C8EB            ;EXIT: HL UNCHANGED, DE=HL+5, A=0
C8EB
C8EB 0E00       SETFALSE: LD C,00H
C8ED 54                  LD D,H
C8EE 5D                  LD E,L
C8EF
C8EF            ;STACK C TO (DE) AS 00 00 C 00 XX
C8EF
C8EF AF         STACKC:  XOR A
C8F0 12                  LD (DE),A         ;'INTEGER'
C8F1 13                  INC DE
C8F2 12                  LD (DE),A         ;'+VE'
C8F3 13                  INC DE
C8F4 79                  LD A,C
C8F5 12                  LD (DE),A         ;LSB FROM PARAM LIST
C8F6 AF                  XOR A
C8F7 13                  INC DE
C8F8 12                  LD (DE),A         ;MSB IS ZERO
C8F9 13                  INC DE
C8FA 13                  INC DE
C8FB C9                  RET
C8FC
C8FC            ;N1 AND N2 - BINARY OPERATION
C8FC
C8FC CD51C9     FPAND:   CALL TSTZERO
C8FF 28EA                JR Z,SETFALSE     ;SET RESULT TO 0 IF N2 WAS ZERO
C901
```

```
C901 C9                    RET               ;ELSE LAST VALUE=N1
C902
C902               ;N1 OR N2 - BINARY OPERATION
C902
C902 CD51C9      FPOR:     CALL TSTZERO
C905 20E1                  JR NZ,SETTRUE     ;SET RESULT TO 1 IF N2 WAS NON-ZERO
C907
C907 C9                    RET               ;ELSE LAST VALUE=N1
C908
C908               ;A$ AND N2 - BINARY OPERATION
C908
C908 CD51C9      FPSAND:   CALL TSTZERO
C90B C0                    RET NZ            ;RETURN A$ IF N2 WAS NON-ZERO
C90C
C90C 1B                    DEC DE
C90D 12                    LD (DE),A
C90E 1B                    DEC DE
C90F 12                    LD (DE),A         ;MAKE STRING LEN=0
C910 13                    INC DE
C911 13                    INC DE
C912 C9                    RET
C913
C913               ;*************************************************************************
C913               ;STRING COMPARISONS. BINARY
C913               ;NOTE: S1<S2 IS TRUE WHEN S1 WOULD APPEAR HIGHER IN AN ALPHABETIC LIST.
C913               ;ENTRY: S1,S2 ON STACK. BINARY OP. EXIT FROM STRCOMP WITH HL=S1 PTR, SET TRUE/
C913               ;FALSE OVERWRITES S1 WITH 0/1, EXITS WITH DE=NEW STKEND
C913
C913               ;S1=S2
C913
C913 CD9D3E      FPSEQUAL: CALL STRCOMP
C916 18C0                  JR ZTRUE          ;SET TRUE ON Z, ELSE FALSE
C918
C918               ;S1<S2
C918
C918 CD9D3E      FPSLESS:  CALL STRCOMP
C91B 18C9                  JR CYTRUE         ;SET TRUE ON CY
C91D
C91D               ;S1>S2
C91D
C91D CD9D3E      FPSGRTR:  CALL STRCOMP
C920 28C9                  JR Z,SETFALSE
C922
C922 180A                  JR NCTRUE         ;SET TRUE ON NC
C924
C924               ;S1<=S2
C924
C924 CD9D3E      FPSLESE:  CALL STRCOMP
C927 28BF                  JR Z,SETTRUE
C929
C929 18BB                  JR CYTRUE         ;SET TRUE ON CY
C92B
C92B               ;S1>=S2
C92B
C92B CD9D3E      FPSGRTE:  CALL STRCOMP
C92E
C92E 3F          NCTRUE:   CCF               ;SET TRUE ON NC
C92F 18B5                  JR CYTRUE
C931
C931               ;S1<>S2
C931
C931 CD9D3E      FPSNOTE:  CALL STRCOMP
C934
C934 28B5        NZTRUE:   JR Z,SETFALSE
C936
C936 18B0                  JR SETTRUE
C938
C938               ;NUMERIC COMPARISONS. BINARY. N1,N2 -> TRUE/FALSE
C938
C938               ;NUMERIC N1=N2
C938
C938 EF          FPNEQUAL: DB CALC
C939 03                    DB SUBN
C93A 5C                    DB NOT            ;SET TRUE IF ZERO
C93B 34                    DB EXIT2
C93C
C93C               ;NUMERIC N1<N2
C93C
C93C EF          FPNLESS:  DB CALC
C93D 03                    DB SUBN
C93E 2C                    DB LESS0          ;SET TRUE IF N1-N2 GIVES -VE
C93F 34                    DB EXIT2
C940
C940               ;NUMERIC N1>N2
C940
C940 EF          FPNGRTR:  DB CALC
C941 03                    DB SUBN
C942 2E                    DB GRTR0          ;SET TRUE IF N1-N2 GIVES +VE, NON-ZERO
C943 34                    DB EXIT2
C944
C944               ;NUMERIC N1<=N2
C944
C944 EF          FPNLESE:  DB CALC
C945 03                    DB SUBN
C946 2D                    DB LESE0          ;SET TRUE IF N1-N2 GIVES -VE OR ZERO
C947 34                    DB EXIT2
C948
C948
C948               ;NUMERIC N1>=N2
C948
C948 EF          FPNGRTE:  DB CALC
C949 03                    DB SUBN
C94A 2F                    DB GRTE0          ;SET TRUE IF N1-N2 GIVES +VE
```

```
C94B 34                     DB EXIT2
C94C
C94C                ;NUMERIC N1<>N2
C94C
C94C EF             FPNNOTE:    DB CALC
C94D 03                     DB SUBN
C94E 5C                     DB NOT              ;SET TRUE IF N1=N2
C94F 5C                     DB NOT              ;SET TRUE IF N1<>N2
C950 34                     DB EXIT2
C951
C951                ;TEST IF ZERO AT (DE). Z IF ZERO, ELSE NZ. HL UNCHANGED
C951
C951 EB             TSTZERO:    EX DE,HL
C952
C952 7E             TSTZERO2:   LD A,(HL)           ;EXP
C953 23                     INC HL
C954 23                     INC HL
C955 B6                     OR (HL)             ;LSB
C956 23                     INC HL
C957 B6                     OR (HL)             ;MSB
C958 2B                     DEC HL
C959 2B                     DEC HL
C95A 2B                     DEC HL
C95B EB                     EX DE,HL
C95C C9                     RET
C95D
C95D                ;N1 MOD N2. RETURNS N1-N2*INT(N1/N2). BINARY
C95D
C95D EF             FPMOD:      DB CALC             ;N1,N2
C95E 25                     DB DUP              ;N1,N2,N2
C95F 1C                     DB SWOP13           ;N2,N2,N1
C960 25                     DB DUP              ;N2,N2,N1,N1
C961 1C                     DB SWOP13           ;N2,N1,N1,N2
C962 05                     DB DIVN
C963 44                     DB INT              ;N2,N1,INT(N1/N2)
C964 D3                     DB STO3             ;COMES IN HANDY FOR UNSTADDR
C965 1D                     DB SWOP23           ;N1,N2,INT(N1/N2)
C966 00                     DB MULT             ;N1,N2*INT(N1/N2)
C967 03                     DB SUBN             ;N1-N2*INT(N1/N2)
C968 34                     DB EXIT2
C969
C969                ;INTEGER DIVIDE. RETURNS INT (N1/N2). BINARY
C969
C969 EF             FPIDIV:     DB CALC             ;N1,N2
C96A 05                     DB DIVN             ;N1/N2
C96B 44                     DB INT              ;INT(N1/N2)
C96C 34                     DB EXIT2
C96D
C96D
C96D                ;N1 BOR N2 (BIT-WISE OR). BINARY
C96D
C96D CD81C9         FPBOR:      CALL GETHLBC
C970 B5                     OR L
C971 4F                     LD C,A
C972 78                     LD A,B
C973 B4                     OR H
C974 1807                   JR BSTKBCH
C976
C976                ;N1 BAND N2 (BIT-WISE AND). BINARY
C976
C976 CD81C9         FPBAND:     CALL GETHLBC
C979 A5                     AND L
C97A 4F                     LD C,A
C97B 78                     LD A,B
C97C A4                     AND H
C97D
C97D 47             BSTKBCH:    LD B,A
C97E C3DD1C                 JP STACKBC
C981
C981                ;GET 2 NUMBERS FROM FPCS AS INTEGERS IN BC (N1) AND HL (N2). A=C
C981
C981 CD2E1D         GETHLBC:    CALL GETINT
C984 C5                     PUSH BC
C985 CD2E1D                 CALL GETINT
C988 E1                     POP HL
C989 C9                     RET
C98A
C98A                        INCLUDE TRANSEND.SAM   ;SIN, COS, EXP, ETC
C98A
C98A                ;TRANSEND.SAM - TRANSCENDENTAL FUNCTIONS, AND ARITHMETIC
C98A                ;W.E. THOMSON FASTER SINE/COSE
C98A
C98A EF             FPSIN:      DB CALC             ;X
C98B 2B                     DB REDARG           ;W
C98C 25                     DB DUP
C98D 25                     DB DUP
C98E 00                     DB MULT             ;W,W*W
C98F C8                     DB STOD0            ;W
C990 F0                     DB STKHALFPI        ;W, FIRST TERM
C991 28                     DB SOMELIT
C992 19                     DB 25               ;25 BYTES FOLLOW (5 CONSTS)
C993
C993 80A55DE72A             DB 80H,0A5H,5DH,0E7H,2AH
C998 7D2335E036             DB 7DH,23H,35H,0E0H,36H
C99D 79996897AE             DB 79H,99H,68H,97H,0AEH
C9A2 74280A109E             DB 74H,28H,0AH,10H,9EH
C9A7 6EE64F4819             DB 6EH,0E6H,4FH,48H,19H
C9AC
C9AC 21                     DB LDBREG           ;COUNTER
C9AD 05                     DB 05H
C9AE
C9AE D8             SINELP:     DB RCL0
C9AF 00                     DB MULT
```

```
C9B0 01                     DB ADDN
C9B1 22                     DB DECB
C9B2 FC                     DB 0FCH           ;LOOP TO SINELP, 5 TIMES
C9B3
C9B3 00                     DB MULT
C9B4 34                     DB EXIT2
C9B5
C9B5              ;COSINE=SINE(ANGLE+PI/2)
C9B5
C9B5 EF           FPCOS:    DB CALC
C9B6 F0                     DB STKHALFPI
C9B7 01                     DB ADDN
C9B8 39                     DB SIN
C9B9 34                     DB EXIT2
C9BA
C9BA              ;TAN=SIN X/COS X
C9BA
C9BA EF           FPTAN:    DB CALC
C9BB 25                     DB DUP
C9BC 39                     DB SIN
C9BD 06                     DB SWOP
C9BE 3A                     DB COS
C9BF 05                     DB DIVN
C9C0 34                     DB EXIT2
C9C1
C9C1              ;REDARG - REDUCE ANGLE ARGUMENT TO -180 TO +180 DEGREE RANGE
C9C1
C9C1 EF           FPREDARG: DB CALC
C9C2 27                     DB FIVELIT
C9C3 7E22F9                 DB 7EH,22H,0F9H
C9C6 836E                   DB 83H,6EH        ;1/(2*PI)
C9C8 00                     DB MULT           ;X/(2*PI)=X IN 360 DEGREE UNITS
C9C9 25                     DB DUP
C9CA E0                     DB STKHALF        ;HALF =180 DEGREES IN THIS CONTEXT
C9CB 01                     DB ADDN
C9CC 44                     DB INT            ;X IN 360 DEGREE UNITS, INT 360 DEGREE UNITS+.5
C9CD 03                     DB SUBN           ;X REDUCED TO -180 TO +180 DEGREES
C9CE 26                     DB ONELIT
C9CF 04                     DB 04H
C9D0 00                     DB MULT           ;REDUCED X*4
C9D1 25                     DB DUP
C9D2 41                     DB ABS
C9D3 E6                     DB STKFONE
C9D4 03                     DB SUBN
C9D5 25                     DB DUP
C9D6 2E                     DB GRTR0
C9D7 1E                     DB JPTRUE         ;TO REDARG2
C9D8 03                     DB 3
C9D9
C9D9 07                     DB DROP
C9DA 34                     DB EXIT2
C9DB
C9DB E6           REDARG2:  DB STKFONE        ;(FLOATING PT)
C9DC 03                     DB SUBN
C9DD 06                     DB SWOP
C9DE 2C                     DB LESS0
C9DF 1E                     DB JPTRUE         ;TO REDARG3
C9E0 02                     DB 2
C9E1 5D                     DB NEGATE
C9E2
C9E2 34           REDARG3:  DB EXIT2
C9E3
C9E3
C9E3              ;EXPONENTIAL - FPC UNARY FUNCTION. (W.E.T.)
C9E3              ;EXP(N)=e^X
C9E3
C9E3 E5           FPEXP:    PUSH HL           ;PTR TO EXP. OF X
C9E4
C9E4 EF                     DB CALC           ;X
C9E5 27                     DB FIVELIT
C9E6 8138                   DB 81H,38H
C9E8 AA3B29                 DB 0AAH,3BH,29H   ;X,1/LN 2
C9EB
C9EB 00                     DB MULT           ;X/LN 2=Y. NOW 2^Y IS REQUIRED RESULT
C9EC 33                     DB EXIT
C9ED
C9ED E1                     POP HL
C9EE
C9EE              ;CALCULATE 2^Y
C9EE
C9EE E5           FPPOWR2:  PUSH HL           ;PTR TO EXP. OF TOP ENTRY
C9EF
C9EF EF                     DB CALC
C9F0 25                     DB DUP
C9F1 44                     DB INT            ;Y,INT Y
C9F2 D1                     DB STO1           ;THE INTEGER PART OF THE POWER CAN JUST
C9F3                                          ;BE ADDED TO EXP OF X (LATER).
C9F3 03                     DB SUBN           ;Y-INT Y=W. NOW DO CLEVER THINGS WITH THE
C9F4                                          ;FRACTIONAL PART OF THE POWER
C9F4
C9F4 C8                     DB STOD0          ;(M0=W)
C9F5 E6                     DB STKFONE        ;FIRST CONST IS 1 (FP FORM)
C9F6 28                     DB SOMELIT
C9F7 23                     DB 35             ;35 BYTES FOLLOW (7 CONSTS)
C9F8
C9F8 8031721816             DB 80H,31H,72H,18H,16H
C9FD 7E75FDE5E7             DB 7EH,75H,0FDH,0E5H,0E7H
CA02 7C6359854A             DB 7CH,63H,59H,85H,4AH
CA07 7A1D821142             DB 7AH,1DH,82H,11H,42H
CA0C 7730071F00             DB 77H,30H,07H,1FH,00H
CA11 7415F05192             DB 74H,15H,0F0H,51H,92H
CA16 7135A06F0B             DB 71H,35H,0A0H,6FH,0BH
CA1B
```

```
CA1B 21                     DB LDBREG
CA1C 07                     DB 07H              ;COUNTER
CA1D
CA1D D8           EXPLP:    DB RCL0
CA1E 00                     DB MULT
CA1F 01                     DB ADDN
CA20 22                     DB DECB
CA21 FC                     DB 0FCH             ;LOOP TO EXPLP, 7 TIMES
CA22
CA22 D9                     DB RCL1             ;2^W,INT Y
CA23 33                     DB EXIT
CA24
CA24 CD651D                 CALL FPTOA          ;GET INTEGER PART OF POWER. CY IF >FF, Z IF +VE.
CA27 280A                   JR Z,FPEXP3         ;JR IF +VE (BC HOLDS RESULT)
CA29
CA29 3004                   JR NC,FPEXP2        ;JR IF IN RANGE (-255 TO -1 AS FF TO 01)
CA2B
CA2B E1                     POP HL              ;SET ZERO IF UNDERFLOW
CA2C
CA2C C3EBC8       SETFALSH: JP SETFALSE
CA2F
CA2F ED44         FPEXP2:   NEG
CA31 4F                     LD C,A
CA32 05                     DEC B               ;NEGATE BC IF -VE
CA33
CA33 E1           FPEXP3:   POP HL              ;PTR TO 2**W
CA34 5E                     LD E,(HL)
CA35 1600                   LD D,0              ;DE=EXP OF 2^W
CA37 EB                     EX DE,HL            ;HL=EXP
CA38 09                     ADD HL,BC           ;ADD POWER TO GET NEW EXPONENT
CA39 EB                     EX DE,HL            ;DE=NEW EXP, HL PTS TO LAST VALUE
CA3A 14                     INC D
CA3B 28EF                   JR Z,SETFALSH       ;ZERO RESULT IF UNDERFLOW
CA3D
CA3D 15                     DEC D
CA3E C2D117                 JP NZ,NTLERR        ;ERROR IF OVERFLOW
CA41
CA41 7B                     LD A,E
CA42 A7                     AND A
CA43 28E7                   JR Z,SETFALSH
CA45
CA45 77                     LD (HL),A
CA46 C3DECE                 JP SETUPDE
CA49
CA49
CA49              ;TO POWER - N1**N2, BINARY
CA49              ;SPECIAL CASES: N1**0=1, 0**NZ=0
CA49
CA49 13           FPPOWER:  INC DE
CA4A 13                     INC DE
CA4B 1A                     LD A,(DE)
CA4C E63F                   AND 3FH
CA4E 47                     LD B,A              ;IF N2=0-3FH, B=N2
CA4F
CA4F EF                     DB CALC             ;N1,N2
CA50 25                     DB DUP              ;N1,N2,N2
CA51 23                     DB STKBREG          ;N1,N2,N2,BREG
CA52 03                     DB SUBN             ;N1,N2,N2-BREG
CA53 1F                     DB JPFALSE          ;JP TO IPOWER IF BREG=N2
CA54 0C                     DB 0CH
CA55
CA55 06                     DB SWOP             ;N2,N1
CA56 25                     DB DUP              ;N2,N1,N1
CA57 1F                     DB JPFALSE          ;JP TO ZPOWER IF N1=ZERO
CA58 05                     DB 05H
CA59
CA59 3F                     DB LOGN             ;N2,LN N1
CA5A 00                     DB MULT             ;N2*LN N1
CA5B 40                     DB EXP              ;E**(N2*LN N1)
CA5C 34                     DB EXIT2
CA5D
CA5D              ;ZERO TO A NON-ZERO POWER
CA5D
CA5D 06           ZPOWER:   DB SWOP             ;0,N2
CA5E 07                     DB DROP             ;0
CA5F 34                     DB EXIT2            ;0 TO ANY POWER GIVES RESULT 0
CA60
CA60              ;N1 TO POWER 0-3FH
CA60
CA60 07           IPOWER:   DB DROP             ;N1
CA61 C8                     DB STOD0            ;(MEM0=N1)
CA62 E9                     DB STKONE           ;1
CA63 23                     DB STKBREG          ;1,BREG
CA64 1F                     DB JPFALSE          ;1
CA65 05                     DB 05H              ;JP IF N1 TO POWER 0 - RESULT=1
CA66
CA66 D8                     DB RCL0             ;T,N1
CA67 00                     DB MULT             ;NEW T
CA68 22                     DB DECB
CA69 FD                     DB 0FDH             ;LOOP, DOING N1*N1*N1*N1...BREG TIMES
CA6A
CA6A 34                     DB EXIT2
CA6B
CA6B              ;*************************************************************************
CA6B              ;LOGN - NATURAL LOG
CA6B
CA6B EF           FPLOGN:   DB CALC             ;X
CA6C 31                     DB RESTACK
CA6D 25                     DB DUP              ;X,X
CA6E 2D                     DB LESE0            ;X,1/0
CA6F 07                     DB DROP             ;X
CA70 33                     DB EXIT
CA71
```

```
CA71 13                     INC DE
CA72 13                     INC DE
CA73 1A                     LD A,(DE)
CA74 A7                     AND A
CA75 C2E83E                 JP NZ,INVARG        ;INVALID ARG IF X IS ZERO OR -VE
CA78
CA78 7E                     LD A,(HL)           ;A=EXPONENT OF X
CA79 3680                   LD (HL),80H         ;X REDUCED TO 0.5-0.999999 RANGE
CA7B 47                     LD B,A
CA7C
CA7C EF                     DB CALC
CA7D 23                     DB STKBREG          ;NX,EXP
CA7E 26                     DB ONELIT
CA7F 80                     DB 80H              ;NX,EXP,80H
CA80 03                     DB SUBN             ;NX,TRUE EXP (PERHAPS -VE)
CA81 06                     DB SWOP             ;TE,NX
CA82 25                     DB DUP              ;TE,NX,NX
CA83 27                     DB FIVELIT
CA84 80CCCC                 DB 80H,0CCH,0CCH
CA87 CCCD                   DB 0CCH,0CDH        ;TE,NX,NX,-0.8
CA89 01                     DB ADDN             ;TE,NX,NX-0.8
CA8A 2E                     DB GRTR0            ;TE,NX,1/0
CA8B 1D                     DB SWOP23           ;NX,TE,1/0
CA8C 1E                     DB JPTRUE           ;TO LOGN3
CA8D 07                     DB 07H
CA8E
CA8E E6                     DB STKFONE          ;NX,TE,1
CA8F 03                     DB SUBN             ;NX,TE-1
CA90 06                     DB SWOP             ;TE-1,NX
CA91 E0                     DB STKHALF          ;TE-1,NX,0.5
CA92 05                     DB DIVN             ;TE-1,2*NX
CA93 06                     DB SWOP             ;2*NX,TE-1
CA94
CA94 27          LOGN3:     DB FIVELIT
CA95 803172                 DB 80H,31H,72H
CA98 17F8                   DB 17H,0F8H         ;NX (OR 2*NX), TE (OR TE-1), LN2
CA9A 00                     DB MULT             ;NX,TE*LN 2
CA9B 06                     DB SWOP
CA9C E6                     DB STKFONE          ;TE*LN 2-1,NX,1
CA9D 03                     DB SUBN             ;TE*LN 2-1,NX-1
CA9E 25                     DB DUP
CA9F 27                     DB FIVELIT
CAA0 822000                 DB 82H,20H,00H      ;TE*LN 2-1,NX-1,NX-1,2.5
CAA3 0000                   DB 00H,00H
CAA5 00                     DB MULT             ;              ,(NX-1)*2.5
CAA6 E0                     DB STKHALF          ;
CAA7 03                     DB SUBN             ;              ,2.5*NX-3
CAA8 C8                     DB STOD0
CAA9 28                     DB SOMELIT
CAAA 3C                     DB 60               ;12 CONSTANTS
CAAB
CAAB 806E238093             DB 80H,6EH,23H,80H,93H
CAB0 7DA79C7E5E             DB 7DH,0A7H,9CH,7EH,5EH
CAB5 7A1B43CA36             DB 7AH,1BH,43H,0CAH,36H
CABA 77A0FE5CFC             DB 77H,0A0H,0FEH,5CH,0FCH
CABF 74319FB400             DB 74H,31H,9FH,0B4H,00H
CAC4 71CBDA9600             DB 71H,0CBH,0DAH,96H,00H
CAC9 6E706F6100             DB 6EH,70H,6FH,61H,00H
CACE 6C90AA0000             DB 6CH,90H,0AAH,00H,00H
CAD3 6930C50000             DB 69H,30H,0C5H,00H,00H
CAD8 66DAA50000             DB 66H,0DAH,0A5H,00H,00H
CADD 6409000000             DB 64H,09H,00H,00H,00H
CAE2 61AC000000             DB 61H,0ACH,00H,00H,00H
CAE7
CAE7 33                     DB EXIT
CAE8
CAE8 185B                   JR SERIES
CAEA
CAEA             ;*************************************************************************
CAEA             ;ARCTAN
CAEA
CAEA EF          FPARCTAN:  DB CALC
CAEB 25                     DB DUP              ;X,X
CAEC 41                     DB ABS              ;X,ABS X        **
CAED E6                     DB STKFONE          ;X,ABS X,1
CAEE 03                     DB SUBN             ;X,ABS X-1
CAEF 2F                     DB GRTE0            ;X,1/0
CAF0 25                     DB DUP              ;X,1/0,1/0 (1 IF ABS X>=1, ELSE 0)
CAF1 1F                     DB JPFALSE          ;JP IF ABS X<1, ARCTAN2, WITH X,0 ON FPCS
CAF2 0B                     DB 0BH
CAF3
CAF3 06                     DB SWOP             ;1,X
CAF4 05                     DB DIVN             ;1/X
CAF5 5D                     DB NEGATE           ;-1/X
CAF6 25                     DB DUP              ;-1/X,-1/X
CAF7 F0                     DB STKHALFPI        ;-1/X,-1/X,PI/2
CAF8 06                     DB SWOP             ;-1/X,PI/2,-1/X
CAF9 2C                     DB LESS0            ;-1/X,PI/2,1/0
CAFA 1E                     DB JPTRUE           ;JP IF -1/X IS -VE, ARCTAN2. (-1/X, PI/2 ON FPCS)
CAFB 02                     DB 02H
CAFC
CAFC 5D                     DB NEGATE           ;NEGATE PI/2 IF -1/X IS -VE
CAFD
CAFD 06          ARCTAN2:   DB SWOP             ;(+/-PI/2 OR 0),V
CAFE 25                     DB DUP              ;               V,V
CAFF 25                     DB DUP              ;               V,V,V
CB00 00                     DB MULT             ;               V,V*V
CB01 E0                     DB STKHALF          ;               V,V*V,0.5
CB02 05                     DB DIVN             ;               V,2*V*V
CB03 E6                     DB STKFONE          ;               V,2*V*V,1
CB04 03                     DB SUBN             ;(+/-PI/2 OR 0),V,2*V*V-1
CB05 C8                     DB STOD0            ;(+/-PI/2 OR 0),V
CB06
```

```
CB06 28                     DB SOMELIT
CB07 3C                     DB 60               ;12 CONSTANTS
CB08
CB08 8061A1B30C             DB 80H,61H,0A1H,0B3H,0CH
CB0D 7CD8DE63BE             DB 7CH,0D8H,0DEH,63H,0BEH
CB12 7936731B5D             DB 79H,36H,73H,1BH,5DH
CB17 76B50936BE             DB 76H,0B5H,09H,36H,0BEH
CB1C 7342C40000             DB 73H,42H,0C4H,00H,00H
CB21 70DBE8B400             DB 70H,0DBH,0E8H,0B4H,00H
CB26 6E00367500             DB 6EH,00H,36H,75H,00H
CB2B 6B98FD0000             DB 6BH,98H,0FDH,00H,00H
CB30 6839BC0000             DB 68H,39H,0BCH,00H,00H
CB35 65E48D0000             DB 65H,0E4H,8DH,00H,00H
CB3A 630E000000             DB 63H,0EH,00H,00H,00H
CB3F 60B2000000             DB 60H,0B2H,00H,00H,00H
CB44
CB44 33                     DB EXIT
CB45
CB45 060C       SERIES:     LD B,12             ;LOOP COUNTER
CB47
CB47 EF                     DB CALC
CB48 D8                     DB RCL0             ;(+/-PI/2 OR 0),V,12 CONSTS,2*V*V-1
CB49                                            ;IGNORE ALL EXCEPT CONSTS AND 2*V*V-1 (CALL IT X)
CB49 E0                     DB STKHALF          ;C,C,..C,X,0.5
CB4A 05                     DB DIVN             ;       ,2*X
CB4B C8                     DB STOD0            ;
CB4C E1                     DB STKZERO          ;0
CB4D D1                     DB STO1             ;0 (INITIAL N)
CB4E
CB4E 25         SERILP:     DB DUP              ;   N,N
CB4F D8                     DB RCL0             ;   N,N,M0
CB50 00                     DB MULT             ;   N,N*M0
CB51 D9                     DB RCL1             ;   N,N*M0,M1
CB52 D2                     DB STO2
CB53 03                     DB SUBN             ;,C,N,N*M0-M1
CB54 1D                     DB SWOP23           ;,N,C,N*M0-M1
CB55 01                     DB ADDN             ;,N,C+N*M0-M1
CB56 06                     DB SWOP             ;,C+N*M0-M1,N
CB57 C9                     DB STOD1            ;,C+N*M0-M1
CB58 22                     DB DECB             ;DEC BREG, JR NZ, TO SERILP
CB59 F5                     DB 0F5H
CB5A
CB5A DA                     DB RCL2
CB5B 03                     DB SUBN
CB5C 00                     DB MULT
CB5D 01                     DB ADDN
CB5E 34                     DB EXIT2
CB5F
CB5F            ;ARCSIN
CB5F
CB5F EF         FPARCSIN:   DB CALC             ;X
CB60 25                     DB DUP              ;X,X
CB61 25                     DB DUP              ;X,X,X
CB62 00                     DB MULT             ;X,X*X
CB63 E6                     DB STKFONE
CB64 06                     DB SWOP
CB65 03                     DB SUBN             ;X,1-(X*X)
CB66 43                     DB SQR
CB67 E6                     DB STKFONE
CB68 01                     DB ADDN             ;X,SQR(1-(X*X))+1
CB69 05                     DB DIVN             ;TAN
CB6A 3E                     DB ATN
CB6B E0                     DB STKHALF
CB6C 05                     DB DIVN
CB6D 34                     DB EXIT2
CB6E
CB6E            ;ARCCOS. (ACS X=PI/2-ASN X)
CB6E
CB6E EF         FPARCCOS:   DB CALC
CB6F 3C                     DB ASN
CB70 F0                     DB STKHALFPI
CB71 06                     DB SWOP
CB72 03                     DB SUBN
CB73 34                     DB EXIT2
CB74
CB74                        INCLUDE MULT.SAM       ;ARITHMETIC
CB74
CB74            ;MULT.SAM - MULTIPLY, DIVIDE, SUBTRACT, ADD
CB74
CB74            ;*******************************************************************************
CB74            ;QMULT - MULTIPLY HL*DE. RESULT IN HL. FASTER IF EITHER NUMBER IS ONE BYTE.
CB74            ;USES HL,AF. CY IF OVERFLOW OF 16 BITS.
CB74
CB74 C5         QMULT:          PUSH BC
CB75 0608                       LD B,8              ;BITS TO DO
CB77 7C                         LD A,H
CB78 A7                         AND A
CB79 2803                       JR Z,QMULT2         ;JR IF HL NUMBER IS ONLY 8 BITS - QUICK MULT OK
CB7B
CB7B EB                         EX DE,HL            ;SWOP NUMBERS AND SEE IF ORIG DE IS 8 BIT
CB7C 7C                         LD A,H
CB7D A7                         AND A
CB7E
CB7E 4D         QMULT2:         LD C,L
CB7F 210000                     LD HL,0
CB82 280D                       JR Z,SHORTMUL
CB84
CB84 29         QHLLP2:         ADD HL,HL           ;DOUBLE RESULT
CB85 3816                       JR C,QOVERFL
CB87
CB87 17                         RLA                 ;TEST NEXT BIT OF ORIG H
CB88 3003                       JR NC,NOQADN
CB8A
```

```
CB8A 19                         ADD HL,DE
CB8B 3810                       JR C,QOVERFL
CB8D
CB8D 10F5       NOQADN:         DJNZ QHLLP2
CB8F
CB8F 0608                       LD B,8
CB91
CB91 79         SHORTMUL:       LD A,C              ;A=ORIG L
CB92
CB92 29         QHLLOOP:        ADD HL,HL           ;DOUBLE RESULT
CB93 3808                       JR C,QOVERFL
CB95
CB95 17                         RLA                 ;TEST NEXT BIT OF ORIG HL
CB96 3003                       JR NC,NOQADDIN
CB98
CB98 19                         ADD HL,DE
CB99 3802                       JR C,QOVERFL
CB9B
CB9B 10F5       NOQADDIN:       DJNZ QHLLOOP
CB9D
CB9D C1         QOVERFL:        POP BC
CB9E C9                         RET                 ; RESULT IS IN HL IF NC SIGNALS 'OK'
CB9F            ;***************************************************************************
CB9F            ;FPC MULTIPLY. BINARY OP WITH HL AND DE PTING TO N1,N2
CB9F
CB9F 1A         FPMULT:         LD A,(DE)
CBA0 B6                         OR (HL)
CBA1 2054                       JR NZ,FPMULT2       ;JR IF EITHER NUMBER IS FLOATING-POINT
CBA3
CBA3 D5                         PUSH DE             ;N2 PTR
CBA4 E5                         PUSH HL             ;N1 PTR
CBA5 CDDCCB                     CALL FETCHI         ;GET N1 IN DE, SGN IN C
CBA8 D5                         PUSH DE
CBA9 41                         LD B,C
CBAA 23                         INC HL
CBAB 23                         INC HL              ;PT TO N2
CBAC CDDCCB                     CALL FETCHI
CBAF E1                         POP HL              ;N1
CBB0 79                         LD A,C
CBB1 A8                         XOR B
CBB2 4F                         LD C,A              ;HL=N1, DE=N2, C=SGN OF RESULT
CBB3 CD74CB                     CALL QMULT          ;HL=HL*DE
CBB6 D1                         POP DE              ;N1 PTR
CBB7 EB                         EX DE,HL            ;HL=N1 PTR, DE=RESULT
CBB8 383C                       JR C,QMOVERF        ;JR IF OVERFLOW OF 16 BITS
CBBA
CBBA 1807                       JR STICZ
CBBC
CBBC            ;LOAD (HL) WITH INTEGER DE, SGN A
CBBC
CBBC FE02       STORADE:    CP 2
CBBE DE00                   SBC A,0             ;01->00, FF->FF
CBC0 2F                     CPL                 ;01->FF, FF->00
CBC1 4F                     LD C,A              ;C=SGN
CBC2 D5                     PUSH DE               ;JUST TO BALANCE STACK **
CBC3
CBC3 7A         STICZ:      LD A,D
CBC4 B3                     OR E
CBC5 2001                   JR NZ,STOREI
CBC7
CBC7 4F                     LD C,A              ;ENSURE SGN OF ZERO IS +VE
CBC8            ;***************************************************************************
CBC8            ;STORE INTEGER TO (HL). DE=INTEGER, C=SGN. ENTRY WITH STKEND ON STACK
CBC8
CBC8 3600       STOREI:         LD (HL),0           ;'INTEGER'
CBCA 23                         INC HL
CBCB 71                         LD (HL),C           ;SGN
CBCC 23                         INC HL
CBCD 0C                         INC C
CBCE 2007                       JR NZ,STORPI
CBD0
CBD0 7B                         LD A,E              ;NEGATE DE IF SGN IS -VE
CBD1 2F                         CPL
CBD2 5F                         LD E,A
CBD3 7A                         LD A,D
CBD4 2F                         CPL
CBD5 57                         LD D,A
CBD6 13                         INC DE
CBD7
CBD7 73         STORPI:         LD (HL),E
CBD8 23                         INC HL
CBD9 72                         LD (HL),D
CBDA D1                         POP DE              ;NEW STKEND
CBDB C9                         RET
CBDC            ;***************************************************************************
CBDC            ;FETCH INTEGER FROM (HL). DE=INTEGER, C=SGN
CBDC
CBDC 23         FETCHI:         INC HL
CBDD 4E                         LD C,(HL)
CBDE 23                         INC HL
CBDF 5E                         LD E,(HL)
CBE0 23                         INC HL
CBE1 56                         LD D,(HL)
CBE2 0D                         DEC C
CBE3 0C                         INC C
CBE4 C8                         RET Z               ;RET IF +VE
CBE5
CBE5 1B         NEGDE:          DEC DE
CBE6
CBE6 7B         CPLDE:          LD A,E              ;NEGATE DE, SINCE SGN IS -VE
CBE7 2F                         CPL
```

```
CBE8 5F                     LD E,A
CBE9 7A                     LD A,D
CBEA 2F                     CPL
CBEB 57                     LD D,A
CBEC B3                     OR E            ;Z IF DE=0
CBED C9                     RET
CBEE
CBEE CDE5CB     NEGDEDE:    CALL NEGDE
CBF1 D9                     EXX             ;EXIT WITH REGS SWITCHED
CBF2 20F2                   JR NZ,CPLDE     ;CPL MSW UNLESS Z SHOWS
CBF4
CBF4 18EF                   JR NEGDE
CBF6
CBF6
CBF6            ;DEAL WITH INTEGER MULTIPLY OVERFLOW
CBF6
CBF6 D1         QMOVERF:    POP DE          ;N2 PTR
CBF7
CBF7            ;*************************************************************************
CBF7            ;FLOATING POINT MULTIPLY OF 2 NUMBERS. N1, N2 ARE ON FPCS AT (HL) AND (DE)
CBF7
CBF7 CD39CE     FPMULT2:    CALL DFPFORM    ;GET FP FORMS ON FPCS. EXIT WITH Z IF N1 IS A
CBFA                                        ;POWER OF TWO, CY IF IT IS ZERO. F'=DITTO FOR N2
CBFA D8                     RET C           ;IF N1=0, RET (BIN. OP MEANS ONLY 0 LEFT ON FPCS)
CBFB
CBFB 08                     EX AF,AF'
CBFC 382C                   JR C,MSETZ2     ;JR IF N2 IS ZERO - SET ZERO RESULT **
CBFE
CBFE E5                     PUSH HL         ;N1 PTR
CBFF 08                     EX AF,AF'
CC00 2803                   JR Z,FPQM1      ;JR IF N1 IS A POWER OF TWO
CC02
CC02 08                     EX AF,AF'
CC03 2028                   JR NZ,FPMULT3   ;JR IF N2 ISN'T A POWER OF TWO
CC05
CC05            ;SPECIAL CASE - EITHER NUMBER (OR BOTH) IS A POWER OF TWO, I.E. EXP 80 00 00 00
CC05            ;OR EXP 00 00 00 00. XOR WILL GIVE CORRECT SGN BIT AND COPY NON-ZERO MANTISSA
CC05            ;BITS. THEN EXPS CAN BE ADDED.
CC05
CC05 1A         FPQM1:      LD A,(DE)
CC06 4F                     LD C,A
CC07 0600                   LD B,0
CC09 6E                     LD L,(HL)
CC0A 60                     LD H,B
CC0B 09                     ADD HL,BC
CC0C 017FFF                 LD BC,-129
CC0F
CC0F            ;ENTRY FROM QUICK DIVIDE
CC0F
CC0F 09         FPQMDC:     ADD HL,BC
CC10 24                     INC H
CC11 2816                   JR Z,MSETZERO   ;SET ZERO IF UNDERFLOW
CC13
CC13 25                     DEC H
CC14 C2D117                 JP NZ,NTLERR    ;ERROR IF OVERFLOW
CC17
CC17 7D                     LD A,L          ;EXP OF RESULT
CC18 A7                     AND A
CC19 280E                   JR Z,MSETZERO
CC1B
CC1B E1                     POP HL          ;N1 PTR
CC1C 77                     LD (HL),A
CC1D 0604                   LD B,4          ;4 MANT BYTES
CC1F
CC1F 23         FPQMULP:    INC HL
CC20 13                     INC DE
CC21 1A                     LD A,(DE)
CC22 AE                     XOR (HL)
CC23 77                     LD (HL),A
CC24 10F9                   DJNZ FPQMULP
CC26
CC26 23                     INC HL
CC27 EB                     EX DE,HL        ;DE=NEW STKEND
CC28 C9                     RET
CC29
CC29 E1         MSETZERO:   POP HL
CC2A
CC2A C3EBC8     MSETZ2:     JP SETFALSE
CC2D
CC2D CD0BCE     FPMULT3:    CALL MUDIADSR   ;GET N1 IN B DE'DE, N2 IN C A'B'C'A, PUT
CC30                                        ;RESULT SIGN ON STACK, SET TRUE NUMERIC BITS
CC30
CC30 C5                     PUSH BC         ;EXP BYTES
CC31 210000                 LD HL,0         ;ZERO RESULT LSW
CC34 D9                     EXX
CC35 210000                 LD HL,0         ;ZERO RESULT MSW
CC38 D9                     EXX
CC39
CC39            ;MULTIPLIER IS IN A',B',C',A
CC39            ;RESULT IS IN HL'HL
CC39            ;MULTIPLICAND IS IN DE'DE
CC39
CC39 0E04                   LD C,4          ;4 BYTES IN MULTIPLIER
CC3B
CC3B 0608       FPMULCLP:   LD B,8          ;8 BITS PER MULTIPLIER BYTE
CC3D A7                     AND A
CC3E 200B                   JR NZ,FPMULBLP ;JR IF ANY BIT IN MULTIPLIER IS HIGH
CC40
CC40 45                     LD B,L          ;EXTRA BITS TO B
CC41 6C                     LD L,H
CC42
CC42 D9                     EXX
CC43 7D                     LD A,L
```

```
CC44 6C                         LD L,H
CC45 2600                       LD H,0
CC47 D9                         EXX
CC48
CC48 67                         LD H,A          ;0->H'->L'->H->L->B. SHIFT RESULT RIGHT 8 TIMES -
CC49                                            ;THIS IS ALL THAT HAPPENS IF ALL MULTIPLIER BITS=0
CC49 1816                       JR NXMBYTE      ;GET NEXT MULTIPLIER BYTE
CC4B
CC4B 1F         FPMULBLP:       RRA             ;TEST EACH BIT IN MULTIPLIER, FROM RHS TO LHS
CC4C 3005                       JR NC,FPMULT25 ;JR IF NO ADD OF MULTIPLICAND
CC4E
CC4E 19                         ADD HL,DE       ;ELSE ADD HL'HL,DE'DE
CC4F D9                         EXX
CC50 ED5A                       ADC HL,DE
CC52 D9                         EXX
CC53
CC53 D9         FPMULT25:       EXX             ;HALVE RESULT IN HL'HL
CC54 CB1C                       RR H
CC56 CB1D                       RR L            ;HALVE HL'
CC58 D9                         EXX
CC59 CB1C                       RR H
CC5B CB1D                       RR L            ;HALVE HL
CC5D 10EC                       DJNZ FPMULBLP
CC5F
CC5F 1F                         RRA
CC60 47                         LD B,A          ;SAVE BITS THAT 'FELL OFF' HL'HL IN CASE THIS IS
CC61                                            ;FINAL BYTE OF MULTIPLIER.
CC61 D9         NXMBYTE:        EXX
CC62 79                         LD A,C
CC63 48                         LD C,B
CC64 08                         EX AF,AF'
CC65 47                         LD B,A
CC66 08                         EX AF,AF'       ;A'->B'->C'->A. A IS NEXT MULTIPLIER BYTE
CC67 D9                         EXX
CC68
CC68 0D                         DEC C
CC69 20D0                       JR NZ,FPMULCLP
CC6B
CC6B 78                         LD A,B          ;A=EXTRA BITS (7 AND 6 MATTER)
CC6C 41                         LD B,C          ;B=0
CC6D EB                         EX DE,HL        ;PROTECT RESULT LSW IN DE
CC6E E1                         POP HL          ;EXP BYTES (BOTH NON-ZERO)
CC6F 4C                         LD C,H          ;BC=N1 EXP (B IS 0)
CC70 60                         LD H,B          ;HL=N2 EXP
CC71 09                         ADD HL,BC       ;HL=EXP SUM+80H+80H
CC72 0180FF                     LD BC,-80H
CC75
CC75            ;ENTRY USED BY DIVIDE
CC75            ;RESULT IS IN HL'DE, EXTRA BITS IN A. BC=+/-80H. BIT 7 OF STACKED AF=RESULT SGN
CC75            ;HL=COMBINED EXP. BELOW SGN ON STACK IS N1 PTR
CC75
CC75 09         DIVISE:         ADD HL,BC       ;SUBTRACT 80H FROM COMBINED EXP TO CORRECT FOR 2
CC76                                            ;+80Hs AS A RESULT OF ADDING 2 EXP BYTES.
CC76                                            ;(OR VV. IF DIVIS - ADD 80H TO COMP FOR 80H-80H)
CC76 D9                         EXX
CC77 CB7C                       BIT 7,H         ;H'
CC79 200A                       JR NZ,MULNORM   ;IF NO LEADING ZERO BITS, OK
CC7B
CC7B D9                         EXX
CC7C 2B                         DEC HL          ;DEC COMBINED EXP.
CC7D                                            ;MOVE RESULT TO THE LEFT, ADD IN A 'LOST'
CC7D 17                         RLA             ;BIT THAT HAS BEEN IN 'A'
CC7E CB13                       RL E            ;SINCE LAST HALVING OF RESULT.
CC80 CB12                       RL D
CC82 D9                         EXX             ;SMALLEST MULT IS .1000*.1000 GIVES .01 SO
CC83
CC83 ED6A                       ADC HL,HL       ;1 MOVE IS ALWAYS ENOUGH TO NORMALISE RESULT.
CC85                                            ;(DITTO FOR DIV)
CC85 D9         MULNORM:        EXX
CC86 17                         RLA             ;LOOK AT A 'LOST' BIT FOR FINAL ROUNDING
CC87
CC87            ;ENTRY FROM FPADDN WITH CY IF NEEDS ROUND UP, HL'DE=RESULT, HL=EXP, SGN AND N1
CC87            ;PTR ON STACK
CC87
CC87 3015       FPADDNEN:       JR NC,RNDDONE   ;JR IF NO NEED TO ROUND UP
CC89
CC89 1C                         INC E
CC8A 2012                       JR NZ,RNDDONE
CC8C
CC8C 14                         INC D
CC8D 200F                       JR NZ,RNDDONE
CC8F
CC8F D9                         EXX
CC90 2C                         INC L
CC91 D9                         EXX
CC92 200A                       JR NZ,RNDDONE
CC94
CC94 D9                         EXX
CC95 24                         INC H
CC96 D9                         EXX
CC97 2005                       JR NZ,RNDDONE
CC99
CC99 23                         INC HL          ;ONLY IF ROUNDING RIPPLES RIGHT FORWARDS INC EXP
CC9A D9                         EXX
CC9B 2680                       LD H,80H        ;AND MAKE MANTISSA .1000000
CC9D D9                         EXX
CC9E
CC9E F1         RNDDONE:        POP AF          ;BIT 7=SGN BIT
CC9F
CC9F            ;ENTRY FROM FPSUBN WITH BIT 7,A=SGN, HL'DE=MANT, HL=EXP, (SP)=N1 PTR
CC9F
CC9F 24         FPSUBNEN:       INC H
CCA0 CA29CC                     JP Z,MSETZERO   ;IF EXP UNDERFLOWED, USE ZERO
```

```
CCA3
CCA3 25                     DEC H           ;ZERO UNLESS OVERFLOW
CCA4 C2D117                 JP NZ,NTLERR    ;H MUST BE ZERO NOW, SINCE WE ONLY USE 1 BYTE EXPS!
CCA7
CCA7 2C                     INC L           ;RESULT NOW IN L HL'DE
CCA8 2D                     DEC L
CCA9 CA29CC                 JP Z,MSETZERO   ;JR IF EXP IS ZERO
CCAC
CCAC D9                     EXX
CCAD E5                     PUSH HL         ;GET RESULT MSW FROM HL'
CCAE D9                     EXX
CCAF
CCAF C1                     POP BC          ;RESULT MSW
CCB0 A8                     XOR B
CCB1 E680                   AND 80H         ;USE BIT 7 FROM A, REST FROM B. A=SGNED MANT 1
CCB3 A8                     XOR B
CCB4 45                     LD B,L
CCB5
CCB5 E1         LDMRSLT:    POP HL          ;N1 PTR
CCB6 70                     LD (HL),B
CCB7 23                     INC HL
CCB8 77                     LD (HL),A
CCB9 23                     INC HL
CCBA 71                     LD (HL),C
CCBB 23                     INC HL
CCBC 72                     LD (HL),D
CCBD 23                     INC HL
CCBE 73                     LD (HL),E
CCBF 23                     INC HL
CCC0 EB                     EX DE,HL        ;DE PTS TO N2 (WILL BE STKEND)
CCC1 C9                     RET
CCC2            ;DIVISION
CCC2            ;34-BIT DIVISION. MANTISSAS COULD BE 0.999/0.5=1.999 OR 0.5/0.999=0.5
CCC2            ;RANGE OF RESULTS=0.5-1.999 BUT METHOD GIVES HALF THIS (0.25-0.999). ALWAYS
CCC2            ;INC EXPONENT TO DOUBLE VALUE. A 33RD BIT MAY BE NEEDED TO NORMALISE RESULT
CCC2            ;TO 0.5-0.999, AND A FINAL BIT IS NEEDED FOR ROUNDING. (SHARED WITH MULTIPLY)
CCC2
CCC2
CCC2 CD39CE     FPDIVN:     CALL DFPFORM    ;GET FP FORMS ON FPCS. EXIT WITH Z IF N1 IS A
CCC5                                        ;POWER OF TWO, Z/C IF IT IS ZERO. F'=DITTO FOR N2
CCC5 D8                     RET C           ;IF N1=0, RET (BIN. OP MEANS ONLY 0 LEFT ON FPCS
CCC6
CCC6 E5                     PUSH HL         ;N1 PTR
CCC7 08                     EX AF,AF'
CCC8 DAD117                 JP C,NTLERR     ;JP IF N2 IS ZERO - OVERFLOW FROM DIVN BY ZERO
CCCB
CCCB 200E                   JR NZ,FPDIVN2   ;JR IF N2 ISN'T A POWER OF TWO
CCCD
CCCD            ;SPECIAL CASE - N2 IS A POWER OF TWO, E.G. EXP 80 00 00 00 OR EXP 00 00 00 00.
CCCD            ;THIS ALLOWS A SIMPLE EXP SUBTRACTION.
CCCD
CCCD 1A         FPQD1:      LD A,(DE)
CCCE 4F                     LD C,A
CCCF 0600                   LD B,0
CCD1 6E                     LD L,(HL)
CCD2 60                     LD H,B          ;NC HERE
CCD3 ED42                   SBC HL,BC       ;SUB N1 EXP,N2 EXP
CCD5 018100                 LD BC,129
CCD8 C30FCC                 JP FPQMDC
CCDB
CCDB CD0BCE     FPDIVN2:    CALL MUDIADSR
CCDE EB                     EX DE,HL
CCDF D9                     EXX
CCE0 EB                     EX DE,HL
CCE1 D9                     EXX             ;GET DIVISOR IN DE'DE, DIVIDEND IN HL'HL
CCE2                                        ;SGN OF RESULT TO STACK, SET TRUE NUM. BITS
CCE2 CDEECB                 CALL NEGDEDE
CCE5 D9                     EXX             ;NEGATE DIVISOR IN DE'DE. COMP FOR EXX IN NEGDEDE
CCE6
CCE6 C5                     PUSH BC         ;EXP BYTES
CCE7
CCE7 01FC08                 LD BC,08FCH     ;8 BITS PER BYTE OF RESULT.
CCEA                                        ;C GOES FD,FE,FF,00 FOR 4 BYTES OF RESULT,
CCEA                                        ;SAVES 33RD BIT AS CY ON FINAL LOOP WITH C=0
CCEA 1817                   JR FPDIVE
CCEC
CCEC 0608       FPDIV3:     LD B,8          ;8 BITS PER BYTE
CCEE
CCEE D9                     EXX
CCEF 08                     EX AF,AF'
CCF0 78                     LD A,B
CCF1 08                     EX AF,AF'
CCF2 41                     LD B,C
CCF3 4F                     LD C,A          ;A'<- B'<- C'<-A. MOVE RESULT BYTES
CCF4 D9                     EXX
CCF5
CCF5 29         FPDIVLP:    ADD HL,HL
CCF6 D9                     EXX
CCF7 ED6A                   ADC HL,HL       ;MOVE DIVIDEND LEFT
CCF9 D9                     EXX
CCFA 3007                   JR NC,FPDIVE
CCFC
CCFC 19                     ADD HL,DE       ;IF CY, ADD NEG DIVISOR, SCF
CCFD D9                     EXX
CCFE ED5A                   ADC HL,DE
CD00 37                     SCF
CD01 180C                   JR FPDIV4
CD03
CD03 19         FPDIVE:     ADD HL,DE       ;ADD NEGATED DIVISOR IN DE'DE TO DIVIDEND IN HL'HL
CD04 D9                     EXX
CD05 ED5A                   ADC HL,DE       ;MSW
CD07 3806                   JR C,FPDIV4     ;JR IF OK, ELSE REVERSE
CD09
```

```
CD09 D9                     EXX
CD0A ED52                   SBC HL,DE
CD0C D9                     EXX
CD0D ED52                   SBC HL,DE           ;MSW. NC
CD0F
CD0F D9           FPDIV4:   EXX
CD10 17                     RLA                 ;SHIFT RESULT BIT IN
CD11 10E2                   DJNZ FPDIVLP
CD13
CD13 0C                     INC C
CD14 FAECCC                 JP M,FPDIV3         ;JP IF C=FD/FE/FF
CD17
CD17 F5                     PUSH AF             ;SAVE LAST WHOLE RESULT BYTE, OR 33RD AND 34TH BITS
CD18
CD18 0602                   LD B,2
CD1A 28D9                   JR Z,FPDIVLP        ;JR IF STILL GOT 33RD AND 34TH BITS TO DO
CD1C
CD1C D9                     EXX                 ;RESULT FROM A'B'C'A TO HL' DE
CD1D 79                     LD A,C
CD1E D9                     EXX
CD1F 57                     LD D,A              ;D
CD20 D9                     EXX
CD21 68                     LD L,B              ;L'
CD22 08                     EX AF,AF'
CD23 67                     LD H,A              ;H'
CD24 D9                     EXX
CD25 F1                     POP AF              ;33/34
CD26 0F                     RRCA
CD27 0F                     RRCA                ;33/34 BITS TO BITS 7 AND 6
CD28 E1                     POP HL
CD29 5C                     LD E,H              ;E
CD2A
CD2A C1                     POP BC              ;EXP BYTES
CD2B 68                     LD L,B
CD2C 2600                   LD H,0
CD2E 44                     LD B,H              ;HL=N1 EXP, BC=N2 EXP
CD2F A7                     AND A
CD30 ED42                   SBC HL,BC
CD32 0E81                   LD C,81H            ;80H TO BE ADDED TO COMP FOR 80H-80H IN EXP SBC,
CD34                                            ;PLUS 1 TO COMP FOR RESULT IN 0.25-0.999 RANGE
CD34
CD34              ;HL'DE=MANT, EXTRA BITS IN A. BC=81H. BIT 7 OF STACKED AF=SGN. (SP+2)=N1 PTR
CD34
CD34 C375CC                 JP DIVISE           ;NORMALISE  (SHARED WITH MULIPLY)
CD37
CD37              ;ADDITION/SUBTRACTION
CD37              ;FPC BINARY OPERATION.
CD37
CD37 EB           FPSUBN:       EX DE,HL
CD38 E5                         PUSH HL
CD39 CD8FD1                     CALL FPNEGAT      ;FOR SUBTRACTION, NEGATE N2, THEN ADD
CD3C E1                         POP HL
CD3D EB                         EX DE,HL
CD3E
CD3E 1A           FPADDN:       LD A,(DE)
CD3F B6                         OR (HL)
CD40 2034                       JR NZ,FPADDN2     ;JR IF EITHER NUMBER IS FLOATING-POINT
CD42
CD42 D5                         PUSH DE           ;N2 PTR
CD43 23                         INC HL
CD44 E5                         PUSH HL           ;PTR TO SGN N1
CD45 23                         INC HL
CD46 4E                         LD C,(HL)
CD47 23                         INC HL
CD48 46                         LD B,(HL)         ;BC=N1
CD49 23                         INC HL
CD4A 23                         INC HL
CD4B 23                         INC HL
CD4C 7E                         LD A,(HL)         ;SGN N2
CD4D 23                         INC HL
CD4E 5E                         LD E,(HL)
CD4F 23                         INC HL
CD50 56                         LD D,(HL)         ;DE=N2
CD51 EB                         EX DE,HL          ;HL=N2
CD52 09                         ADD HL,BC
CD53 0600                       LD B,0
CD55 EB                         EX DE,HL          ;DE=N1+N2
CD56 E1                         POP HL            ;PTR TO N1 SGN
CD57 8E                         ADC A,(HL)
CD58 0F                         RRCA
CD59 88                         ADC A,B
CD5A 2018                       JR NZ,ADDNOVERF
CD5C
CD5C 9F                         SBC A,A
CD5D 77                         LD (HL),A         ;SIGN
CD5E
CD5E 7A                         LD A,D            ;DEAL WITH MINUS ZERO
CD5F B3                         OR E
CD60 200A                       JR NZ,MIN0OK      ;JR IF CANNOT BE 00 FF 00 00
CD62
CD62 7E                         LD A,(HL)
CD63 3C                         INC A
CD64 2006                       JR NZ,MIN0OK      ;DITTO
CD66
CD66 2B                         DEC HL
CD67 3691                       LD (HL),91H       ;EXP
CD69 23                         INC HL
CD6A 3680                       LD (HL),80H       ;MAKE RESULT=65536
CD6C
CD6C 23           MIN0OK:       INC HL
CD6D 73                         LD (HL),E
CD6E 23                         INC HL
CD6F 72                         LD (HL),D
```

```
CD70 23                         INC HL
CD71 70                         LD (HL),B       ;ZERO MANT 4 IN CASE RESULT IS 65536
CD72 D1                         POP DE          ;DE=NEW STKEND
CD73 C9                         RET
CD74
CD74            ;*******************************************************************************
CD74 D1         ADDNOVERF:      POP DE
CD75 2B                         DEC HL          ;GET N1 AND N2 PTRS CORRECT
CD76
CD76            ;N1+N2
CD76 E5         FPADDN2:        PUSH HL         ;N1 PTR
CD77 CD39CE                     CALL DFPFORM    ;USE FULL FP FORMS
CD7A CD0BCE                     CALL MUDIADSR   ;N1 IN DE'DE, N2 IN HL'HL, EXPS IN BC
CD7D                                            ;TRUE NUMERIC BITS, XORED SGNS ON STACK
CD7D 79                         LD A,C
CD7E 90                         SUB B           ;SUB N2 EXP, N1 EXP. CY IF N1 EXP >N2 EXP
CD7F 300C                       JR NC,FPADD2    ;JR IF MAX EXP IN C ALREADY
CD81
CD81 48                         LD C,B
CD82 ED44                       NEG             ;A=EXP DIFF.
CD84 EB                         EX DE,HL
CD85 D9                         EXX
CD86 EB                         EX DE,HL        ;SWOP MANTISSAS OF N1 AND N2 SO THAT HL'HL
CD87                                            ;IS THE NUMBER WITH THE LARGER EXP. AND DE'DE
CD87 D9                         EXX             ;IS THE NUMBER WITH LOWER EXP (NEEDING SHIFT)
CD88 CDEFCD                     CALL ADDALIGN   ;B=0
CD8B 1809                       JR FPADD3
CD8D
CD8D C4EFCD     FPADD2:         CALL NZ,ADDALIGN ;CALL UNLESS EXPS MATCH - NO NEED TO SHIFT
CD90 EB                         EX DE,HL
CD91 D9                         EXX
CD92 EB                         EX DE,HL
CD93 D9                         EXX
CD94 0600                       LD B,0          ;BC=COMMON EXP
CD96
CD96 F1         FPADD3:         POP AF          ;SGNS DO/DONT MATCH FLAG. N1 IN HL'HL NOW
CD97 17                         RLA
CD98 381D                       JR C,FPSUBN2    ;SUBTRACT IF SIGNS MISMATCH - SUBTRACT
CD9A
CD9A 19                         ADD HL,DE
CD9B
CD9B D9                         EXX
CD9C ED5A                       ADC HL,DE
CD9E D9                         EXX
CD9F
CD9F EB                         EX DE,HL        ;RESULT IN HL'DE
CDA0 E1                         POP HL          ;N1 PTR
CDA1 E5                         PUSH HL
CDA2 23                         INC HL          ;PT TO SGN OF RESULT
CDA3 300B                       JR NC,FPADD4    ;JR IF EXP OK AS IT IS
CDA5
CDA5 03                         INC BC          ;INC MAX (COMMON) EXP
CDA6
CDA6 D9                         EXX
CDA7 CB1C                       RR H            ;BIT 7 BECOMES 1
CDA9 CB1D                       RR L
CDAB D9                         EXX
CDAC
CDAC CB1A                       RR D
CDAE CB1B                       RR E            ;NORMALIZE MANTISSA. CY IF NEEDS ROUNDING
CDB0                                            ;NOTE: -VE NUMBERS ARE ROUNDED IN THE SAM WAY,
CDB0                                            ;AS WITH MULT AND DIVN, SO BECOME MORE -VE.
CDB0
CDB0 7E         FPADD4:         LD A,(HL)
CDB1 F5                         PUSH AF         ;SGN OF RESULT
CDB2 60                         LD H,B
CDB3 69                         LD L,C          ;HL=EXP, HL'DE=RESULT, SGN AND N1 PTR ON STACK
CDB4 C387CC                     JP FPADDNEN     ;CY IF NEEDS ROUNDING UP
CDB7
CDB7
CDB7            ;SGN DON'T MATCH SO SUBTRACT. E.G. N1+(-N2) OR -N1+N2. BC=COMMON EXP
CDB7
CDB7 A7         FPSUBN2:        AND A
CDB8 ED52                       SBC HL,DE
CDBA D9                         EXX
CDBB ED52                       SBC HL,DE       ;MSW
CDBD 2009                       JR NZ,FPSUBN3   ;JR IF MSB OF RESULT IS NZ
CDBF
CDBF 08                         EX AF,AF'       ;PROTECT CY (Z,CY POSSIBLE)**
CDC0 D9                         EXX
CDC1 7C                         LD A,H
CDC2 B5                         OR L
CDC3 D9                         EXX
CDC4
CDC4 CA29CC                     JP Z,MSETZERO   ;JP IF LSW OF RESULT IS ALSO ZERO - ZERO RESULT
CDC7
CDC7 08                         EX AF,AF'       ;** SUBN BUG FIX
CDC8
CDC8            ;HL'HL=RESULT (NON-ZERO). CY WILL SHOW IF SGN NEEDS REVERSAL
CDC8
CDC8 D1         FPSUBN3:        POP DE
CDC9 D5                         PUSH DE         ;N1 PTR
CDCA 13                         INC DE
CDCB 1A                         LD A,(DE)
CDCC 3016                       JR NC,FPSUBN4   ;JR IF NO NEED TO REVERSE ORIG SGN AND NEG RESULT
CDCE
CDCE EE80                       XOR 80H         ;FLIP SGN BIT OF ORIG N1
CDD0 08                         EX AF,AF'
CDD1 EB                         EX DE,HL        ;MSW
CDD2 D9                         EXX
CDD3 EB                         EX DE,HL        ;LSW. RESULT IN DE'DE
CDD4 CDEECB                     CALL NEGDEDE    ;NEGATE DE'DE, EXIT WITH DE' SELECTED
CDD7 EB                         EX DE,HL        ;MSW
```

```
CDD8 D9                         EXX
CDD9 EB                         EX DE,HL          ;LSW. RESULT IN HL'HL
CDDA D9                         EXX
CDDB 08                         EX AF,AF'
CDDC 1806                       JR FPSUBN4
CDDE
CDDE D9           FPSUBNLP:     EXX
CDDF 0B                         DEC BC            ;EXP
CDE0 29                         ADD HL,HL
CDE1 D9                         EXX
CDE2 ED6A                       ADC HL,HL         ;SHIFT HL'HL LEFT
CDE4
CDE4 CB7C         FPSUBN4:      BIT 7,H           ;H'
CDE6 28F6                       JR Z,FPSUBNLP     ;LOOP UNTIL NUMBER IS NORMALIZED
CDE8
CDE8 D9                         EXX
CDE9 EB                         EX DE,HL          ;RESULT IN HL'DE
CDEA 60                         LD H,B
CDEB 69                         LD L,C
CDEC C39FCC                     JP FPSUBNEN       ;BIT 7,A=RESULT SGN, HL'DE=MANT, HL=EXP
CDEF                                              ;(SP)=N1 PTR
CDEF
CDEF
CDEF              ;ADDALIGN SR. ENTRY:DE'DE=MANT, A=BITS
CDEF
CDEF 47           ADDALIGN:     LD B,A
CDF0 FE21                       CP 33             ;CP BITS TO ROTATE (EXP DIFF)
CDF2 3802                       JR C,ADDALILP     ;JR IF LESSER WON'T BE SHIFTED TO ZERO
CDF4
CDF4 0621                       LD B,33           ;ENSURE DE'DE IS ROTATED TO ZERO
CDF6
CDF6 D9           ADDALILP:     EXX
CDF7 CB3A                       SRL D
CDF9 CB1B                       RR E              ;MSW
CDFB D9                         EXX
CDFC CB1A                       RR D
CDFE CB1B                       RR E              ;LSW
CE00 10F4                       DJNZ ADDALILP     ;LOOP UNTIL NUMBERS ALIGNED FOR COMMON EXP. B=0
CE02
CE02 D0                         RET NC            ;RET IF NO NEED TO ROUND UP: ALWAYS TRUE IF ZERO
CE03
CE03 1C                         INC E
CE04 C0                         RET NZ
CE05
CE05 14                         INC D
CE06 C0                         RET NZ
CE07
CE07 D9                         EXX
CE08 13                         INC DE            ;NEVER CARRIES BECAUSE BIT 7 STARTS AS 0
CE09 D9                         EXX
CE0A C9                         RET
CE0B
CE0B              ;MULT/DIV/ADD SR
CE0B              ;ENTRY: HL PTS TO FIRST OF TWO NUMBERS, DE TO SECOND. THEY ARE UNSTACKED TO:
CE0B              ;D'E'DE=(N1 MANTISSA) H'L'HL AND A'B'C'A=(N2 MANTISSA) B/C=(N1/N2 EXP)
CE0B              ;(SP)=SGN OF RESULT (BIT 7). D' AND H' HAVE TRUE NUMERIC BITS SET
CE0B
CE0B 13           MUDIADSR:     INC DE
CE0C 1A                         LD A,(DE)         ;N2 SGN
CE0D 46                         LD B,(HL)         ;N1 EXP
CE0E 23                         INC HL            ;PT TO N1 SGN
CE0F AE                         XOR (HL)          ;BIT 7 IS XORED SGN BITS - UNLIKE=1
CE10 D1                         POP DE            ;RET ADDR
CE11 F5                         PUSH AF
CE12 D5                         PUSH DE           ;RET ADDR
CE13 56                         LD D,(HL)         ;N1 MANT 1
CE14 CBFA                       SET 7,D           ;TRUE NUMERIC BIT SET
CE16 23                         INC HL
CE17 5E                         LD E,(HL)
CE18 23                         INC HL
CE19 D5                         PUSH DE           ;N1 MANT 1 AND 2
CE1A 56                         LD D,(HL)
CE1B 23                         INC HL
CE1C 5E                         LD E,(HL)         ;DE=N1 MANT 3 AND 4
CE1D 23                         INC HL
CE1E 4E                         LD C,(HL)         ;BC=N1/N2 EXP
CE1F 23                         INC HL
CE20 08                         EX AF,AF'
CE21 7E                         LD A,(HL)         ;A'=N2 MANT 1
CE22 F680                       OR 80H
CE24 08                         EX AF,AF'
CE25 23                         INC HL
CE26 7E                         LD A,(HL)
CE27 23                         INC HL
CE28
CE28 D9                         EXX
CE29 47                         LD B,A            ;B'=N2 MANT 2
CE2A 6F                         LD L,A            ;L'=N2 MANT 2 ALSO
CE2B 08                         EX AF,AF'
CE2C 67                         LD H,A            ;H'=N2 MANT 1
CE2D 08                         EX AF,AF'
CE2E D9                         EXX
CE2F
CE2F 7E                         LD A,(HL)
CE30 23                         INC HL
CE31
CE31 D9                         EXX
CE32 4F                         LD C,A            ;BC'=N2 MANT 2 AND 3
CE33 D1                         POP DE            ;DE'=N1 MANT 1 AND 2
CE34 D9                         EXX
CE35
CE35 6E                         LD L,(HL)         ;L=N2 MANT 4
CE36 67                         LD H,A            ;H=N2 MANT 3
```

```
CE37 7D                         LD A,L          ;A=N2 MANT 4
CE38 C9                         RET
CE39
CE39                ;DOUBLE FPFORM. EQU TO EX DE,HL:CALL FPFORM:EX AF,AF':EX DE,HL: JP FPFORM
CE39                ;EXIT: NUMBER AT HL AND NUMBER AT DE ARE IN FP FORM. F'=Z IF DE NUMBER IS
CE39                ;A POWER OF TWO, Z IF HL NUMBER IS.
CE39
CE39 CD3DCE         DFPFORM:    CALL FPFORMX
CE3C 08                         EX AF,AF'
CE3D
CE3D EB             FPFORMX:    EX DE,HL
CE3E
CE3E                ;CHANGE NUMBER AT (HL) TO FP FORM IF NEEDED
CE3E                ;EXIT: HL AND DE UNCHANGED, A, BC CORRUPT, Z IF NUMBER IS A POWER OF 2, CY IF 0
CE3E
CE3E 7E             FPFORM:     LD A,(HL)
CE3F A7                         AND A
CE40 2810                       JR Z,FPFORM2    ;JR IF INTEGER
CE42
CE42 23                         INC HL
CE43 7E                         LD A,(HL)
CE44 2B                         DEC HL
CE45 E67F                       AND 7FH         ;TEST MANT 1, IGNORING SGN BIT
CE47 C0                         RET NZ          ;RET IF NOT A POWER OF TWO
CE48
CE48 E5                         PUSH HL         ;NOW DO A LONGER TEST FOR POWERS OF 2. OTHER MANT
CE49 23                         INC HL          ;BYTES MUST BE ZERO
CE4A 23                         INC HL
CE4B B6                         OR (HL)         ;TEST MANT 2
CE4C 23                         INC HL
CE4D B6                         OR (HL)         ;MANT 3
CE4E 23                         INC HL
CE4F B6                         OR (HL)         ;MANT 4
CE50 E1                         POP HL
CE51 C9                         RET             ;Z MEANS NUMBER IS AN EXACT POWER OF TWO
CE52
CE52 D5             FPFORM2:    PUSH DE
CE53 CDDCCB                     CALL FETCHI     ;INTEGER IN DE, HL PTS TO MANT 3
CE56 AF                         XOR A
CE57 23                         INC HL
CE58 77                         LD (HL),A       ;MANT 4
CE59 2B                         DEC HL
CE5A 77                         LD (HL),A       ;MANT 3 AND 4 WILL BE ZERO
CE5B 2B                         DEC HL
CE5C 2B                         DEC HL          ;PT TO MANT 1
CE5D 7A                         LD A,D
CE5E 0E90                       LD C,90H
CE60 A7                         AND A
CE61 2008                       JR NZ,FPFORM3   ;JR IF INT IS >8 BITS
CE63
CE63 B3                         OR E
CE64 281A                       JR Z,FPFORM4    ;RET IF INT IS ZERO. ELSE A=E
CE66
CE66 1E00                       LD E,0          ;LS PART OF NUMBER
CE68 0E89                       LD C,89H        ;INIT EXP
CE6A
CE6A 0D             FPFORMLP:   DEC C
CE6B
CE6B CB13           FPFORM3:    RL E            ;NC INITIALLY AND LATER...
CE6D 17                         RLA
CE6E 30FA                       JR NC,FPFORMLP  ;SHIFT INTEGER TILL IT OVERFLOWS ON THE LHS
CE70
CE70 CB16                       RL (HL)         ;SGN TO CY
CE72 1F                         RRA
CE73 CB1B                       RR E            ;ROT. SGN BIT INTO AB - REPLACES TRUE NUMERIC BIT
CE75 77                         LD (HL),A
CE76 23                         INC HL
CE77 73                         LD (HL),E
CE78 2B                         DEC HL
CE79 2B                         DEC HL
CE7A 71                         LD (HL),C       ;PLACE EXPONENT
CE7B E67F                       AND 7FH         ;IGNORE SGN BIT
CE7D B3                         OR E            ;Z IF POWER OF TWO
CE7E D1                         POP DE
CE7F C9                         RET
CE80
CE80 2B             FPFORM4:    DEC HL
CE81 D1                         POP DE
CE82 37                         SCF             ;'ZERO'
CE83 C9                         RET
CE84
CE84                        INCLUDE ROM1FNS.SAM    ;POINT
CE84                ;ROM1FNS.SAM. CODE, LEN, INKEY$, BUTTON, SVAR, CHR$, BIN$, HEX$, MEM$, STR$,
CE84                ;CONCAT, SQR, ABS, NEGATE, SGN, INT, TRUNCATE, ATTR, POINT
CE84                ;*************************************************************************
CE84                ;
CE84
CE84 3E             FPVAL:      DB 3EH          ;'LD A,0AFH'
CE85
CE85 AF             FPVALS:     XOR A
CE86 DDE5                       PUSH IX         ;** BUG FIX
CE88 ED4B955A                   LD BC,(CHADP-1) ;B=CHADP
CE8C C5                         PUSH BC
CE8D 2A975A                     LD HL,(CHAD)
CE90 E5                         PUSH HL
CE91 A7                         AND A
CE92 F5                         PUSH AF         ;A=0 IF VAL$, NZ IF VAL
CE93                        ;  CALL UNSTKPRT    ;DE=START, BC=LEN, A=PORT
CE93 CDDC3F                     CALL GETSTRING
CE96 03                         INC BC          ;LEN OF AT LEAST 1; ROOM FOR TERMINATOR
CE97                        ; LD (BCSTORE),BC
CE97                ;           CALL R1OFFCL    ;**
CE97 CD4D3F                     CALL SCOPYWK    ;COPY STRING PLUS 1 BYTE OF JUNK TO WKSPACE
CE9A                    ;    DEC HL             ;TERMINATE WITH 0DH
```

```
CE9A                    ;     LD (HL),0DH       ;TERMINATE STRING COPY IN WKSPACE
CE9A ED53975A                 LD (CHAD),DE
CE9E DBFB                     IN A,(URPORT)
CEA0 32965A                   LD (CHADP),A
CEA3 D5                       PUSH DE
CEA4 CD0102                   CALL R1OFFCL
CEA7 7538                     DW TOKDE          ;TOKENISE FROM DE ON
CEA9 E1                       POP HL
CEAA 22975A                   LD (CHAD),HL
CEAD E5                       PUSH HL
CEAE 213B5C                   LD HL,FLAGS
CEB1 CBBE                     RES 7,(HL)        ;'SYNTAX CHECK' SO FP FORMS INSERTED
CEB3 CD083B                   CALL EXPTEXPR     ;RETURN WITH Z FOR STRING, NZ FOR NUM.
CEB6 08                       EX AF,AF'
CEB7 7E                       LD A,(HL)         ;TERMINATOR
CEB8 FE0D                     CP 0DH
CEBA 2027                     JR NZ,VALNONS     ;CHECK FOR TERMINATOR
CEBC
CEBC E1                       POP HL            ;STRING START IN WKSPACE
CEBD F1                       POP AF            ;VAL/VAL$ FLAG
CEBE 2003                     JR NZ,FPVAL2      ;JR IF VAL
CEC0
CEC0 08                       EX AF,AF'
CEC1 2803                     JR Z,FPVALOK      ;IF VAL$, STRING RESULT IS OK. ELSE EX AF, AF'
CEC3                                            ;GIVES Z, THEN ERROR
CEC3 08           FPVAL2:     EX AF,AF'
CEC4 281D                     JR Z,VALNONS      ;IF VAL, STRING EXPR IS AN ERROR
CEC6
CEC6 22975A       FPVALOK:    LD (CHAD),HL
CEC9 213B5C                   LD HL,FLAGS
CECC CBFE                     SET 7,(HL)        ;'RUNNING'
CECE CDE314                   CALL SCANNING     ;GET EXPR TO FPC STACK
CED1 E1                       POP HL
CED2 22975A                   LD (CHAD),HL      ;ORIGINAL CHAD
CED5 F1                       POP AF
CED6 32965A                   LD (CHADP),A
CED9 CDDF3F                   CALL SELURPG
CEDC DDE1                     POP IX            ;** BUG FIX
CEDE
CEDE ED5B655C     SETUPDE:    LD DE,(STKEND)
CEE2 C9                       RET
CEE3
CEE3 CF           VALNONS:    RST 08H
CEE4 1D                       DB 29
CEE5
CEE5 CF           FPDVAR:     RST 08H           ;LET DOS STACK ADDR OF ITS VARS
CEE6 8B                       DB DVHK
CEE7 21                       DB 21H            ;'JR+2'
CEE8
CEE8 CF           FPEOF:      RST 08H
CEE9 8C                       DB EOFHK          ;DOS EOF
CEEA 21                       DB 21H            ;'JR+2'
CEEB
CEEB CF           FPPTR:      RST 08H
CEEC 8D                       DB PTRHK          ;DOS PTR
CEED 1805                     JR SUDH
CEEF
CEEF CD143B       IMPATHS:    CALL SABORTER
CEF2
CEF2 CF                       RST 08H
CEF3 8E                       DB PATHHK
CEF4
CEF4 18E8         SUDH:       JR SETUPDE
CEF6
CEF6
CEF6 E7           IMRND:      RST 20H           ;SKIP 'RND'
CEF7 CDC63A                   CALL RUNFLG
CEFA 3024                     JR NC,IMRND4      ;JR IF NOT RUNNING
CEFC
CEFC 0600                     LD B,0
CEFE 2A765C                   LD HL,(SEED)
CF01 1EFD                     LD E,0FDH
CF03 55                       LD D,L
CF04 7C                       LD A,H
CF05 29                       ADD HL,HL
CF06 98                       SBC A,B
CF07 EB                       EX DE,HL
CF08 ED52                     SBC HL,DE
CF0A 98                       SBC A,B
CF0B 4F                       LD C,A
CF0C ED42                     SBC HL,BC
CF0E 3001                     JR NC,IMRND1
CF10
CF10 23                       INC HL
CF11
CF11 22765C       IMRND1:     LD (SEED),HL
CF14 CDD61C                   CALL STACKHL
CF17
CF17 EF                       DB CALC
CF18 31                       DB RESTACK        ;EXP WILL BE 00 IF ZERO, ELSE 81-90H
CF19 33                       DB EXIT
CF1A
CF1A 7E                       LD A,(HL)
CF1B D610                     SUB 10H           ;DIVIDE BY 65536
CF1D 3801                     JR C,IMRND4       ;LEAVE IT ALONE IF ZERO
CF1F
CF1F 77                       LD (HL),A         ;NEW EXP
CF20
CF20 DF           IMRND4:     RST 18H
CF21 FE28                     CP "("
CF23 C0                       RET NZ
CF24
CF24 CDC03A                   CALL SEX1NUMCB    ;SKIP, GET 'N)', PASS CLOSING BRACKET
```

```
CF27 D0                    RET NC            ;RET IF NOT RUNNING
CF28
CF28 EF                    DB CALC           ;RND,PARAM
CF29 E9                    DB STKONE         ;RND,P,1
CF2A 01                    DB ADDN           ;RND,P+1
CF2B 00                    DB MULT           ;RND*(P+1)
CF2C 30                    DB TRUNC
CF2D 34                    DB EXIT2          ;EXIT WITH INTEGER BETWEEN 0 AND PARAM
CF2E
CF2E             ;ATTR(L,C)
CF2E
CF2E CDAE3A      IMATTR:   CALL EXB2NUMB  ;CHECK (X,Y). CY IF RUNNING
CF31 D0                    RET NC
CF32
CF32 212018                LD HL,1820H       ;LINE/COL LIMITS (PLUS 1)
CF35 CDA4CF                CALL GETCP        ;GET CHAR POSITION WITHIN LIMITS, IN DE
CF38 3A405A                LD A,(MODE)
CF3B FE02                  CP 2
CF3D D29B12                JP NC,INVMERR      ;INVALID MODE UNLESS 0 OR 1
CF40
CF40 A7                    AND A
CF41 2808                  JR Z,DOATT2       ;JR IF MODE 0
CF43
CF43 CDD03D                CALL M1DEADDR     ;GET CHAR ADDR
CF46 7A                    LD A,D
CF47 C620                  ADD A,20H         ;GET ATTR ADDR (ADD 8K)
CF49 180B                  JR DOATT3
CF4B
CF4B CDE43D      DOATT2:   CALL M0DEADDR
CF4E 7A                    LD A,D
CF4F 0F                    RRCA
CF50 0F                    RRCA
CF51 0F                    RRCA
CF52 E603                  AND 03H
CF54 F698                  OR 98H            ;FORM ATTR ADDR IN AE
CF56
CF56 57          DOATT3:   LD D,A
CF57 EB                    EX DE,HL
CF58 CDBB3F                CALL SREAD
CF5B 1842                  JR POATC
CF5D
CF5D             ;POINT(X,Y)
CF5D
CF5D CDAE3A      IMPOINT:  CALL EXB2NUMB  ;CHECK (X,Y). CY IF RUNNING
CF60 D0                    RET NC
CF61
CF61 CD9327                CALL GTFCOORDS    ;GET COORDS IN B,C OR B,HL (THIN) AND CY
CF64 CD213E                CALL ANYPIXAD     ;GET ADDRESS IN HL, PIXEL OFFSET IN A
CF67 3C                    INC A
CF68 47                    LD B,A            ;B=X MOD 8+1
CF69 CDBB3F                CALL SREAD
CF6C 5F                    LD E,A            ;E=SCREEN BYTE
CF6D 1601                  LD D,01H          ;MODE 0/1 MASK
CF6F 3A405A                LD A,(MODE)
CF72 FE02                  CP 2
CF74 381B                  JR C,DPOINTC      ;JR IF MODE 0/1
CF76
CF76 160F                  LD D,0FH          ;MODE 3 MASK
CF78 201D                  JR NZ,M3POINT     ;JR IF MODE 3
CF7A
CF7A 1603                  LD D,3            ;MODE 2 MASK
CF7C 7B                    LD A,E
CF7D 17                    RLA
CF7E CB1B                  RR E
CF80 AB                    XOR E
CF81 E6AA                  AND 0AAH
CF83 AB                    XOR E
CF84 5F                    LD E,A            ;SWAP ODD/EVEN BITS IN MODE 2
CF85
CF85 3A4D5A                LD A,(THFATT)
CF88 A7                    AND A
CF89 200C                  JR NZ,M3POINT     ;DO MODE 3 POINT IF FATPIX 1
CF8B
CF8B 78                    LD A,B
CF8C 3D                    DEC A
CF8D A2                    AND D             ;AND 3
CF8E 3C                    INC A
CF8F 87                    ADD A,A           ;IF THIN PIX, A=2/4/6/8 FOR PIX 0/1/2/3
CF90 47                    LD B,A
CF91
CF91 7B          DPOINTC:  LD A,E
CF92
CF92 07          POINLP:   RLCA
CF93 10FD                  DJNZ POINLP
CF95
CF95 1807                  JR M3ODPT
CF97
CF97 7B          M3POINT:  LD A,E
CF98 0604                  LD B,4
CF9A CB41                  BIT 0,C           ;SEE IF X IS ODD
CF9C 28F4                  JR Z,POINLP
CF9E
CF9E A2          M3ODPT:   AND D
CF9F
CF9F CDC63F      POATC:    CALL RCURP
CFA2 1845                  JR STKAB          ;ATTR OR POINT
CFA4
CFA4
CFA4             ;GET CHAR POSITION LESS THAN LIMITS H/L (LINE/COL) IN DE (LINE/COL)
CFA4
CFA4 E5          GETCP:    PUSH HL           ;LINE/COL MAX
CFA5 CD331D                CALL GETBYTE      ;COL
CFA8 E1                    POP HL
```

```
CFA9 BD                 CP L
CFAA 300A               JR NC,OSERR
CFAC
CFAC C5                 PUSH BC
CFAD E5                 PUSH HL
CFAE CD331D             CALL GETBYTE      ;LINE
CFB1 E1                 POP HL
CFB2 D1                 POP DE
CFB3 57                 LD D,A
CFB4 BC                 CP H
CFB5 D8                 RET C
CFB6
CFB6 CF         OSERR:  RST 08H
CFB7 20                 DB 32             ;'Off screen'
CFB8
CFB8
CFB8            ;*************************************************************************
CFB8            ;INKEY$ #N
CFB8
CFB8 111E11     FPINKEY:LD DE,1100H+30    ;LIMIT <17
CFBB CD621F             CALL LIMBYTE
CFBE
CFBE 2A515C             LD HL,(CURCHL)
CFC1 E5                 PUSH HL
CFC2 CD6311             CALL SETSTRM      ;SET STREAM 'A'
CFC5 CDBA01             CALL INPUTAD      ;SCAN INPUT STREAM WITHOUT WAITING
CFC8 E1                 POP HL            ;PREV. CHANNEL
CFC9 F5                 PUSH AF           ;INKEY VALUE
CFCA CD7011             CALL CHANFLAG     ;RESTORE ORIG CHANNEL
CFCD F1                 POP AF
CFCE
CFCE 57         FPINKEN:LD D,A
CFCF 3825               JR C,CHRINKC      ;IF GOT KEY, D=BYTE. COPY TO WKSPACE
CFD1
CFD1 AF                 XOR A             ;IF NO INPUT - NULL STRING
CFD2 1815               JR STKAB
CFD4
CFD4
CFD4            ;MOUSE BUTTON N (1/2/3) RETURNS 1 IF PRESSED, 0 IF NOT
CFD4
CFD4 111E04     FPBUTTON:LD DE,0400H+30   ;LIMIT TO 0-3
CFD7 CD621F             CALL LIMBYTE
CFDA FE03               CP 3              ;CY IF 0/1/2, NC IF 3
CFDC DEFF               SBC A,0FFH        ;00,01,02,04
CFDE 2001               JR NZ,FPBT2
CFE0
CFE0 2F                 CPL               ;'BUTTON 0' TEST ALL 3
CFE1
CFE1 218F5B     FPBT2:  LD HL,BUTSTAT     ;HAS BITS 2-0 SET FOR BUTTONS 3-1
CFE4 A6                 AND (HL)
CFE5 2802               JR Z,STKAB
CFE7
CFE7 3E01               LD A,1
CFE9
CFE9 C3DA1C     STKAB:  JP STACKA
CFEC
CFEC            ;SVAR JUST ADDS BASE ADDR OF SYSTEM VARS TO ARGUMENT
CFEC
CFEC EF         FPSVAR: DB CALC
CFED 2A                 DB LKADDRW
CFEE 44FE               DW VVAR2
CFF0 01                 DB ADDN
CFF1 34                 DB EXIT2
CFF2
CFF2            ;*************************************************************************
CFF2            ;CHR$ - CREATE 1 CHAR STRING IN WKSPACE, PARAMS ON FPCS
CFF2
CFF2 CD331D     FPCHRS: CALL GETBYTE      ;B=0, L=BYTE
CFF5 55                 LD D,L
CFF6
CFF6 010100     CHRINKC:LD BC,1           ;LEN=1
CFF9
CFF9 21C95A     INKYEN: LD HL,TEMPW1+1
CFFC 73                 LD (HL),E
CFFD 2B                 DEC HL
CFFE 72                 LD (HL),D
CFFF C3673C             JP CWKSTK         ;COPY BC FROM (HL) TO WKSPACE, PARAMS TO FPCS
D002                    ;EXITS WITH DE=STKEND
D002
D002            ;*************************************************************************
D002            ;BIN$ N. RETURN BINARY STRING 8 OR 16 DIGITS LONG
D002
D002 CD2E1D     FPBINS: CALL GETINT
D005 EB                 EX DE,HL          ;INT IN DE
D006 7A                 LD A,D
D007 0E08               LD C,8            ;ASSUME LEN 8 FOR RESULT STRING
D009 A7                 AND A
D00A 2802               JR Z,BINS2
D00C
D00C 0E10               LD C,16           ;USE 16 IF NEEDED
D00E
D00E 213151     BINS2:  LD HL,MEMVAL+16   ;GET HL POINTING TEMP $ BUFFER IN MEMS
D011 41                 LD B,C
D012
D012 2B         BINSLP: DEC HL
D013 CB3A               SRL D
D015 CB1B               RR E              ;SHIFT LS BITS OUT FIRST
D017 3A035A             LD A,(BIN1DIG)
D01A 3803               JR C,BINS4        ;LEAVE IT IF CY
D01C
D01C 3A045A             LD A,(BIN0DIG)
D01F
D01F 77         BINS4:  LD (HL),A
```

```
D020 10F0                  DJNZ BINSLP
D022                                       ;HL=START, BC=LEN
D022 C3673C      BCWKHP:   JP CWKSTK        ;COPY BC FROM (HL) TO WKSPACE, PARAMS TO FPCS
D025
D025             ;*********************************************************************
D025             ;HEX$ N. RETURN HEX VERSION OF N, 2, 4 OR 6 DIGITS LONG, ACCORDING TO MAGNITUDE
D025
D025 CD8C3F      FPHEXS:   CALL UNSTLEN     ;ABC=PAGE/'ADDR' **
D028 0F                    RRCA
D029 0F                    RRCA            ;LS 2 BITS TO POSN 7,6
D02A 67                    LD H,A
D02B A8                    XOR B
D02C E6C0                  AND 0C0H
D02E A8                    XOR B
D02F 47                    LD B,A          ;HBC=20-BIT NUMBER
D030 1E00                  LD E,0          ;INIT DIGIT COUNTER
D032 B4                    OR H
D033 2811                  JR Z,HEX2DIG    ;JR IF MS 2 BYTES (OF 3)=0
D035
D035 7C                    LD A,H
D036 E60F                  AND 0FH
D038 2809                  JR Z,HEX4DIG    ;JR IF MSB=0
D03A
D03A 08                    EX AF,AF'
D03B 78                    LD A,B
D03C 08                    EX AF,AF'       ;SAVE MIDDLE BYTE IN A'
D03D 47                    LD B,A          ;B=MS BYTE
D03E CD58D0                CALL HEXASCSR
D041 08                    EX AF,AF'
D042 47                    LD B,A          ;MIDDLE BYTE
D043
D043 CD58D0      HEX4DIG:  CALL HEXASCSR   ;DERIVE 2 DIGITS FROM B, STACK THEM
D046
D046 41          HEX2DIG:  LD B,C          ;LS BYTE
D047 CD58D0                CALL HEXASCSR
D04A 4B                    LD C,E
D04B 0600                  LD B,0
D04D 212751                LD HL,MEMVAL+6  ;TEMP BUFFER
D050 41                    LD B,C
D051
D051 2B          HEXPUTLP: DEC HL
D052 F1                    POP AF          ;UNSTACK AN ASCII DIGIT
D053 77                    LD (HL),A
D054 10FB                  DJNZ HEXPUTLP
D056                                       ;HL=START-1, BC=LEN
D056 18CA                  JR BCWKHP       ;STACK STRING FROM HL ON FPCS
D058
D058
D058             ;HEX$ ASCII SR TO CONVERT VALUE IN B TO 2 ASCII HEX DIGITS ON STACK
D058
D058 1C          HEXASCSR: INC E
D059 1C                    INC E           ;COUNT OF DIGITS ON STACK=COUNT+2
D05A 78                    LD A,B
D05B 1602                  LD D,2          ;2 DIGITS
D05D E1                    POP HL          ;RET ADDR
D05E 0F                    RRCA            ;MS NIBBLE FIRST
D05F 0F                    RRCA
D060 0F                    RRCA
D061 0F                    RRCA
D062
D062 E60F        HEXSRLP:  AND 0FH
D064 FE0A                  CP 0AH
D066 DE69                  SBC A,69H       ;
D068 27                    DAA             ;A STRANGE TWIDDLE THAT GIVES THE RIGHT ANSWER!
D069 F5                    PUSH AF         ;SAVE ASCII 0-9, A-F
D06A 78                    LD A,B
D06B 15                    DEC D
D06C 20F4                  JR NZ,HEXSRLP
D06E
D06E E9                    JP (HL)         ;RETURN
D06F
D06F             ;*********************************************************************
D06F             ;PT 2 OF MEMORY$
D06F
D06F EF          MEMRYSP2: DB CALC
D070 06                    DB SWOP         ;N2,N1
D071 D0                    DB STO0
D072 03                    DB SUBN         ;N2-N1
D073 E9                    DB STKONE       ;ALLOW FOR INCLUSIVE CHAR.
D074 01                    DB ADDN         ;LEN
D075 25                    DB DUP
D076 2C                    DB LESS0        ;LEN, 1/0
D077 1F                    DB JPFALSE      ;JP IF LEN <0
D078 03                    DB 03H
D079
D079 07                    DB DROP
D07A E1                    DB STKZERO      ;NULL STRING IF LEN <0
D07B
D07B D8                    DB RCL0         ;LEN,N1
D07C 33                    DB EXIT
D07D
D07D CD8C3F                CALL UNSTLEN    ;GET AHL=START
D080 CBFC                  SET 7,H         ;PAGE FORM
D082 3D                    DEC A
D083 F5                    PUSH AF
D084 E5                    PUSH HL
D085 CD2E1D                CALL GETINT     ;GET BC=LEN
D088 D1                    POP DE          ;ADDR
D089 F1                    POP AF
D08A C3EB1C                JP STKSTOS
D08D
D08D             ;*********************************************************************
D08D             ;S1+S2. PRODUCE A CONCATENATED STRING IN WKSPACE, PARAMS ON FPCS
```

```
D08D            ;BINARY - ALTHOUGH ONLY RELEVANT FOR ENTRY DE
D08D
D08D DBFB       FPCONCAT:  IN A,(URPORT)
D08F F5                    PUSH AF
D090 D5                    PUSH DE            ;S2 PTR WILL BE NEW STKEND
D091 CD011D                CALL STKFETCH      ;ADE=S2 START, BC=S2 LEN
D094 F5                    PUSH AF
D095 D5                    PUSH DE
D096 C5                    PUSH BC            ;S2 LEN
D097 C5                    PUSH BC
D098 CD011D                CALL STKFETCH
D09B E1                    POP HL             ;S2 LEN
D09C 09                    ADD HL,BC          ;TOTAL LEN
D09D DA283A                JP C,STLERR        ;** BUG FIX
D0A0
D0A0 F5                    PUSH AF            ;PAGE OF S1
D0A1 D5                    PUSH DE            ;START OF S1
D0A2 C5                    PUSH BC            ;LEN OF S1
D0A3 44                    LD B,H
D0A4 4D                    LD C,L             ;BC=TOTAL LEN
D0A5 CD801E                CALL WKROOM        ;GET DE=START, BC=TOTAL LEN, PAGED IN
D0A8 CDE71C                CALL STKSTOREP     ;PARAMS OF STRING TO BE CREATED TO FPCS
D0AB
D0AB C1                    POP BC             ;LEN OF S1
D0AC CD791F                CALL SPLITBC       ;SET UP PAGCOUNT/MODCOUNT
D0AF DBFB                  IN A,(251)
D0B1 4F                    LD C,A             ;DEST=CDE
D0B2 E1                    POP HL             ;START OF S1
D0B3 F1                    POP AF             ;PAGE OF S1
D0B4 CD5E2A                CALL FARLDIR
D0B7 C1                    POP BC             ;LEN OF S2
D0B8 CD791F                CALL SPLITBC
D0BB D1                    POP DE             ;START OF S2
D0BC E1                    POP HL             ;H=PAGE OF S2
D0BD 37                    SCF
D0BE CD682A                CALL FARLDIR2      ;ENTRY PT 2 BECAUSE TEMPW1 AND TEMPB2 = DEST
D0C1 D1                    POP DE             ;NEW STKEND
D0C2 C3453C                JP PPORT
D0C5
D0C5            ;AMPERSAND-PREFACED HEX, E.G. &FC0D OR &12345 (6 DIGITS OK)
D0C5            ;VALUE EVALUATED DURING SYNTAX CHECK AND INSERTED AS A 5-BYTE FORM.
D0C5
D0C5 AF         AMPERSAND: XOR A
D0C6 67                    LD H,A
D0C7 6F                    LD L,A             ;INIT RESULT TO ZERO
D0C8 08                    EX AF,AF'          ; IN A'HL
D0C9
D0C9 EB         AMPDILP:   EX DE,HL
D0CA E7                    RST 20H
D0CB EB                    EX DE,HL           ;HL=RESULT
D0CC D630                  SUB 30H            ;NUMS NOW 00-09, LETS 11-2A, 31-4A
D0CE 3821                  JR C,AMPEND
D0D0
D0D0 FE0A                  CP 0AH
D0D2 380E                  JR C,AMPVALID      ;JR IF DIGIT
D0D4
D0D4 C630                  ADD A,30H
D0D6 F620                  OR 20H             ;LETS NOW 61-7A
D0D8 FE61                  CP "a"
D0DA 3815                  JR C,AMPEND
D0DC
D0DC FE67                  CP "g"
D0DE 3011                  JR NC,AMPEND
D0E0
D0E0 D657                  SUB 57H            ;a-f->0A-0F
D0E2
D0E2 0604       AMPVALID:  LD B,4
D0E4 08                    EX AF,AF'
D0E5
D0E5 29         AMPERLP:   ADD HL,HL
D0E6 17                    RLA
D0E7 DAD117                JP C,NTLERR        ;NUM. TOO LARGE (ONLY SEE MSG WITH EG VAL("&"+A$))
D0EA
D0EA 10F9                  DJNZ AMPERLP
D0EC
D0EC 08                    EX AF,AF'
D0ED B5                    OR L
D0EE 6F                    LD L,A             ;ADD IN NEW HEX DIGIT
D0EF 18D8                  JR AMPDILP
D0F1
D0F1 44         AMPEND:    LD B,H
D0F2 4D                    LD C,L
D0F3 08                    EX AF,AF'
D0F4 A7                    AND A
D0F5 283A                  JR Z,STACKBCH      ;JR IF SMALL INTEGER
D0F7
D0F7 0698                  LD B,98H           ;EXP FOR FF FF FF
D0F9
D0F9 CB7F       AMPALLP:   BIT 7,A
D0FB 2004                  JR NZ,AMPFP
D0FD
D0FD 29                    ADD HL,HL          ;ALIGN MANTISSA IN AHL
D0FE 17                    RLA
D0FF 10F8                  DJNZ AMPALLP       ;B NEVER HITS ZERO!
D101
D101 E67F       AMPFP:     AND 7FH            ;+VE SIGN BIT
D103 5F                    LD E,A
D104 54                    LD D,H
D105 4D                    LD C,L
D106 78                    LD A,B             ;EXP
D107 0600                  LD B,0           ;FP NUM IN A E D C B
D109 C3F01C                JP STKSTORE
D10C
```

```
D10C CDFA01     FPUSRS:     CALL R1OFFJP      ;JP TO MAIN ROUTINE IN ROM 0 WITH ROM1 OFF
D10F F717                   DW R0USRS
D111
D111 CDFA01     FPUSR:      CALL R1OFFJP
D114 FB17                   DW R0USR
D116
D116            ;UNARY
D116 CD6F12     FPPEEK:     CALL PDPSUBR      ;GET HL=ADDR (PAGED IN), A=ORIG URPORT
D119 57                     LD D,A
D11A AF                     XOR A
D11B 180C                   JR FPPDPC
D11D
D11D            ;UNARY
D11D DBFB       FPDPEEK:    IN A,(URPORT)
D11F F5                     PUSH AF
D120 CD9F3F                 CALL NPDPS        ;GET HL=ADDR
D123 23                     INC HL
D124 CD5602                 CALL R1OFRD
D127 2B                     DEC HL
D128 D1                     POP DE
D129
D129 47         FPPDPC:     LD B,A
D12A CD5602                 CALL R1OFRD
D12D 4F                     LD C,A
D12E 7A                     LD A,D
D12F
D12F D3FB       OSBC:       OUT (251),A
D131
D131 C3DD1C     STACKBCH:   JP STACKBC
D134            ;*************************************************************************
D134            ;TRUNC$ - RETURN STRING STRIPPED OF TRAILING SPACES (E.G. FROM ARRAYS)
D134            ; E.G. LET A$=TRUNC$ B$(34)
D134            ;ENTRY WITH STRING ON FPCS
D134
D134 CDE43E     FPTRUSTR:   CALL SBUFFET      ;DE=START, A AND BC=LEN, PAGING UNCHANGED
D137                                          ;LEN 1-255
D137 62                     LD H,D
D138 6B                     LD L,E
D139 09                     ADD HL,BC         ;PT PAST END OF STRING
D13A 3E20                   LD A," "
D13C
D13C 2B         TRUNCSLP:   DEC HL
D13D BE                     CP (HL)
D13E 2008                   JR NZ,FPSTRS2     ;JR WITH DE=START, BC=LEN
D140
D140 0D                     DEC C
D141 20F9                   JR NZ,TRUNCSLP    ;TRUNC$ OF ALL-SPACES STRING=NULL STRING
D143
D143 18EC                   JR STACKBCH
D145
D145            ;STR$
D145 CD97D9     FPSTRS:     CALL PFSTRS       ;GET NUMBER AS BC DIGITS AT (DE) IN COMMON MEM
D148
D148            ;FROM TRUNC$
D148
D148 EB         FPSTRS2:    EX DE,HL
D149 C3673C                 JP CWKSTK
D14C
D14C            ;CODE RETURNS ASCII CODE OF FIRST CHAR OF STRING ON FPCS, OR 0 IF STRING IS NULL
D14C
D14C DBFB       FPCODE:     IN A,(251)
D14E F5                     PUSH AF
D14F CDDC3F                 CALL GETSTRING    ;UNSTACK STRING. DE=START, BC=LEN, PAGED IN
D152 78                     LD A,B
D153 B1                     OR C
D154 2804                   JR Z,FPCODE2      ;JR IF NUL STRING
D156
D156 1A                     LD A,(DE)
D157 4F                     LD C,A
D158 0600                   LD B,0
D15A
D15A F1         FPCODE2:    POP AF
D15B 18D2                   JR OSBC
D15D
D15D
D15D            ;LEN RETURNS STRING LENGTH
D15D
D15D CD011D     FPLEN:      CALL STKFETCH     ;BC=LEN, NO PAGING
D160 18CF                   JR STACKBCH
D162
D162            ;SQUARE ROOT (W.E. THOMSON)
D162
D162 EF         FPSQR:      DB CALC
D163 31                     DB RESTACK        ;USE FULL F.P. FORMS
D164 D0                     DB STO0
D165 33                     DB EXIT
D166
D166 7E                     LD A,(HL)
D167 A7                     AND A
D168 C8                     RET Z             ;RET IF SQR(0) WITH ZERO ON FPCS, DE=STKEND
D169
D169 C680                   ADD A,80H         ;GET TRUE EXPONENT. CY IF WAS >=80H
D16B 1F                     RRA               ;/2, WITH BIT 7 AS ORIGINAL
D16C 77                     LD (HL),A
D16D 23                     INC HL
D16E 7E                     LD A,(HL)         ;FETCH SGN BIT
D16F 17                     RLA
D170 DAE83E                 JP C,INVARG       ;ERROR IF SQR OF -VE NUMBER
D173
D173 367F                   LD (HL),7FH       ;MANTISSA STARTS AT ABOUT ONE
D175 0605                   LD B,5            ;5 ITERATIONS
D177
```

```
D177 EF         SQURLP:     DB CALC
D178 25                     DB DUP              ;X,X
D179 D8                     DB RCL0             ;X,X,N
D17A 06                     DB SWOP             ;X,N,X
D17B 05                     DB DIVN             ;X,N/X
D17C 01                     DB ADDN             ;X+N/X
D17D 33                     DB EXIT
D17E
D17E 35                     DEC (HL)            ;DEC EXPONENT (HALVE VALUE)
D17F 10F6                   DJNZ SQURLP
D181
D181 C9                     RET                 ;DE=STKEND
D182
D182            ;*************************************************************************
D182            ;ABS FPCS TOP ENTRY - UNARY FPCS FUNCTION.
D182
D182 7E         FPABS:      LD A,(HL)
D183 23                     INC HL
D184 A7                     AND A
D185 2005                   JR NZ,FPABS2        ;JR IF FP
D187
D187 7E                     LD A,(HL)           ;SGN BYTE
D188 3C                     INC A
D189 C0                     RET NZ              ;RET IF SIGN WAS 0 (+VE), OK
D18A                                            ;ELSE SIGN WAS FF, NOW A=0
D18A 1811                   JR NEGABSC
D18C
D18C CBBE       FPABS2:     RES 7,(HL)          ;SIGN=POS.
D18E C9                     RET
D18F
D18F
D18F            ;*************************************************************************
D18F            ;NEGATE FPCS TOP ENTRY. UNARY FPCS FUNCTION.
D18F
D18F 7E         FPNEGAT:    LD A,(HL)
D190 23                     INC HL
D191 A7                     AND A
D192 2017                   JR NZ,FPNEGAT2      ;JR IF FP
D194
D194 23                     INC HL
D195 B6                     OR (HL)
D196 23                     INC HL
D197 B6                     OR (HL)             ;TEST FOR INTEGER=0
D198 C8                     RET Z               ;DO NOTHING IF SO
D199
D199 2B                     DEC HL
D19A 2B                     DEC HL
D19B 7E                     LD A,(HL)           ;SGN
D19C 2F                     CPL
D19D
D19D 77         NEGABSC:    LD (HL),A           ;REVERSE SIGN (OR MAKE +VE IF ABS)
D19E 23                     INC HL
D19F 7E                     LD A,(HL)
D1A0 2F                     CPL
D1A1 4F                     LD C,A
D1A2 23                     INC HL
D1A3 7E                     LD A,(HL)
D1A4 2F                     CPL
D1A5 47                     LD B,A
D1A6 03                     INC BC              ;NEGATE INTEGER
D1A7 70                     LD (HL),B
D1A8 2B                     DEC HL
D1A9 71                     LD (HL),C           ;LOAD IT BACK
D1AA C9                     RET
D1AB
D1AB 3E80       FPNEGAT2:   LD A,80H
D1AD AE                     XOR (HL)            ;REVERSE SIGN BIT
D1AE 77                     LD (HL),A
D1AF C9                     RET
D1B0
D1B0            ;*************************************************************************
D1B0            ;RETURN SGN N1 (-1, 0 OR 1). UNARY
D1B0
D1B0 CD52C9     FPSGN:      CALL TSTZERO2
D1B3 EB                     EX DE,HL
D1B4 C8                     RET Z               ;SGN ZERO=0
D1B5
D1B5 D5                     PUSH DE
D1B6 23                     INC HL
D1B7 7E                     LD A,(HL)
D1B8 17                     RLA
D1B9 2B                     DEC HL
D1BA
D1BA 9F                     SBC A,A             ;-VE=FF, +VE=00
D1BB 4F                     LD C,A
D1BC 110100                 LD DE,0001H
D1BF C3C8CB                 JP STOREI           ;STORE SIGNED INTEGER, POP DE, RET
D1C2
D1C2            ;*************************************************************************
D1C2            ;INT - EQUIVALENT TO TRUNCATE (ZERO BITS AFTER BINARY POINT) IF +VE
D1C2            ;FOR -VE, TRUNC E.G. -5.9=-5.0, SO SUB 1 TO GET -6.9 -> 6.0
D1C2
D1C2 7E         FPINT:      LD A,(HL)
D1C3 A7                     AND A
D1C4 C8                     RET Z               ;RET IF INTEGER ALREADY (OR ZERO)
D1C5
D1C5 EF                     DB CALC
D1C6 25                     DB DUP              ;N1,N1
D1C7 2E                     DB GRTR0            ;N1,(0 OR 1)
D1C8 1E                     DB JPTRUE           ;JP IF +VE, FPINTP
D1C9 0B                     DB 0BH              ;** BUG FIX
D1CA
D1CA 25                     DB DUP              ;N,N
```

```
D1CB 30                 DB TRUNC          ;N,TRUNC N
D1CC 25                 DB DUP            ;N,TRUNC N,TRUNC N
D1CD 1C                 DB SWOP13         ;TRUNC N,TRUNC N,N
D1CE 03                 DB SUBN           ;TRUNC N,0 IF N WAS A WHOLE NUMBER
D1CF 1F                 DB JPFALSE        ;TRUNC N. JP IF VALUE WAS INTEGER, TO EXIT2
D1D0 03                 DB 03H
D1D1
D1D1 E9                 DB STKONE         ;TRUNC N,1
D1D2 03                 DB SUBN           ;TRUNC N-1
D1D3 34                 DB EXIT2
D1D4
D1D4 30         FPINTP: DB TRUNC          ;INT(N)
D1D5 34                 DB EXIT2
D1D6
D1D6            ;*************************************************************************
D1D6            ;TRUNCATE - ZERO THE BITS AFTER BINARY POINT, FORCE INTEGER FORM IF POSSIBLE.
D1D6            ;UNARY
D1D6
D1D6 7E         FPTRUNCT: LD A,(HL)
D1D7 A7                 AND A
D1D8 C8                 RET Z             ;RET IF INTEGER
D1D9
D1D9 0608               LD B,8            ;FOR LATER
D1DB FE81               CP 81H
D1DD 3831               JR C,TRUNC0       ;JR IF LESS THAN 1
D1DF
D1DF FE91               CP 91H
D1E1 3027               JR NC,TRUNCLG     ;JR IF >=65536
D1E3
D1E3 D5                 PUSH DE
D1E4 23                 INC HL
D1E5 56                 LD D,(HL)
D1E6 23                 INC HL
D1E7 5E                 LD E,(HL)
D1E8 2B                 DEC HL
D1E9 2B                 DEC HL
D1EA 0EFF               LD C,0FFH
D1EC CB7A               BIT 7,D
D1EE 2003               JR NZ,TRUNCI2     ;JR IF -VE WITH C=SIGN BYTE FOR SMALL INTEGER
D1F0
D1F0 CBFA               SET 7,D           ;TRUE NUMERIC BIT
D1F2 0C                 INC C             ;SIGN BYTE FOR +VE=0
D1F3
D1F3 2F         TRUNCI2: CPL
D1F4 C691               ADD A,91H
D1F6 90                 SUB B             ;B=8
D1F7 80                 ADD A,B
D1F8 3804               JR C,TRUNCI3
D1FA
D1FA 5A                 LD E,D
D1FB 1600               LD D,0
D1FD 90                 SUB B
D1FE
D1FE 2807       TRUNCI3: JR Z,TRUNCI5
D200
D200 47                 LD B,A
D201
D201 CB3A       TRUNCI4: SRL D
D203 CB1B               RR E
D205 10FA               DJNZ TRUNCI4
D207
D207 C3C8CB     TRUNCI5: JP STOREI
D20A
D20A C660       TRUNCLG: ADD A,60H
D20C F0                 RET P
D20D
D20D 2F                 CPL
D20E 3C                 INC A             ;NEG
D20F 22                 DB 22H            ;"JR+2"
D210
D210            ;NIL-BYTES.
D210
D210 3E28       TRUNC0:  LD A,28H         ;ZERO 40 BITS - ALL OF THEM!
D212
D212 D5         NILBYTES: PUSH DE
D213 EB                 EX DE,HL          ;PT TO AFTER LAST BYTE OF NUMBER
D214 22                 DB 22H            ;"JR+2"
D215
D215 3600       NILBYLP: LD (HL),0
D217
D217 2B         NILBYT2: DEC HL
D218 90                 SUB B             ;B=8
D219 30FA               JR NC,NILBYLP
D21B
D21B 80                 ADD A,B           ;CY SET
D21C 2807               JR Z,NILBYEND
D21E
D21E 47                 LD B,A
D21F 9F                 SBC A,A           ;A=FF
D220
D220 87         NILBMASK: ADD A,A         ;MASK FOR RHS BITS PRODUCED
D221 10FD               DJNZ NILBMASK
D223
D223 A6                 AND (HL)
D224 77                 LD (HL),A         ;DO THE MASKING
D225
D225 EB         NILBYEND: EX DE,HL
D226 D1                 POP DE
D227 C9                 RET
D228
D228
D228            ;*************************************************************************
D228            ;UDG A$. UDG ADDR. E.G. UDG "A". TRANSLATOR CHANGES USR "A" TO UDG "(CHR$ 144)"
```

```
D228                ;UNARY. ENTRY WITH STRING ON FPCS AT (HL)
D228
D228 CDE43E     FPUDG:      CALL SBUFFET
D22B 3D                     DEC A
D22C
D22C C2E83E     IAHOP:      JP NZ,INVARG        ;LEN MUST BE 1
D22F
D22F 1A                     LD A,(DE)           ;READ CHAR FROM BUFFER
D230 FE20                   CP 20H
D232 38F8                   JR C,IAHOP
D234
D234 2A365C                 LD HL,(CHARS)
D237 FE80                   CP 80H
D239 3810                   JR C,FPUDG3         ;JR IF RANGE 20H-7FH
D23B
D23B 2A7B5C                 LD HL,(UDG)
D23E 1180FB                 LD DE,-144*8
D241 19                     ADD HL,DE           ;ALLOW FOR UDG VAR. POINTING TO CHR$ 144
D242 FEA9                   CP 169
D244 3805                   JR C,FPUDG3         ;JR IF CHR$ 80H-A8H (LOW UDGS)
D246
D246 D6A9                   SUB 169
D248 2A7D5C                 LD HL,(HUDG)
D24B
D24B EB         FPUDG3:     EX DE,HL
D24C 6F                     LD L,A
D24D 2600                   LD H,0
D24F 29                     ADD HL,HL
D250 29                     ADD HL,HL
D251 29                     ADD HL,HL
D252 19                     ADD HL,DE
D253 C3D61C                 JP STACKHL          ;STACK CHAR PATTERN ADDR
D256
D256
D256                        INCLUDE SCRSEL1.SAM     ;OPEN/CLOSE SCREEN/PAGES, INTS, BOOTEX, SOUND,
D256                ;SCRSEL1.SAM.  SCREEN, OPEN SCREEN, CLOSE SCREEN, SOUND
D256
D256                ;SELECT SCREEN C
D256
D256 CD82D2     JSCRN:      CALL SCRNTLK2
D259 1806                   JR JSCR2
D25B
D25B                ;SCREEN N - PRINT/PLOT ETC ON SCREEN N, DISPLAY PAGE N IF (CURDISP)=0
D25B
D25B CD583A     SCREEN:     CALL SYNTAX6
D25E
D25E CD7FD2                 CALL SCRNTLK1       ;GET MODE/PAGE FOR SCREEN C, Z IF UNUSED
D261
D261 2823       JSCR2:      JR Z,ISCEH          ;'INVALID SCREEN NUMBER' IF NOT OPEN
D263
D263 3A785A                 LD A,(CUSCRNP)
D266 ED5B9D5C               LD DE,(SCPTR)
D26A 12                     LD (DE),A           ;UPDATE SCLIST MODE AS WE SWITCH OUT THIS SCREEN
D26B 229D5C                 LD (SCPTR),HL
D26E 7E                     LD A,(HL)
D26F F5                     PUSH AF
D270 CDDF3B                 CALL SSVARS         ;SAVE SCREEN VARS TO STORE IN SCREEN PAGE
D273 F1                     POP AF
D274 CDDA3B                 CALL PRSVARS         ;COPY VARS FROM SELECTED SCREEN TO SYS VARS
D277 3A775A                 LD A,(CURDISP)
D27A A7                     AND A
D27B CAAE3B                 JP Z,DEFDISP
D27E
D27E C9                     RET                 ;RET - DISPLAY FIXED ON A GIVEN SCREEN
D27F
D27F
D27F CD331D     SCRNTLK1:   CALL GETBYTE        ;GET SCREEN NUMBER FROM FPCS
D282
D282                ;GET MODE/PAGE FOR SCREEN C FROM LIST. RETURN NZ IF OPEN, Z AND A=FF IF CLOSED
D282                ;EXIT WITH HL POINTING TO ENTRY FOR SCREEN. SCREEN NUMBER 1-16 OR ERROR MSG.
D282
D282 0D         SCRNTLK2:   DEC C
D283 79                     LD A,C
D284 FE10                   CP 10H
D286
D286 D2AC3B     ISCEH:      JP NC,ISCRERR       ;LIMIT SCREENS TO ORIG. OF 1-16
D289
D289 21A05C                 LD HL,SCLIST
D28C 0600                   LD B,0
D28E 09                     ADD HL,BC
D28F 7E                     LD A,(HL)           ;BIT 7=0, BITS 6-5=MODE, BITS 4-0=PAGE (BIT 0=0)
D290 FEFF                   CP 0FFH
D292 C9                     RET                 ;Z IF FF (CLOSED)
D293
D293                ;*********************************************************************************
D293                ;CLOSE SCREEN N
D293                ;CLOSE #N
D293                ;CLOSE N  - CLOSE N PAGES
D293
D293 FE23       CLSCRN:     CP "#"
D295 2022                   JR NZ,CLNCH         ;JR IF NOT A STREAM CLOSE
D297
D297 CD573A                 CALL SSYNTAX6       ;CLOSE #S
D29A
D29A CD4011                 CALL STRMINFO
D29D C8                     RET Z               ;RET IF CLOSED
D29E
D29E E5                     PUSH HL
D29F CD12D4                 CALL CHLTCHK
D2A2 2804                   JR Z,CLOS1          ;OK IF K, S, P, $ OR B **
D2A4                                            ;HL=PTR TO LETTER IN CHANNEL, A=LETTER
D2A4 D1                     POP DE              ;STREAM PTR
D2A5 CF                     RST 08H             ;CLOSE A NON-K/S/P STREAM.
```

```
D2A6 87                     DB CSHK           ;DOS CLOSE.
D2A7 C9                     RET
D2A8
D2A8 110000     CLOS1:      LD DE,0           ;VALUE FOR STRM PTR IF CLOSING 4-15
D2AB 79                     LD A,C
D2AC FE04                   CP 4
D2AE D240D3                 JP NC,OLT4
D2B1
D2B1 5F                     LD E,A
D2B2 219002                 LD HL,STRMTAB+5
D2B5 19                     ADD HL,DE         ;PT HL TO INITIAL VALUE FOR STREAMS 0-3
D2B6 C33FD3                 JP OPEN2
D2B9
D2B9 FEE7       CLNCH:      CP 0E7H           ;SCREENTOK
D2BB 2823                   JR Z,CLSC0
D2BD
D2BD            ;CLOSE N PAGES
D2BD
D2BD CD583A                 CALL SYNTAX6      ;NUMBER OF PAGES TO CLOSE
D2C0
D2C0 CD00D4     CLTO:       CALL OCPSR        ;GET L=PAGES USED NOW, HL PTING TO PAST LAST
D2C3                                          ;ENTRY IN ALLOCT, B/C=PAGES TO CLOSE
D2C3 3AB15C                 LD A,(RAMTOPP)
D2C6 3C                     INC A             ;NO. OF PAGES THAT *MUST* BE USED BY BASIC
D2C7 5F                     LD E,A            ;(UNLESS RAMTOP MOVED)
D2C8 7D                     LD A,L
D2C9 91                     SUB C             ;PAGES THAT WILL BE LEFT AFTER CLOSE
D2CA 3801                   JR C,OMH
D2CC
D2CC BB                     CP E
D2CD
D2CD DA4B3F     OMH:        JP C,OOMERR       ;MUST BE >=MINIMUM NO.
D2D0
D2D0 AF                     XOR A
D2D1
D2D1 2B         CLPL1:      DEC HL
D2D2 77                     LD (HL),A         ;FREE PAGE
D2D3 10FC                   DJNZ CLPL1
D2D5
D2D5 2B         SETLPG:     DEC HL
D2D6 7D                     LD A,L
D2D7 32B05C                 LD (LASTPAGE),A
D2DA C9                     RET
D2DB
D2DB CD82D2     JCLSCR:     CALL SCRNTLK2
D2DE 1806                   JR JCS2
D2E0
D2E0 CD573A     CLSC0:      CALL SSYNTAX6      ;N
D2E3
D2E3 CD7FD2                 CALL SCRNTLK1
D2E6
D2E6 C8         JCS2:       RET Z             ;END IF NOT USED YET
D2E7
D2E7 0C                     INC C
D2E8 0D                     DEC C             ;Z IF ORIG SCREEN NUMBER WAS 1 (C=0)
D2E9 CAAC3B                 JP Z,ISCRERR      ;CANNOT CLOSE SCREEN 1!
D2EC
D2EC E61F                   AND 1FH           ;ISOLATE PAGE USED BY SCREEN TO CLOSE
D2EE 47                     LD B,A
D2EF 3A785A                 LD A,(CUSCRNP)
D2F2 E61F                   AND 1FH
D2F4 B8                     CP B
D2F5 2002                   JR NZ,CNCS
D2F7
D2F7 CF                     RST 08H           ;CANNOT CLOSE CURRENT SCREEN
D2F8 2E                     DB 46             ;'Current screen'
D2F9
D2F9 36FF       CNCS:       LD (HL),0FFH      ;MARK SCREEN ENTRY AS CLOSED
D2FB 68                     LD L,B            ;HL WILL PT TO ALLOC TABLE ENTRY FOR SCRN PAGE
D2FC 2651                   LD H,ALLOCT/256
D2FE AF                     XOR A
D2FF 77                     LD (HL),A
D300 23                     INC HL
D301 77                     LD (HL),A
D302 C9                     RET
D303
D303            ;*************************************************************************
D303            ;OPEN SCREEN N,M<,C> (NUMBER, MODE)
D303            ;OPEN #N,A$  (OPEN A STREAM TO A CHANNEL)
D303            ;OPEN N      (OPEN A NUMBER OF PAGES FOR USE)
D303
D303 FE23       OPSCRN:     CP "#"
D305 C250D3                 JP NZ,OPNCH       ;JR IF NOT 'OPEN TO A CHANNEL'
D308
D308            ;CHECK FOR OPEN #S;A$ OR OPEN #S,A$
D308
D308 CDE33A                 CALL SEXPT1NUM
D30B CD813A                 CALL INSISCSC     ;',/;'
D30E CD663A                 CALL SYNTAXA
D311
D311 CD8A18                 CALL SWOP12       ;GIVES NAME,STREAM
D314 CD4011                 CALL STRMINFO     ;Z IF CLOSED. HL PTS TO STRM PTR MSB
D317 E5                     PUSH HL
D318 2809                   JR Z,OPEN1        ;JR IF CHANNEL IS CLOSED
D31A
D31A CD12D4                 CALL CHLTCHK
D31D 2002                   JR NZ,SAOERR      ;ERROR IF NOT OPEN TO K/S/P/$/B/N ALREADY
D31F
D31F 3002                   JR NC,OPEN1       ;JR IF OPEN TO K/S/P, ERROR IF OPEN TO $/B/N
D321
D321 CF         SAOERR:     RST 08H
D322 2D                     DB 45             ;'Stream is already open'
D323
```

```
D323 CDEA3E     OPEN1:      CALL SBFSR          ;COPY FILE NAME TO BUFFER. DE=START, A/C=LEN
D326 CA5CE0                 JP Z,IFNER          ;JP IF LEN ZERO
D329
D329 3D                     DEC A
D32A 200F                   JR NZ,INVCHP        ;ERROR (OR DOS) IF NAME LONGER THAN 1. E.G.
D32C                                            ;OPEN #4;"S" IS OK BUT OPEN #4;"FILE" JUMPS
D32C 1A                     LD A,(DE)
D32D F620                   OR 20H              ;LETTERS BECOME L.C. AND '$' UNCHANGED
D32F 2146D3                 LD HL,CLTAB
D332 0605                   LD B,5              ;CHECK 5 CHANNEL TYPES
D334
D334 BE         OPCL:       CP (HL)
D335 23                     INC HL
D336 2807                   JR Z,OPEN2
D338
D338 23                     INC HL
D339 10F9                   DJNZ OPCL
D33B
D33B E1         INVCHP:     POP HL              ;PTR TO STREAM PTR MSB
D33C                                            ;DE POINTS TO NAME, C=LEN
D33C CF                     RST 08H             ;OPEN #S,A$ WITH A$ NOT K, S, P, $, OR B OR LEN>1
D33D 86                     DB OSHK             ;DOS OPEN
D33E C9                     RET
D33F
D33F 5E         OPEN2:      LD E,(HL)           ;GET DISPLACEMENT FROM TABLE
D340
D340 E1         OLT4:       POP HL
D341 3600                   LD (HL),0
D343 2B                     DEC HL
D344 73                     LD (HL),E
D345 C9                     RET
D346
D346 6B01       CLTAB:      DB "k",1
D348 7306                   DB "s",6
D34A 7010                   DB "p",16
D34C 2415                   DB "$",21
D34E 621A                   DB "b",26
D350
D350 FEE7       OPNCH:      CP 0E7H             ; SCREENTOK
D352 283C                   JR Z,OPSCR0
D354
D354            ;OPEN N PAGES  OR OPEN TO PAGE N
D354
D354 FE8E                   CP TOTOK
D356 201A                   JR NZ,NOTP
D358
D358 CD573A                 CALL SSYNTAX6
D35B
D35B CD331D                 CALL GETBYTE
D35E 0D                     DEC C               ;E.G. OPEN TO 1 DOES NOTHING IF LASTPAGE=0
D35F 3AB05C                 LD A,(LASTPAGE)
D362 91                     SUB C
D363 C8                     RET Z               ;RET IF OPEN TO CORRECT PAGE ALREADY
D364
D364 3806                   JR C,TOPO
D366
D366 CDDA1C                 CALL STACKA
D369 C3C0D2                 JP CLTO
D36C
D36C ED44       TOPO:       NEG                 ;GET PAGES TO OPEN
D36E CDDA1C                 CALL STACKA
D371 BF                     CP A
D372
D372            ;OPEN N PAGES - NZ ON ENTRY
D372
D372 C4583A     NOTP:       CALL NZ,SYNTAX6     ;NUMBER OF PAGES TO OPEN
D375
D375 CD00D4                 CALL OCPSR          ;GET HL PTING TO PAST LAST CURRENT PAGE
D378                                            ;ENTRY IN ALLOCT, B/C=PAGES TO CLOSE
D378
D378 34         OPL1:       INC (HL)
D379 35                     DEC (HL)
D37A C24B3F                 JP NZ,OOMERR        ;ERROR IF NOT ENOUGH FREE PAGES ABOVE CONTEXT'S
D37D                                            ;CURRENT PAGES
D37D 23                     INC HL
D37E 10F8                   DJNZ OPL1
D380
D380 41                     LD B,C
D381 CDD5D2                 CALL SETLPG         ;DEC HL, SET LAST PAGE
D384
D384 3640       OPL4:       LD (HL),40H         ;RESERVE PAGE
D386 2B                     DEC HL
D387 10FB                   DJNZ OPL4
D389
D389 C9                     RET
D38A
D38A C5         JOPSCR:     PUSH BC             ;MODE IN B
D38B CD82D2                 CALL SCRNTLK2       ;CHECK SCREEN C
D38E 180D                   JR JOPS2
D390
D390            ;OPEN SCREEN N,M    (NUMBER, MODE)
D390
D390 CD5E3A     OPSCR0:     CALL SSYNTAX8       ;N,M
D393
D393 112204                 LD DE,0400H+34
D396 CD5F1F                 CALL LIMDB          ;MODE 0-3 FROM ORIG OF 1-4
D399 F5                     PUSH AF             ;MODE
D39A CD7FD2                 CALL SCRNTLK1       ;GET N FROM FPCS, LOOK IN TABLE FOR SCREEN N
D39D
D39D 2802       JOPS2:      JR Z,OPSCR4         ;OK IF NOT USED YET
D39F
D39F CF                     RST 08H
D3A0 2C                     DB 44               ;'Screen already open'
```

```
D3A1
D3A1 E5          OPSCR4:    PUSH HL            ;SCREEN LIST PTR
D3A2 212051                 LD HL,ALLOCT+20H   ;HL PTS TO ALLOCT TERMINATOR
D3A5
D3A5 2D          OPSCRLP:   DEC L
D3A6 7E                     LD A,(HL)
D3A7 2D                     DEC L
D3A8 CA4B3F                 JP Z,OOMERR        ;OUT OF MEMORY IF NO SPACE FOR NEW SCREEN
D3AB
D3AB B6                     OR (HL)            ;Z IF 2 PAGES UNUSED (0)
D3AC 20F7                   JR NZ,OPSCRLP      ;LOOP UNTIL FOUND A FREE 32K BLOCK, EVEN START PAGE
D3AE
D3AE D1                     POP DE             ;SCLIST PTR
D3AF F1                     POP AF             ;MODE TO OPEN IN
D3B0 F5                     PUSH AF
D3B1
D3B1             ;A=MODE, HL PTS TO ALLOCT ENTRY, DE TO SCLIST ENTRY
D3B1             ;MARK ALLOCT, SCLIST
D3B1
D3B1 36C0                   LD (HL),0C0H
D3B3 23                     INC HL
D3B4 36C0                   LD (HL),0C0H       ;RESERVE PAGES IN SYSTEM PAGE ALLOC TABLE
D3B6 2B                     DEC HL             ;L=PAGE NUMBER 02-1EH
D3B7 0F                     RRCA
D3B8 0F                     RRCA
D3B9 CB3F                   SRL A              ;0MMX XXXX
D3BB AD                     XOR L
D3BC E6E0                   AND 0E0H           ;0MM FROM A, PAGE FROM L
D3BE AD                     XOR L
D3BF 12                     LD (DE),A          ;MODE/PAGE DATA TO SCRN LIST
D3C0 F5                     PUSH AF            ;MODE/PAGE FOR NEW SCREEN
D3C1 CDDF3B                 CALL SSVARS        ;STORE CURRENT SCREEN VARS SINCE WE ARE FIDDLING...
D3C4 CDEBD3                 CALL SDISRC        ;STORE CURRENT PALTAB
D3C7 21785A                 LD HL,CUSCRNP
D3CA 46                     LD B,(HL)
D3CB F1                     POP AF             ;NEW MODE/PAGE
D3CC 77                     LD (HL),A
D3CD D1                     POP DE             ;D=MODE FOR NEW SCREEN
D3CE C5                     PUSH BC            ;B=NORMAL CUSCRNP
D3CF D5                     PUSH DE            ;MODE
D3D0 21445A                 LD HL,THFATP
D3D3 46                     LD B,(HL)
D3D4 3601                   LD (HL),1          ;ENSURE NO XRG CHANGE
D3D6 F1                     POP AF             ;MODE
D3D7 C5                     PUSH BC
D3D8 CDEBD6                 CALL MODPT2        ;CLEAR NEW SCREEN IN DESIRED MODE, SET UP EXPAN.
D3DB                                           ;TABLES, ETC
D3DB
D3DB CDEBD3                 CALL SDISRC        ;COPY PALTAB TO NEW SCREEN AREA
D3DE F1                     POP AF
D3DF 32445A                 LD (THFATP),A      ;ORIG STATUS
D3E2 CDDF3B                 CALL SSVARS        ;SET UP VARS IN NEW SCREEN PAGE
D3E5 F1                     POP AF             ;ORIG CUSCRNP
D3E6 CDDA3B                 CALL PRSVARS       ;RESTORE SCREEN VARS
D3E9 37                     SCF                ;"RESTORE PALTAB"
D3EA 3E                     DB 3EH             ;"JR+1"
D3EB
D3EB A7          SDISRC:    AND A              ;NC - "SAVE PALTAB"
D3EC
D3EC 3A785A                 LD A,(CUSCRNP)
D3EF C3BE3B                 JP SDISR
D3F2
D3F2
D3F2             ;SEE IF PAGE L IS IN USE AS A SCREEN PAGE
D3F2
D3F2 0610        SPGLOOK:   LD B,10H
D3F4 11A05C                 LD DE,SCLIST
D3F7
D3F7 1A          SPGLKLP:   LD A,(DE)
D3F8 13                     INC DE
D3F9 E61F                   AND 1FH
D3FB BD                     CP L
D3FC C8                     RET Z              ;RET IF THIS CONTEXT USES PG L AS A SCREEN ALREADY
D3FD
D3FD 10F8                   DJNZ SPGLKLP
D3FF C9                     RET                ;NZ IF UNUSED
D400
D400             ;OPEN/CLOSE PAGE SR
D400
D400 111E1E      OCPSR:     LD DE,1E00H+30
D403 CD5F1F                 CALL LIMDB         ;ALLOW ONLY 1-30
D406 41                     LD B,C             ;B AND C=PAGES TO OPEN
D407 210051                 LD HL,ALLOCT
D40A
D40A 23          OCL1:      INC HL
D40B 7E                     LD A,(HL)
D40C FE40                   CP 40H
D40E 28FA                   JR Z,OCL1          ;LOOK PAST LAST PAGE RESERVED BY BASIC
D410
D410 55                     LD D,L              ;NUMBER OF PAGES CURRENTLY USED
D411 C9                     RET
D412
D412
D412             ;CHANNEL LETTER CHECK.
D412             ;ENTRY: DE=DISP FROM CHANS TO CHANNEL.
D412             ;EXIT: HL POINTS TO CHANNEL LETTER, Z,NC IF K/S/P. Z,CY IF $/B
D412
D412 2A4F5C      CHLTCHK:   LD HL,(CHANS)
D415 19                     ADD HL,DE          ;PT TO 2ND BYTE OF CHANNEL
D416 23                     INC HL
D417 23                     INC HL
D418 23                     INC HL
D419 7E                     LD A,(HL)
```

```
D41A FE4B                   CP "K"
D41C C8                     RET Z
D41D
D41D FE53                   CP "S"
D41F C8                     RET Z
D420
D420 FE50                   CP "P"
D422 C8                     RET Z
D423
D423 FE24                   CP "$"
D425 37                     SCF
D426 C8                     RET Z
D427
D427 FE42                   CP "B"
D429 37                     SCF
D42A C8                     RET Z
D42B
D42B A7                     AND A
D42C C9                     RET
D42D
D42D
D42D              ;INTERRUPTS. ENTRY VIA RST 38H
D42D
D42D 79           INTS:     LD A,C
D42E 32D15A                 LD (LASTSTAT),A
D431 C5                     PUSH BC             ;HMPR/STAT
D432 D5                     PUSH DE
D433 1F                     RRA
D434 D2E2D4                 JP NC,LINEINT
D437
D437 1F                     RRA
D438 300B                   JR NC,COMINT
D43A
D43A 1F                     RRA
D43B 300D                   JR NC,MIPINT
D43D
D43D 1F                     RRA
D43E 3017                   JR NC,FRAMINT
D440
D440              ;MIDI OUTPUT (OR OTHER) INTERRUPT
D440
D440 2AEA5A                 LD HL,(MOPV)
D443 180A                   JR CMMDIC
D445
D445              ;COMS (EX MOUSE) INTERRUPT
D445
D445 2AE65A       COMINT:   LD HL,(COMSV)
D448 1805                   JR CMMDIC
D44A
D44A              ;MIDI INPUT INTERRUPT
D44A
D44A 2AE85A       MIPINT:   LD HL,(MIPV)
D44D DBFD                   IN A,(MDIPORT)    ;READ MIDI INPUT BYTE (AUTOMATIC INT CANCEL?)
D44F
D44F 24           CMMDIC:   INC H
D450 25                     DEC H
D451 C40500                 CALL NZ,HLJUMP
D454
D454 C30AD5                 JP INTEND
D457
D457 2AE25A       FRAMINT:  LD HL,(FRAMIV)
D45A 7C                     LD A,H
D45B B5                     OR L
D45C C40500                 CALL NZ,HLJUMP
D45F
D45F 210056                 LD HL,LINICOLS
D462 22095A                 LD (LINIPTR),HL           ;RESET LINE INT COL CHANGE LIST PTR
D465                                                  ;TO START
D465 7E                     LD A,(HL)                 ;LINE TO INT ON,
D466 D3F9                   OUT (STATPORT),A          ;OR FF FOR 'NEVER INTERRUPT'
D468
D468 3AC75A                 LD A,(PALFLAG)
D46B 21E755                 LD HL,PALTAB+15
D46E 1F                     RRA
D46F 3002                   JR NC,FRMI3
D471
D471 2EFB                   LD L,>(PALTAB+35)
D473
D473 01F810       FRMI3:    LD BC,16*256+CLUTPORT
D476 EDBB                   OTDR                      ;SET UP PALETTE MEMORIES
D478
D478 21C65A                 LD HL,SPEEDIC             ;COUNTER FOR DELAY BETWEEN SWAPPING INKS
D47B 35                     DEC (HL)
D47C 201A                   JR NZ,FRMI5
D47E
D47E 23                     INC HL                    ;PT TO PALETTE FLAG
D47F 34                     INC (HL)                  ;FLIP BIT 0 OF PALFLAG - USE OTHER PALET
D480 3A085A                 LD A,(SPEEDINK)
D483 2B                     DEC HL
D484 77                     LD (HL),A                 ;RELOAD COUNTER TILL NEXT FLASH
D485 210056                 LD HL,LINICOLS
D488 180A                   JR FRMI4
D48A
D48A 23           FISCL:    INC HL
D48B 23                     INC HL
D48C 7E                     LD A,(HL)         ;COL 1
D48D 23                     INC HL
D48E 46                     LD B,(HL)         ;COL 2
D48F 77                     LD (HL),A
D490 2B                     DEC HL
D491 70                     LD (HL),B
D492 23                     INC HL
D493 23                     INC HL            ;NEXT LINE
```

```
D494
D494 7E          FRMI4:     LD A,(HL)
D495 3C                     INC A
D496 20F2                   JR NZ,FISCL
D498
D498 21785C      FRMI5:     LD HL,FRAMES
D49B 34                     INC (HL)
D49C 2025                   JR NZ,INTS3
D49E
D49E 3AC45A                 LD A,(SOFFCT)     ;DECED EVERY 5.1 SECS OR SO, SET TO 0 BY KEYBD USE.
D4A1 3D                     DEC A             ;IF DECED TO ZERO, KYBD NOT USED FOR ABOUT 22 MINS.
D4A2 32C45A                 LD (SOFFCT),A
D4A5 200D                   JR NZ,INTS2       ;JR IF USED WITHIN LAST 22 MINS.
D4A7
D4A7 3A325A                 LD A,(SOFE)
D4AA A7                     AND A
D4AB 2007                   JR NZ,INTS2       ;JR IF 'SCREEN OFF' DISABLED
D4AD
D4AD 3E80                   LD A,80H
D4AF D3FE                   OUT (KEYPORT),A
D4B1 32C55A                 LD (SOFLG),A      ;'SCREEN OFF'
D4B4
D4B4 23          INTS2:     INC HL
D4B5 34                     INC (HL)          ;INC SECOND BYTE OF FRAMES
D4B6 200B                   JR NZ,INTS3
D4B8
D4B8 23                     INC HL
D4B9 34                     INC (HL)          ;AND THIRD
D4BA 2007                   JR NZ,INTS3
D4BC
D4BC 2A7F5C                 LD HL,(FRAMES34)
D4BF 23                     INC HL            ;FOURTH AND FIFTH
D4C0 227F5C                 LD (FRAMES34),HL
D4C3
D4C3 2AFC5A      INTS3:     LD HL,(MOUSV)
D4C6 25                     DEC H
D4C7 7C                     LD A,H
D4C8 24                     INC H
D4C9 200F                   JR NZ,INTS4       ;IF JR NOT TAKEN, A=FF
D4CB
D4CB DBFE                   IN A,(KEYPORT)
D4CD 218E5B                 LD HL,MSEDP
D4D0 0608                   LD B,8            ;READ MOUSE 9 TIMES TO CANCEL IT
D4D2
D4D2 3EFF        MSDML:     LD A,0FFH
D4D4 DBFE                   IN A,(KEYPORT)
D4D6 77                     LD (HL),A
D4D7 23                     INC HL
D4D8 10F8                   DJNZ MSDML        ;ALWAYS Z HERE
D4DA
D4DA C40500      INTS4:     CALL NZ,HLJUMP
D4DD
D4DD CD10D5      INTS5:     CALL KEYRD2
D4E0 1828                   JR INTEND
D4E2
D4E2
D4E2             ;ENTRY:
D4E2             ;LINIPTR POINTS TO SCAN LINE VALUE IN 4-BYTE CURRENT ENTRY (SCAN/PAL/INK1/INK2)
D4E2             ;BORDER TIME HAS FINISHED BEFORE WE GET HERE
D4E2
D4E2 2A095A      LINEINT:   LD HL,(LINIPTR)
D4E5
D4E5 56          LNINSCAN:  LD D,(HL)                 ;D=SCAN LINE IN LINICOL ENTRY
D4E6 01F801                 LD BC,0100H+CLUTPORT      ;PORT BC READS PEN Y (SCAN LINE)
D4E9 DBFE                   IN A,(KEYPORT)
D4EB E620                   AND 20H
D4ED 2005                   JR NZ,LINILP              ;JR IF LIGHT PEN KEEPS PEN Y FROZEN
D4EF
D4EF ED78        LNWAITLP:  IN A,(C)
D4F1 BA                     CP D
D4F2 28FB                   JR Z,LNWAITLP             ;WAIT FOR NEXT SCAN'S BORDER
D4F4
D4F4 23          LINILP:    INC HL
D4F5 46                     LD B,(HL)                 ;PALETTE MEMORY NO.
D4F6 23                     INC HL
D4F7 7E                     LD A,(HL)                 ;NEW VALUE
D4F8 ED79                   OUT (C),A                 ;OUT (BC) WRITES DESIRED PAL. MEM.
D4FA 23                     INC HL                    ;SKIP ALTERNATE VALUE FOR FLASHING
D4FB 23                     INC HL
D4FC 7E                     LD A,(HL)                 ;NEXT SCAN VALUE (MIGHT BE THE SAME)
D4FD 92                     SUB D
D4FE 28F4                   JR Z,LINILP               ;DO MORE CHANGES FOR THIS SCAN (SCAN D+1)
D500                                                  ;TAKES ABOUT 32 T'S FOR FIRST COLOUR,
D500                                                  ;THEN ABOUT 80 T'S PER LOOP, SO SHOULD
D500                                                  ;GET 2 CHANGES IN BORDER TIME, AND ABOUT
D500                                                  ;3 IN MIDDLE PART.
D500                                                  ;(IF WE USE TOO MANY CHANGES, WE MIGHT
D500                                                  ;MISS A CLOSE-FOLLOWING LINE INT CHANGE
D500                                                  ;AND THEN MISS ALL LATER CHANGES COS
D500                                                  ;LINE INT REG NOT SET UP.)
D500
D500 FE01                   CP 1                      ;IS NEXT ENTRY FOR NEXT SCAN?
D502 28E1                   JR Z,LNINSCAN             ;IF SO, WAIT FOR IT
D504
D504 82                     ADD A,D
D505 22095A                 LD (LINIPTR),HL
D508 D3F9                   OUT (STATPORT),A          ;IF A>191, NO MORE COLOUR CHANGES
D50A
D50A 2AD25A      INTEND:    LD HL,(SPSTORE)
D50D D1                     POP DE
D50E F1                     POP AF          ;A=FORMER HMPR
D50F C9                     RET
D510
```

```
D510
D510                ;FROM INTS
D510
D510 CD33D5     KEYRD2:     CALL KINTER         ;SCAN KEYBD, PLACE CHAR IN BUFFER IF THERE IS ONE.
D513 2A955C                 LD HL,(KBQP)        ;L=KEY BUFFER QUEUE END (POSN TO PLACE CHAR AT)
D516                                            ;H=QUEUE HEAD (PTS TO NEXT CHAR TO BE READ)
D516                                            ;BOTH ARE DISPLACEMENTS FROM KBQB (BUFFER START)
D516                                            ;IF H=L, BUFFER IS EMPTY.
D516 7C                     LD A,H
D517 BD                     CP L
D518 C8                     RET Z               ;RET IF NO CHARS IN BUFFER
D519
D519 213B5C                 LD HL,FLAGS
D51C CB6E                   BIT 5,(HL)
D51E C0                     RET NZ              ;RET IF LAST KEY NOT READ YET (KEY AVAILABLE)
D51F
D51F CBEE                   SET 5,(HL)          ;'KEY PRESSED'
D521 6F                     LD L,A
D522 2600                   LD H,0
D524 3C                     INC A
D525 E607                   AND 07H
D527 32965C                 LD (KBQP+1),A       ;NEW HEAD POSN REFLECTS CHAR TRANSFER TO COME
D52A 118D5C                 LD DE,KBQB
D52D 19                     ADD HL,DE           ;PT TO HEAD
D52E 7E                     LD A,(HL)
D52F 32085C                 LD (LASTK),A        ;TRANSFER TO LASTK
D532 C9                     RET
D533
D533 CDBCD5     KINTER:     CALL KEYSCAN
D536 3E00                   LD A,0
D538 2055                   JR NZ,LDLH          ;JR IF NO KEY PRESSED
D53A                                            ;(NLASTH=0 IF NO KEY OR JUST A SHIFT KEY. USED BY
D53A                                            ;AUTO-REPEAT)
D53A 21565C                 LD HL,NLASTH
D53D 7B                     LD A,E              ;UNSHIFTED KEY CODE
D53E BE                     CP (HL)
D53F 280A                   JR Z,KBCR           ;JR IF SAME KEY AS LAST TIME
D541
D541 23                     INC HL
D542 BE                     CP (HL)
D543 2806                   JR Z,KBCR           ;OR TIME BEFORE
D545
D545 FE41                   CP 65               ;CODE FOR ENTER KEY - PRONE TO STUTTER
D547 2002                   JR NZ,KBCR
D549
D549 23                     INC HL              ;EXTRA CHECK FOR ENTER KEY ONLY
D54A BE                     CP (HL)             ;TIME-BEFORE-TIME BEFORE
D54B
D54B F5         KBCR:       PUSH AF
D54C CD9AD5                 CALL KYVL           ;GET A=CHAR, USING D AND E
D54F 4F                     LD C,A
D550 21055C                 LD HL,REPCT
D553 F1                     POP AF
D554 201F                   JR NZ,KBDI          ;JR IF NOT SAME KEY AS LAST TIME, OR TIME BEFORE
D556
D556 35                     DEC (HL)
D557 C0                     RET NZ              ;RET IF NOT TIME TO REPEAT KEY
D558
D558 E5                     PUSH HL             ;ELSE CHECK IT IS STILL HELD DOWN (NOT SOME OTHER
D559 11F65B                 LD DE,KBUFF+8       ;KEY)
D55C 2A035C                 LD HL,(LKPB)
D55F 7C                     LD A,H              ;A=1 FOR PORT FE, 9 FOR FF
D560 83                     ADD A,E
D561 5F                     LD E,A
D562 1A                     LD A,(DE)           ;A=BYTE FOR DESIRED PORT FROM KBUFF BIT MAP
D563                                            ;'CURRENT' DATA
D563                                            ;L=1 FOR BIT 7, 8 FOR BIT 0
D563 07         KBXL:       RLCA
D564 2D                     DEC L
D565 20FC                   JR NZ,KBXL          ;GET DESIRED KEY BIT TO CARRY
D567
D567 E1                     POP HL
D568 D8                     RET C               ;NO AUTO-REPEAT IF KEY NO LONGER HELD DOWN
D569
D569 34                     INC (HL)
D56A 3A3B5C                 LD A,(FLAGS)
D56D E620                   AND 20H
D56F C0                     RET NZ              ;RET IF STILL KEYS IN BUFFER (DO NOT ACCUMULATE
D570                                            ;AUTO-REPEATING KEYS)
D570
D570 3A0A5C                 LD A,(REPPER)       ;RELOAD REPCT FROM REPPER (DELAY BETWEEN REPEATS)
D573 1803                   JR KBRK
D575
D575 3A095C     KBDI:       LD A,(REPDEL)
D578
D578 77         KBRK:       LD (HL),A           ;NEW KEY - SET UP REPCT FOR INITIAL DELAY BEFORE
D579                                            ;AUTO-REPEAT
D579
D579 2A955C                 LD HL,(KBQP)        ;L=KEY BUFFER QUEUE END (POSN TO PLACE CHAR AT)
D57C                                            ;H=QUEUE HEAD (PTS TO NEXT CHAR TO BE READ)
D57C                                            ;BOTH ARE DISPLACEMENTS FROM KBQB (BUFFER START)
D57C                                            ;IF H=L, BUFFER IS EMPTY. IF *NEW* POSN OF H=L,
D57C                                            ;BUFFER IS FULL
D57C 7D                     LD A,L
D57D 3C                     INC A
D57E E607                   AND 07H
D580 BC                     CP H
D581 C8                     RET Z               ;RET IF BUFFER FULL (WRAPPED SO END=HEAD)
D582
D582 32955C                 LD (KBQP),A
D585 2600                   LD H,0
D587 118D5C                 LD DE,KBQB
D58A 19                     ADD HL,DE           ;PT TO END
```

```
D58B 71                 LD (HL),C       ;PLACE CHAR IN BUFFER
D58C 3A065C             LD A,(LASTKV)
D58F
D58F 21565C     LDLH:   LD HL,NLASTH
D592 4E                 LD C,(HL)
D593 77                 LD (HL),A
D594 23                 INC HL
D595 46                 LD B,(HL)
D596 71                 LD (HL),C
D597 23                 INC HL
D598 70                 LD (HL),B
D599 C9                 RET
D59A
D59A            ;GET KEY CODE FROM MAP USING E AND D (SHIFT)
D59A
D59A 2AD85B     KYVL:   LD HL,(KBTAB)
D59D 7A                 LD A,D          ;00 IF NO SHIFT, OR 1/2/3 FOR CAPS/SYM/CNTRL
D59E 1600               LD D,0
D5A0 19                 ADD HL,DE
D5A1 A7                 AND A
D5A2 2806               JR Z,KINT4      ;JR IF NO SHIFT USED
D5A4
D5A4 1E46               LD E,70
D5A6
D5A6 19         KYVLP:  ADD HL,DE       ;PT TO CAPS SHIFT/SYM/CNTRL TABLE
D5A7 3D                 DEC A
D5A8 20FC               JR NZ,KYVLP
D5AA
D5AA 3A6A5C     KINT4:  LD A,(FLAGS2)
D5AD E608               AND 08H         ;Z=CAPS LOCK OFF
D5AF 7E                 LD A,(HL)
D5B0 C8                 RET Z
D5B1
D5B1 CD1E3B             CALL ALPHA
D5B4 D0                 RET NC          ;RET IF NOT A LETTER
D5B5
D5B5 E6DF               AND 0DFH        ;FORCE LOWER CASE LETTERS TO UPPER CASE
D5B7 C9                 RET
D5B8
D5B8            ;KEYSCAN.
D5B8            ;ACTION: LOOK FOR CHANGED BITS IN KEYBOARD STATE SINCE LAST SCAN, SO THAT
D5B8            ;SEVERAL KEYS CAN BE PRESSED DOWN AND A NEW KEY WILL STILL BE NOTICED. IGNORE
D5B8            ;KEY RELEASES. IF THERE HAVE BEEN NO NEW PRESSES, RETURN THE SAME KEYSCAN
D5B8            ;CODES AS LAST TIME, UNLESS ALL KEYS ARE RELEASED - THEN RETURN NULL CODE.
D5B8
D5B8            ;ENTRY: NO CONDITIONS
D5B8            ;EXIT: IF NZ, NO KEY PRESSED AND DE=FFFF
D5B8            ;       IF Z, E=KEY VALUE
D5B8            ;             IF D=0, NO SHIFT, ELSE D=1 (CAPS SH.) 2 (SYM. SH.) OR 3 (CONTROL)
D5B8
D5B8            ;ENTER AT KEYSCAN+3 WITH HL=DIFFERENT 18-BYTE KBUFF IF DESIRED
D5B8
D5B8            ;USED BY INKEY$ - 2 SCANS IN CASE OF INTERRUPT
D5B8
D5B8 CDBCD5     TWOKSC:  CALL KEYSCAN
D5BB C8                  RET Z
D5BC
D5BC 21EE5B     KEYSCAN: LD HL,KBUFF    ;18-BYTE STORE
D5BF E5                  PUSH HL
D5C0 01FEFE              LD BC,0FE00H+KEYPORT
D5C3                                    ;C HAS PORT FOR BITS 4-0, B STARTS
D5C3                                    ;WITH A8 LOW FOR 1ST KEY ROW
D5C3 16E0                LD D,0E0H      ;MASK TO KEEP BITS 7-5 OF A, USE 4-0 OF E
D5C5
D5C5 ED58       KBSL:    IN E,(C)       ;READ BITS 4-0
D5C7 78                  LD A,B         ;A WILL BE ON HI ADDR LINES DURING PORT READ
D5C8 DBF9                IN A,(STATPORT) ;READ BITS 7-5
D5CA AB                  XOR E
D5CB A2                  AND D
D5CC AB                  XOR E
D5CD 04                  INC B
D5CE 280B                JR Z,KBEL      ;JR IF WE JUST DID 'SPECIAL' PORT FFFEH
D5D0
D5D0 77                  LD (HL),A
D5D1 05                  DEC B
D5D2 23                  INC HL
D5D3 CB00                RLC B          ;NEXT PORT (FEFE, FDFE, FBFE...7FFE)
D5D5 38EE                JR C,KBSL      ;LOOP UNTIL BACK TO PORT FEFE AGAIN
D5D7
D5D7 06FF                LD B,0FFH
D5D9 18EA                JR KBSL
D5DB
D5DB F6E1       KBEL:    OR 0E1H        ;ONLY BITS 4-0 VALID FOR PORT FFFE, AND FORCE
D5DD 77                  LD (HL),A      ;BIT 0 (CONTROL KEY BIT) HI TOO
D5DE E1                  POP HL         ;KBUFF
D5DF 54                  LD D,H
D5E0 5D                  LD E,L
D5E1 CBC6                SET 0,(HL)     ;'NO CAPS SHIFT' (SCAN SEPARATELY FOR IT)
D5E3 23                  INC HL
D5E4 23                  INC HL
D5E5 23                  INC HL
D5E6 CBEE                SET 5,(HL)     ;'NO ESC'
D5E8 23                  INC HL
D5E9 23                  INC HL
D5EA 23                  INC HL
D5EB 23                  INC HL
D5EC CBCE                SET 1,(HL)     ;'NO SYM. SHIFT' (SCAN SEPARATELY FOR IT)
D5EE 23                  INC HL
D5EF 23                  INC HL         ;PT TO LAST SCAN DATA (BYTES 10-18)
D5F0 0609                LD B,9         ;(HL=KBUFF+9, DE=KBUFF)
D5F2
D5F2 1A         KBCL:    LD A,(DE)      ;THIS SCAN DATA
D5F3 4E                  LD C,(HL)      ;LAST SCAN DATA
```

```
D5F4 77                  LD (HL),A          ;LAST SCAN DATA UPDATED WITH NEW DATA - BUT WE
D5F5                                        ;HAVE IT IN C NOW
D5F5 A9                  XOR C
D5F6 2F                  CPL                ;GET ALTERED BITS SINCE LAST SCAN AS 0'S
D5F7 B6                  OR (HL)            ;KEEP AS 0'S IF CHANGE WAS 1->0 (PRESSED)
D5F8 12                  LD (DE),A          ;'CHANGED' DATA TO KBUFF
D5F9 23                  INC HL
D5FA 13                  INC DE
D5FB 10F5                DJNZ KBCL
D5FD
D5FD 0609                LD B,9
D5FF
D5FF 1B          KBDL:   DEC DE
D600 1A                  LD A,(DE)
D601 3C                  INC A
D602 2011                JR NZ,KBYK         ;JR IF ANY BIT RESET
D604
D604 10F9                DJNZ KBDL
D606
D606             ;NO NEW PRESSES SINCE LAST SCAN. HL=KBUFF+18
D606
D606 0609                LD B,9
D608 ED5B065C            LD DE,(LASTKV)     ;E=LAST VALUE
D60C
D60C 2B          KBAKL:  DEC HL
D60D 7E                  LD A,(HL)
D60E 3C                  INC A
D60F 202B                JR NZ,KBSH         ;JR IF KBUFF SHOWS ANY KEY PRESSED APART FROM SHIFT
D611                                        ;USE LAST VALUE, PLUS CURRENT SHIFT STATUS
D611 10F9                DJNZ KBAKL
D613
D613 04                  INC B              ;NZ
D614 C9                  RET
D615
D615 3D          KBYK:   DEC A
D616 0E09                LD C,9
D618
D618 0D          KBBL:   DEC C
D619 1F                  RRA
D61A 38FC                JR C,KBBL          ;CHANGE BIT POSN TO NUMBER IN C (1-8)
D61C
D61C ED43035C            LD (LKPB),BC       ;LAST KEY PORT/BIT SAVED FOR AUTO-REPEAT CHECKING
D620                                        ;B=1 IF LAST PORT = FEFE, 9 IF FFFE. C=8 FOR BIT
D620                                        ;0, 1 FOR BIT 7
D620 79                  LD A,C
D621 87                  ADD A,A            ;*2
D622 87                  ADD A,A            ;*4
D623 87                  ADD A,A            ;*8
D624 81                  ADD A,C            ;*9 (09-48H)
D625 90                  SUB B              ;SUB 1-9 TO GET 00-47H
D626 5F                  LD E,A             ;SCAN CODE TO E
D627 21C45A              LD HL,SOFFCT
D62A AF                  XOR A
D62B 77                  LD (HL),A          ;ZERO SCREEN OFF COUNTER - KEYBOARD USED
D62C 23                  INC HL
D62D 86                  ADD A,(HL)         ;SOFLG
D62E 280C                JR Z,KBSH          ;JR IF SCREEN NOT TURNED OFF BY NO KEY USE
D630
D630 AF                  XOR A
D631 77                  LD (HL),A          ;'ON'
D632 3A4B5C              LD A,(BORDCOL)
D635 E67F                AND 7FH
D637 324B5C              LD (BORDCOL),A
D63A D3FE                OUT (KEYPORT),A
D63C
D63C 01FEFE      KBSH:   LD BC,0FEFEH
D63F ED78                IN A,(C)
D641 1601                LD D,1
D643 A2                  AND D
D644 2812                JR Z,KBLD          ;JR IF CAPS SHIFT - D=1
D646                                        ;ELSE D=1
D646 04                  INC B
D647 ED78                IN A,(C)
D649 A2                  AND D
D64A 1603                LD D,3
D64C 280A                JR Z,KBLD          ;JR IF CONTROL - D=3
D64E
D64E 15                  DEC D
D64F 067F                LD B,7FH
D651 ED78                IN A,(C)
D653 A2                  AND D
D654 2802                JR Z,KBLD          ;JR IF SYM SHIFT - D=2
D656
D656 15                  DEC D
D657 15                  DEC D              ;Z=KEY OBTAINED. D=0 FOR NO SHIFT
D658
D658 ED53065C    KBLD:   LD (LASTKV),DE
D65C C9                  RET
D65D                                             ;KEYRD2
D65D
D65D                     INCLUDE SCRSEL2.SAM     ; GOTO, CONT, GETTOKEN, MODPT2, AUTO
D65D             ;SCRSEL2.SAM.
D65D
D65D 219ED6      GOSUB:  LD HL,GOSUB2
D660 DD                  DB 0DDH            ;"JR+3"
D661
D661 219219      GOTO:   LD HL,GOTO2
D664
D664 E5          GSUBTOC:PUSH HL
D665 FEDE                CP 0DEH            ;ONTOK
D667 2808                JR Z,GOTON
D669
D669 CDE43A              CALL EXPT1NUM      ;LINE NUMBER
```

```
D66C E1                  POP HL             ;ADDR OF GOTO OR GOSUB ROUTINE
D66D CD153B              CALL CHKEND
D670
D670 E9                  JP (HL)
D671
D671            ;E.G. GOSUB (OR GOTO) ON X;100,234,400
D671
D671 CDE33A     GOTON:   CALL SEXPT1NUM     ;SKIP 'ON', EVAL NUMBER
D674 F5                  PUSH AF
D675 FE3B                CP ":"
D677 C2290D              JP NZ,NONSENSE
D67A
D67A F1                  POP AF
D67B 3809                JR C,GOTON2        ;JR IF RUNNING
D67D
D67D E1                  POP HL             ;GOTO/GOSUB ADDR
D67E
D67E CDE33A     GOTONSL: CALL SEXPT1NUM     ;SKIP E.G. 100,200,400
D681 FE2C                CP ","
D683 28F9                JR Z,GOTONSL
D685
D685 C9                  RET                ;TO NEXT STAT
D686
D686 CD651D     GOTON2:  CALL FPTOA         ;GET NUMBER AFTER 'ON' IN C AND A
D689                                        ;CY IF >255
D689 9F                  SBC A,A            ;A=255 IF OUT OF RANGE, ELSE 0
D68A 2801                JR Z,GOTONRL       ;C IS OK IF A=0
D68C
D68C 4F                  LD C,A             ;ELSE LET C=255
D68D
D68D C5         GOTONRL: PUSH BC            ;C=EXPR NUMBER
D68E CDE33A              CALL SEXPT1NUM
D691 C1                  POP BC
D692 0D                  DEC C
D693 C8                  RET Z              ;RET TO GOTO OR GOSUB IF THIS IS DESIRED EXPR
D694
D694 CD121D              CALL FDELETE       ;DISCARD EXPR FROM FPCS
D697 DF                  RST 18H
D698 FE2C                CP ","
D69A 28F1                JR Z,GOTONRL       ;LOOP IF THERE ARE MORE LINE NOS TO SKIP
D69C
D69C D1                  POP DE             ;GOTO/GOSUB ADDR
D69D C9                  RET                ;TO NEXT STAT - LINE NOS RAN OUT.
D69E
D69E CD2E1D     GOSUB2:  CALL GETINT
D6A1 7C                  LD A,H
D6A2 3C                  INC A
D6A3 CA391D              JP Z,IOORERR
D6A6
D6A6 0600                LD B,00H           ;'GOSUB' TYPE
D6A8 E5                  PUSH HL            ;LINE NO.
D6A9 CDF118              CALL BSTKE         ;STACK RETURN ADDR
D6AC E1                  POP HL
D6AD C39A19              JP GOTO3
D6B0
D6B0            ;GET TOKEN SR - VIA JUMP TABLE
D6B0
D6B0 4F         GETTOKEN: LD C,A
D6B1 08                  EX AF,AF'          ;NO OF WORDS TO CHECK,+1
D6B2
D6B2 23         GTTOK1:  INC HL
D6B3
D6B3 CB7E       GTTOK2:  BIT 7,(HL)
D6B5 28FB                JR Z,GTTOK1        ;LOOP TILL WE FIND THE END OF A KEYWORD
D6B7                                        ;32 TS PER LOOP
D6B7
D6B7 0D                  DEC C
D6B8 C8                  RET Z              ;RET IF WE HAVE CHECKED THEM ALL - Z=FAILED
D6B9                                        ;DE=FIRST LETTER OF ELINE WORD COPY
D6B9 23                  INC HL
D6BA 1A                  LD A,(DE)
D6BB AE                  XOR (HL)
D6BC E6DF                AND 0DFH
D6BE 20F3                JR NZ,GTTOK2       ;LOOP BACK AND CHECK ANOTHER WORD IF MATCH FAILS
D6C0                                        ;ABOUT 80 TS PER LOOP IF NO MATCH. ABOUT 35USEC
D6C0                                        ;PER 5-LETTER WORD OR 5.3MSEC FOR 150 WORDS.
D6C0 D5                  PUSH DE            ;PTR TO FIRST LETTER IN ELINE WORD
D6C1 13                  INC DE
D6C2
D6C2 23         GTTOK3:  INC HL
D6C3 7E                  LD A,(HL)
D6C4 FE20                CP " "
D6C6 1A                  LD A,(DE)
D6C7 13                  INC DE
D6C8 2004                JR NZ,GTTOK4       ;JR IF NO EMBEDDED SPACE IN KEYWORD IN LIST
D6CA
D6CA BE                  CP (HL)
D6CB 28F5                JR Z,GTTOK3        ;ELSE ACCEPT ONE IN INPUT
D6CD
D6CD 23                  INC HL             ;BUT DON'T INSIST - SKIP LIST PTR IF NO SP IN INPUT
D6CE
D6CE AE         GTTOK4:  XOR (HL)
D6CF E6DF                AND 0DFH
D6D1 28EF                JR Z,GTTOK3        ;LOOP IF STILL MATCHING OK
D6D3
D6D3 E67F                AND 7FH
D6D5 2004                JR NZ,GTTOK5       ;JR IF INPUT AND LIST WORDS DON'T END NOW
D6D7
D6D7 7E                  LD A,(HL)
D6D8 07                  RLCA
D6D9 3803                JR C,GTTOK6        ;OK IF LIST WORD HAS FINISHED
D6DB
D6DB D1         GTTOK5:  POP DE
```

```
D6DC 18D5                  JR GTTOK2
D6DE
D6DE FE7E       GTTOK6:    CP 7EH            ;('=' TERMINATOR =7B AFTER RLCA, '>'=7DH, '$'=49H)
D6E0 3F                    CCF               ;NC IF '=' OR '>' OR '$'
D6E1 1A                    LD A,(DE)         ;INPUT CHAR AFTER MATCHED WORD
D6E2 DC373B                CALL C,ALDU       ;CALL TO INSIST ON NON-LETTER AFTER E.G. PRINTx
D6E5                                         ;BUT NOT AFTER <>,>= OR <= OR CHR$a
D6E5 38F4                  JR C,GTTOK5       ;('=' AND '>' NEVER CALL ALPHA OR JR HERE)
D6E7
D6E7 E1                    POP HL            ;FIRST LETTER OF ELINE WORD COPY
D6E8 08                    EX AF,AF'
D6E9 91                    SUB C             ;A=1 IF FIRST WORD MATCHES, ETC.
D6EA C9                    RET
D6EB
D6EB
D6EB E603       MODPT2:    AND 03H
D6ED 21405A                LD HL,MODE
D6F0 4E                    LD C,(HL)
D6F1 C5                    PUSH BC           ;SAVE PREVIOUS MODE
D6F2 77                    LD (HL),A         ;NEW MODE
D6F3 FE02                  CP 2
D6F5 F5                    PUSH AF
D6F6 D41AD8                CALL NC,SUET      ;SET UP EXPANSION TABLE IF NEEDED
D6F9
D6F9 F1                    POP AF
D6FA F5                    PUSH AF           ;MODE
D6FB 0F                    RRCA
D6FC 0F                    RRCA
D6FD 0F                    RRCA              ;MODE BITS TO BITS 6 AND 5
D6FE 4F                    LD C,A
D6FF 21785A                LD HL,CUSCRNP
D702 7E                    LD A,(HL)         ;KEEP CURRENT VID PAGE,
D703 A9                    XOR C             ;BUT
D704 E69F                  AND 9FH           ;USE BITS 6 AND 5 OF C TO SET MODE
D706 A9                    XOR C
D707 77                    LD (HL),A
D708 DBFC                  IN A,(VIDPORT)
D70A AE                    XOR (HL)
D70B E61F                  AND 1FH
D70D 2003                  JR NZ,MDL2        ;JR IF WE ARE NOT DISPLAYING THIS SCREEN - AVOID
D70F                                         ;ALTERING HARDWARE MODE
D70F 7E                    LD A,(HL)
D710 D3FC                  OUT (VIDPORT),A
D712
D712 F1         MDL2:      POP AF            ;NEW MODE
D713 C1                    POP BC            ;C=PREV MODE
D714 F5                    PUSH AF           ;NEW MODE
D715 FE01                  CP 1              ;CY IF MODE 0
D717 9F                    SBC A,A           ;MODE 0=FF, REST=0
D718 210908                LD HL,0809H       ;CHAR WIDTH/HEIGHT FOR MODES 1-3
D71B 85                    ADD A,L
D71C 6F                    LD L,A            ;DEC HEIGHT TO 8 IF MODE 0, TO FIT ATTR
D71D F1                    POP AF
D71E
D71E CD6CD7                CALL MDSR
D721
D721 3ACE5A                LD A,(TEMPB1)     ;OLD MODE
D724 2813                  JR Z,M2ST         ;JR IF MODE 2 BEING SET
D726
D726 FE02                  CP 2
D728 203D                  JR NZ,ECLS        ;JR IF NOT MODE 2 NOW, AND NOT CHANGING TO MODE 2
D72A
D72A                       ;ELSE IF CHANGING FROM MODE 2 TO ANOTHER MODE, SAVE M2 PALTAB
D72A                       ; 0-3, GET BACK NON-MODE-2 COLOURS
D72A
D72A 2A495B                LD HL,(M3PAPP)
D72D 22485A                LD (M23PAPP),HL
D730 2A4B5B                LD HL,(M3LSC)
D733 22305A                LD (M23LSC),HL
D736 4F                    LD C,A            ;C=NZ - 'HALVE'
D737 1823                  JR M2ST2
D739
D739            ;MODE 2 BEING SET
D739
D739 FE02       M2ST:      CP 2
D73B 282A                  JR Z,ECLS         ;JR IF MODE 2 ALREADY
D73D
D73D 2A485A                LD HL,(M23PAPP)
D740 22495B                LD (M3PAPP),HL
D743 CD80F1                CALL M3TO2        ;CONVERT L
D746 F5                    PUSH AF
D747 6C                    LD L,H
D748 CD80F1                CALL M3TO2        ;CONVERT H
D74B 65                    LD H,L
D74C F1                    POP AF
D74D 6F                    LD L,A
D74E 22485A                LD (M23PAPP),HL   ;PREVENTS STRIPED INKS AFTER MODE 2 SET.
D751 2A305A                LD HL,(M23LSC)
D754 224B5B                LD (M3LSC),HL
D757 CD7BF1                CALL CVLSP        ;CONVERT LS PAPER
D75A 0E00                  LD C,0            ;C='HALVE'
D75C
D75C 3A445A     M2ST2:     LD A,(THFATP)     ;WE ARE CHANGING TO/FROM MODE 2 -  IF
D75F A7                    AND A             ;'THIN' PIXEL STATUS WILL CHANGE,
D760 79                    LD A,C
D761 CCCBD7                CALL Z,SETFP      ;FIDDLE XCOORD/XRG
D764
D764 CDBBED                CALL PALSW
D767
D767 CDFA01     ECLS:      CALL R1OFFJP
D76A 9806                  DW MCLS           ;TURN ROM 1 OFF, JP MCLS
D76C
D76C            ;CALLED FROM CSIZE WITH A AND C=MODE, HL=CSIZE
```

```
D76C
D76C F5          MDSR:      PUSH AF           ;NEW MODE
D76D 79                     LD A,C            ;PREV MODE (SAME IF CALLED FROM CSIZE)
D76E 32CE5A                 LD (TEMPB1),A
D771 22365A                 LD (CSIZE),HL     ;H=WIDTH, L=HEIGHT
D774 3EC0                   LD A,192
D776 1EFC                   LD E,0FCH
D778
D778 1C          MDSL:      INC E
D779 95                     SUB L             ;SUB CHAR HEIGHT
D77A 30FC                   JR NC,MDSL        ;CALC SCREEN HEIGHT IN LINES, MINUS 3
D77C                                          ;(UW BOT)
D77C 85                     ADD A,L           ;'LEFT OVER' PIXELS
D77D 325D5A                 LD (LSOFF),A      ;LOWER SCREEN OFFSET - SO LEFT-OVER PIX BETWEEN
D780                                          ;UPPER SCREEN AND LOWER SCREEN
D780 7D                     LD A,L
D781 87                     ADD A,A           ;DISP OF GRAPHICS ORIGIN FROM SCREEN BOTTOM IS
D782 325C5A                 LD (ORGOFF),A     ;2 CHAR HEIGHTS
D785 161F                   LD D,31           ;RHS FOR MODES 0, 1 AND 3
D787 F1                     POP AF            ;MODE
D788 F5                     PUSH AF
D789 200F                   JR NZ,MDSR3       ;JR IF NOT MODE 2
D78B
D78B 163F                   LD D,63           ;RHS FOR MODE 2 64-COL MODE
D78D 3A355A                 LD A,(FL6OR8)
D790 A7                     AND A
D791 2007                   JR NZ,MDSR3       ;JR IF 8-BIT
D793
D793 1654                   LD D,84           ;RHS FOR MODE 2 85-COL MODE
D795 3E06                   LD A,6
D797 32375A                 LD (CSIZE+1),A    ;WIDTH
D79A
D79A 21385A      MDSR3:     LD HL,UWRHS
D79D 72                     LD (HL),D         ;UWRHS
D79E 23                     INC HL
D79F AF                     XOR A
D7A0 77                     LD (HL),A         ;UWLHS
D7A1 23                     INC HL
D7A2 77                     LD (HL),A         ;UWTOP
D7A3 23                     INC HL
D7A4 73                     LD (HL),E         ;UWBOT
D7A5 23                     INC HL
D7A6 72                     LD (HL),D         ;LWRHS
D7A7 ED535A5A               LD (WINDMAX),DE
D7AB 23                     INC HL
D7AC 77                     LD (HL),A         ;LWLHS
D7AD 23                     INC HL
D7AE 1C                     INC E
D7AF 73                     LD (HL),E         ;LWTOP=UWBOT+1
D7B0 23                     INC HL
D7B1 1C                     INC E
D7B2 73                     LD (HL),E         ;LWBOT=LWTOP+1
D7B3 F1                     POP AF            ;Z IF MODE 2
D7B4 C9                     RET
D7B5
D7B5             ;FATPIX 0=USE THIN PIXELS IN MODE 2. 1=USE FAT PIXELS
D7B5
D7B5 CD583A      FATPIX:    CALL SYNTAX6
D7B8
D7B8 111E02                 LD DE,0200H+30    ;LIMIT TO <2 OR IOOR
D7BB CD621F                 CALL LIMBYTE
D7BE
D7BE 21445A                 LD HL,THFATP
D7C1 BE                     CP (HL)
D7C2 C8                     RET Z             ;RET IF NO CHANGE IN FATPIX STATUS
D7C3
D7C3 77                     LD (HL),A
D7C4 3A405A                 LD A,(MODE)
D7C7 FE02                   CP 2
D7C9 C0                     RET NZ            ;DON'T FIDDLE XCOORD AND XRG UNLESS MODE 2
D7CA
D7CA 7E                     LD A,(HL)
D7CB
D7CB 2A425A      SETFP:     LD HL,(XCOORD)
D7CE A7                     AND A
D7CF 2805                   JR Z,FPX2         ;JR IF CHANGED FROM 1 TO 0 (FAT TO THIN)
D7D1
D7D1 CB3C                   SRL H
D7D3 CB1D                   RR L              ;HALVE X
D7D5 3E                     DB 3EH            ;'JR+1'
D7D6
D7D6 29          FPX2:      ADD HL,HL         ;DOUBLE X COORD TO KEEP SAME SCREEN POSN
D7D7
D7D7 22425A                 LD (XCOORD),HL    ;A=0 FOR DOUBLE, 29H FOR HALVE
D7DA
D7DA F5                     PUSH AF
D7DB CD1F1F                 CALL ADDRNV
D7DE F1                     POP AF
D7DF 115700                 LD DE,87
D7E2 19                     ADD HL,DE         ;PT TO XRG
D7E3 CDFA01                 CALL R1OFFJP
D7E6 152B                   DW CGXRG          ;CHANGE XRG WITH ROM1 OFF **
D7E8
D7E8 CF          PXIOOR:    RST 08H
D7E9 1E                     DB 30             ;IOOR
D7EA
D7EA             ;CSIZE W,H
D7EA             ;WIDTH CAN BE 6 OR 8 BUT 6 ONLY EFFECTIVE IN MODE 2
D7EA             ;HEIGHT CAN BE 6-32, DOUBLE HEIGHT USED IF 16 OR MORE.
D7EA
D7EA CD5F3A      WIDTH:     CALL SYNTAX8
D7ED
D7ED CD331D                 CALL GETBYTE      ;H
```

```
D7F0 C5                 PUSH BC
D7F1 CD331D             CALL GETBYTE      ;W
D7F4 FE06               CP 6
D7F6 2804               JR Z,CSZ2
D7F8
D7F8 FE08               CP 8
D7FA 20EC               JR NZ,PXIOOR
D7FC
D7FC D1         CSZ2:   POP DE            ;E=H
D7FD 57                 LD D,A            ;D=W
D7FE 7B                 LD A,E
D7FF FE06               CP 6
D801 38E5               JR C,PXIOOR       ;HEIGHTS OF 0-5 ARE CRAZY - BUT 6-7 MIGHT BE OK
D803                                      ;WITH THE RIGHT CHARACTER SET.
D803 FE21               CP 33
D805 30E1               JR NC,PXIOOR      ;HEIGHTS OF 6-32 ARE ALLOWED
D807
D807 EB                 EX DE,HL
D808 7C                 LD A,H            ;WIDTH
D809 D606               SUB 6
D80B 32355A             LD (FL6OR8),A     ;0 IF WIDTH 6, ELSE NZ
D80E 3A405A             LD A,(MODE)
D811 4F                 LD C,A            ;'PREVIOUS MODE' NEEDED BY MDST2
D812 FE02               CP 2              ;SET Z IF MODE 2
D814 F5                 PUSH AF
D815 CD6CD7             CALL MDSR         ;MODE - SET WIDOWS, CSIZE. HL=CSIZE
D818 F1                 POP AF
D819 C0                 RET NZ            ;RET IF NOT MODE 2
D81A
D81A                                      ;ELSE SET UP EXPANSION TABLE IF MODE 2, IN CASE
D81A                                      ;SWITCHING BETWEEN M2/M3 O/P TYPES
D81A
D81A            ;SET UP EXPANSION TABLE. ENTRY: A=MODE 2/3, Z IF 2
D81A
D81A 21205B     SUET:   LD HL,EXTAB       ;PT TO EXPANSION TABLE
D81D 0E00               LD C,0            ;FIRST VALUE TO EXPAND
D81F 280F               JR Z,DBTABCLP     ;JR IF MODE 2
D821
D821 79         QUADTCLP:  LD A,C
D822 CD3DD8             CALL QUADBITS     ;A->DE, QUADED BITS
D825 72                 LD (HL),D
D826 23                 INC HL
D827 73                 LD (HL),E
D828 23                 INC HL
D829 0C                 INC C
D82A 79                 LD A,C
D82B FE10               CP 16
D82D 38F2               JR C,QUADTCLP     ;QUAD 00-0FH
D82F
D82F C9                 RET
D830
D830 79         DBTABCLP:  LD A,C
D831 CD41D8             CALL DBBITS       ;A->E, DOUBLED BITS
D834 73                 LD (HL),E
D835 23                 INC HL
D836 0C                 INC C
D837 79                 LD A,C
D838 FE10               CP 16
D83A 38F4               JR C,DBTABCLP     ;DOUBLE 00-0FH
D83C
D83C C9                 RET
D83D
D83D            ;A->DE, QUADED BITS
D83D
D83D CD41D8     QUADBITS:  CALL DBBITS
D840 7B                 LD A,E
D841
D841            ;A->DE WITH ALL BITS DOUBLED
D841
D841 0608       DBBITS:    LD B,8         ;BITS TO DO
D843
D843 0F         DBBITSLP:  RRCA
D844 CB1A               RR D
D846 CB1B               RR E              ;DE HAS DOUBLED BITS
D848 07                 RLCA
D849 0F                 RRCA
D84A CB1A               RR D
D84C CB1B               RR E
D84E 10F3               DJNZ DBBITSLP
D850
D850 C9                 RET
D851
D851            ;AUTO <LINE><,STEP>
D851
D851 CD7A3A     AUTO:   CALL CRCOLON
D854 2811               JR Z,AO1          ;JR IF JUST 'AUTO', USE BOTH DEFAULTS
D856
D856 CD8E3B             CALL GIR2         ;GET LINE VALUE IN HL AND BC
D859 010A00             LD BC,10          ;DEFAULT STEP IS 10
D85C FE2C               CP ","
D85E 200E               JR NZ,AO2         ;JR IF NO STEP
D860
D860 E5                 PUSH HL           ;SAVE LINE VALUE
D861 CD8D3B             CALL GIR          ;GET STEP VALUE IN HL,BC
D864 E1                 POP HL
D865 1807               JR AO2
D867
D867 2A495C     AO1:    LD HL,(EPPC)
D86A 010A00             LD BC,10
D86D 09                 ADD HL,BC         ;DEFAULT LINE VALUE IS EPPC+10
D86E
D86E CDC63A     AO2:    CALL RUNFLG
D871 D0                 RET NC            ;CANNOT USE CHKEND - NEED C
```

```
D872
D872 AF                    XOR A
D873 ED42                  SBC HL,BC
D875 3F                    CCF
D876 8F                    ADC A,A          ;A=0 IF STEP>LINE, ELSE A=1
D877 32885B                LD (AUTOFLG),A   ;0=OFF, NZ=ON
D87A C8                    RET Z            ;RET IF OFF
D87B
D87B 22495C                LD (EPPC),HL     ;INITIAL EPPC=LINE-STEP
D87E ED43895B              LD (AUTOSTEP),BC
D882 D1                    POP DE           ;NEXT STAT
D883 D1                    POP DE           ;ERROR HANDLER
D884 01750E                LD BC,AULL
D887 C31F02                JP R1XJP         ;TURN ROM1 OFF, JP MAINX
D88A
D88A 11004F      SOUND:    LD DE,INSTBUF
D88D
D88D D5          SNDLP:    PUSH DE
D88E CDDC3A                CALL EXPT2NUMS
D891 D1                    POP DE
D892 3015                  JR NC,SND1
D894
D894 D5                    PUSH DE
D895 CD331D                CALL GETBYTE
D898 F5                    PUSH AF
D899 111E20                LD DE,2000H+30
D89C CD621F                CALL LIMBYTE     ;ALLOW 0-31
D89F C1                    POP BC
D8A0 D1                    POP DE
D8A1 12                    LD (DE),A
D8A2 1C                    INC E
D8A3 78                    LD A,B
D8A4 12                    LD (DE),A
D8A5 1C                    INC E
D8A6 CA664F                JP Z,NRFLERR     ;LIMIT TO 254 VALUES IN INSTBUF (127 PAIRS)
D8A9
D8A9 DF          SND1:     RST 18H
D8AA FE3B                  CP ":"
D8AC 2003                  JR NZ,SND2
D8AE
D8AE E7                    RST 20H          ;SKIP ';'
D8AF 18DC                  JR SNDLP
D8B1
D8B1 CD153B      SND2:     CALL CHKEND
D8B4
D8B4 3EFF                  LD A,0FFH
D8B6 12                    LD (DE),A        ;TERMINATE LIST
D8B7 01FF01                LD BC,256+SNDPORT ;SOUND ADDRESS REG PORT
D8BA 21004F                LD HL,INSTBUF    ;HOLDS AT LEAST ONE PAIR
D8BD 1806                  JR SND3
D8BF
D8BF ED51        SNDOPL:   OUT (C),D        ;SET REG
D8C1 05                    DEC B            ;SOUND DATA REG PORT
D8C2 ED59                  OUT (C),E        ;SEND DATA
D8C4 04                    INC B
D8C5
D8C5 56          SND3:     LD D,(HL)
D8C6 2C                    INC L
D8C7 5E                    LD E,(HL)
D8C8 2C                    INC L
D8C9 BA                    CP D
D8CA 20F3                  JR NZ,SNDOPL     ;LOOP UNTIL TERMINATOR HIT
D8CC
D8CC C9                    RET
D8CD
D8CD
D8CD             ;PERFORM BOOT ACTION
D8CD             ;FIND A PAGE FOR DOS AND MARK IT.
D8CD
D8CD CD503A      BOOT:     CALL SYNTAX3
D8D0
D8D0 CD331D                CALL GETBYTE
D8D3 A7                    AND A
D8D4 200F                  JR NZ,BOOTEX     ;E.G. BOOT 1 FORCES BOOT OR RE-BOOT
D8D6                                        ;NO AUTO-LOAD
D8D6 3AC25B                LD A,(DOSFLG)
D8D9 A7                    AND A
D8DA 2803                  JR Z,BOOTNR      ;JR IF DOS NOT RESIDENT - ELSE AUTOLOAD ONLY
D8DC
D8DC CF                    RST 08H
D8DD 88                    DB ALHK          ; DO AUTO-LOAD
D8DE C9                    RET
D8DF
D8DF CDE5D8      BOOTNR:   CALL BOOTEX
D8E2
D8E2 CF                    RST 08H
D8E3 80                    DB BTHK          ;DO AUTO-LOAD, BUT NO ERROR IF NONE
D8E4 C9                    RET
D8E5
D8E5 211F51      BOOTEX:   LD HL,ALLOCT+1FH
D8E8
D8E8 7E          FDPL:     LD A,(HL)
D8E9 A7                    AND A
D8EA 2809                  JR Z,GDP         ;JR IF FREE PAGE
D8EC
D8EC FE60                  CP 60H
D8EE 2805                  JR Z,GDP         ;JR IF RE-BOOTING TO A DOS PAGE
D8F0
D8F0 2D                    DEC L
D8F1 20F5                  JR NZ,FDPL
D8F3
D8F3 CF                    RST 08H
D8F4 01                    DB 1             ;'OUT OF MEMORY' IF NO FREE PAGE FOR DOS
```

```
D8F5
D8F5 7D          GDP:       LD A,L
D8F6 CDDF3F                 CALL SELURPG      ;DOS PAGE IN AT 8000H
D8F9
D8F9                                          ;START OF BRUCE GORDONS CODE
D8F9                                          ;LOAD STUFF NOW
D8F9 0ED0                   LD C,0D0H         ;RESET CHIP
D8FB CD81D9                 CALL SDCX         ;EXITS WITH B=0
D8FE CD89D9                 CALL REST
D901
D901                        ;TEST FOR INDEX HOLE
D901
D901 26FE                   LD H,0FEH         ;HL=FEXX
D903 5C                     LD E,H            ;E COUNTS 2 OUTER LOOPS
D904 0606                   LD B,6
D906
D906 2B          BOOT2:     DEC HL
D907 7C                     LD A,H
D908 B5                     OR L
D909 2005                   JR NZ,BOOT3
D90B
D90B 1C                     INC E
D90C 2002                   JR NZ,BOOT3
D90E
D90E CF                     RST 08H           ;'Missing disc'
D90F 37                     DB 55
D910
D910 DBE0        BOOT3:     IN A,(COMM)
D912 57                     LD D,A
D913 A9                     XOR C
D914 E602                   AND 2
D916 28EE                   JR Z,BOOT2        ;LOOP UNTIL HOLE SIGNAL CHANGES
D918
D918 4A                     LD C,D
D919 10EB                   DJNZ BOOT2
D91B
D91B CD89D9                 CALL REST
D91E 110104                 LD DE,0401H       ;TRACK/SECTOR
D921
D921             ;READ SECTOR AT TRACK D, SECTOR E
D921
D921 AF          RSAD:      XOR A
D922 08                     EX AF,AF'
D923
D923 7B          RSA1:      LD A,E
D924 D3E2                   OUT (SECT),A
D926
D926 CD8ED9      RSA2:      CALL BUSY
D929 DBE1                   IN A,(TRCK)
D92B BA                     CP D
D92C 280B                   JR Z,RSA4
D92E
D92E 0E79                   LD C,STPOUT
D930 3002                   JR NC,RSA3
D932
D932 0E59                   LD C,STPIN
D934
D934 CD7ED9      RSA3:      CALL SADC
D937 18ED                   JR RSA2
D939
D939 F3          RSA4:      DI
D93A 0E80                   LD C,DRSEC
D93C CD7ED9                 CALL SADC
D93F 210080                 LD HL,8000H
D942 01E300                 LD BC,DTRQ
D945 FE                     DB 0FEH           ;"JR+2" (CP EDH: AND D)
D946
D946 EDA2        RSA5:      INI
D948
D948 DBE0        RSA6:      IN A,(COMM)
D94A CB4F                   BIT 1,A
D94C 20F8                   JR NZ,RSA5
D94E
D94E 0F                     RRCA
D94F 38F7                   JR C,RSA6
D951
D951 FB                     EI
D952
D952             ;CHECK DISC ERROR COUNT
D952
D952 E60E                   AND 0EH
D954 2811                   JR Z,BTNOE        ;JR IF NO ERRORS
D956
D956 08                     EX AF,AF'
D957 3C                     INC A
D958 FE05                   CP 5
D95A F5                     PUSH AF
D95B CC89D9                 CALL Z,REST
D95E F1                     POP AF
D95F FE0A                   CP 10
D961 D2BEE2                 JP NC,TERROR
D964
D964 08                     EX AF,AF'
D965 18BC                   JR RSA1
D967
D967 11FF80      BTNOE:     LD DE,80FFH
D96A 2194FB                 LD HL,BTWD
D96D 0604                   LD B,4
D96F
D96F 13          BTCK:      INC DE
D970 1A                     LD A,(DE)
D971 AE                     XOR (HL)
D972 E65F                   AND 5FH           ;IGNORE MISMATCH ON BITS 7 OR 5
```

```
D974 2802                  JR Z,BTLY
D976
D976 CF                    RST 08H
D977 35                    DB 53             ;'NO DOS' IF NOT 'BOOT'
D978
D978 23           BTLY:    INC HL
D979 10F4                  DJNZ BTCK
D97B
D97B C30980                JP 8009H          ;DOS INITIALISE
D97E
D97E CD8ED9       SADC:    CALL BUSY
D981
D981 79           SDCX:    LD A,C
D982 D3E0                  OUT (COMM),A
D984 0600                  LD B,0
D986
D986 10FE         SDC1:    DJNZ SDC1
D988
D988 C9                    RET
D989
D989              ;RESTORE DRIVE TO ZERO
D989
D989 0E09         REST:    LD C,DRES
D98B CD7ED9                CALL SADC
D98E
D98E              ;TEST FOR CHIP BUSY
D98E
D98E DBE0         BUSY:    IN A,(COMM)
D990 0F                    RRCA
D991 D0                    RET NC
D992
D992 CD5D0E                CALL BRKCR
D995 18F7                  JR BUSY
D997
D997                       INCLUDE PRINTFP.SAM
D997              ;PRINTFP.SAM - PRINT A NUMBER
D997
D997
D997              ;*************************************************************************
D997              ;RETURN STR$ OF NUMBER ON FPCS AS BC BYTES AT (DE) IN BUFFER
D997
D997 3E06         PFSTRS:  LD A,6
D999
D999 329A5B       PFSTRSC: LD (FRACLIM),A    ;UP TO 4 LEADING ZEROS IN FRACTIONS BEFORE EFORM
D99C CDC5DA               CALL PRFPBUF       ;IS USED
D99F 3A9D5B               LD A,(DIGITS)
D9A2 A7                   AND A
D9A3 201A                 JR NZ,PFNRND       ;JR IF LESS THAN 9 CHARS - RESULT IS EXACT
D9A5                                         ;ELSE ROUND UP ON 9TH. DIGIT. (NOT A DEC. PT)
D9A5 2A9B5B               LD HL,(NPRPOS)
D9A8 2B                   DEC HL
D9A9 229B5B               LD (NPRPOS),HL     ;IGNORE 9TH. DIGIT OTHER THAN FOR ROUNDING
D9AC 7E                   LD A,(HL)
D9AD FE35                 CP "5"
D9AF 380E                 JR C,PFNRND        ;NO ROUND IF E.G. 12345678.4
D9B1
D9B1 3630         PFRNDLP: LD (HL),"0"
D9B3
D9B3 2B           PFPSLP:  DEC HL
D9B4 7E                    LD A,(HL)
D9B5 FE2E                  CP "."
D9B7 28FA                  JR Z,PFPSLP
D9B9
D9B9 3C                    INC A
D9BA FE3A                  CP "9"+1
D9BC 30F3                  JR NC,PFRNDLP      ;JR IF OVERFLOWED - 0 THIS DIGIT AND KEEP ROUNDING
D9BE                                          ;EXTREME CASE IS ROUND RIGHT BACK TO FIRST ZERO
D9BE                                          ;BEFORE EXITING LOOP
D9BE 77                    LD (HL),A
D9BF
D9BF 2A9B5B       PFNRND:  LD HL,(NPRPOS)
D9C2 11A05B               LD DE,PRNBUFF
D9C5 A7                   AND A
D9C6 ED52                 SBC HL,DE
D9C8 4D                   LD C,L
D9C9 44                   LD B,H
D9CA 2D                   DEC L
D9CB C8                   RET Z              ;IF ONLY 1 CHAR, MUST BE ZERO - LEN 1, AT (DE)
D9CC
D9CC 19                   ADD HL,DE
D9CD 23                   INC HL             ;HL=NPRPOS
D9CE 3A9F5B               LD A,(DECPNTED)
D9D1 A7                   AND A
D9D2 2006                 JR NZ,MTRZDLP      ;DELETE TRAILING ZEROS IF DEC PT USED E.G. 1.2300
D9D4
D9D4 3A9E5B       EFCHECK: LD A,(EPOWER)
D9D7 A7                   AND A
D9D8 280D                 JR Z,PFNPNT        ;JR IF NOT EFORM - LEAVE TRAILING ZEROS
D9DA
D9DA 0D           MTRZDLP: DEC C
D9DB 2B                   DEC HL
D9DC 7E                   LD A,(HL)
D9DD FE30                 CP "0"
D9DF 28F9                 JR Z,MTRZDLP       ;DELETE TRAILING ZEROS
D9E1
D9E1 7E                   LD A,(HL)
D9E2 FE2E                 CP "."
D9E4 28EE                 JR Z,EFCHECK       ;KEEP "." DELETED AND GOTO PFNPNT IF NOT EFORM
D9E6                                         ;ELSE PNT IS NOT REAL - DELETE MORE ZEROS IF NEEDED
D9E6
D9E6 0C                   INC C              ;ELSE "UNDELETE" LAST NON-"0"/NON-"." CHAR
D9E7
D9E7 1A           PFNPNT:  LD A,(DE)
```

```
D9E8 FE2D                CP "-"
D9EA 2002                JR NZ,PFNMIN     ;SKIP A FIRST "-" IF NEEDED
D9EC
D9EC 13                  INC DE
D9ED 1A                  LD A,(DE)
D9EE
D9EE FE30       PFNMIN:  CP "0"
D9F0 2020                JR NZ,PFEXIT     ;JR IF FIRST CHAR USED IN ROUNDING
D9F2                                      ;(=div by 10 if fract: 0998->999 but 0999->1000)
D9F2 3A9E5B              LD A,(EPOWER)
D9F5 3D                  DEC A            ;0->FF
D9F6 FE80                CP 80H           ;CY IF +VE, NZ EPOWER (FRACTION) (80H NOT CRITICAL)
D9F8 CE01                ADC A,1          ;A=ORIG IF 0 OR -VE, ELSE A=A+1
D9FA FE08                CP 8
D9FC 2803                JR Z,PFNEZ       ;JR IF FRACTION ROUNDED UP TO 1
D9FE
D9FE 329E5B              LD (EPOWER),A    ;CAN PUT IN AN EXTRA LEAD ZERO TO COMP FOR
DA01                                      ;"MISSED" DIV COS FIRST ZERO NOT DELETED.
DA01
DA01 0D          PFNEZ:  DEC C            ;COMPENSATE LEN FOR LEADING ZERO TO BE LOST
DA02 79                  LD A,C
DA03 2A9B5B              LD HL,(NPRPOS)
DA06 2B                  DEC HL
DA07 229B5B              LD (NPRPOS),HL   ;COMPENSATE NPRPOS IN CASE 9-DIGIT INTEGER LOOP
DA0A                                      ;USED (SEE LATER)
DA0A 62                  LD H,D
DA0B 6B                  LD L,E
DA0C 23                  INC HL
DA0D 0E0A                LD C,10
DA0F EDB0                LDIR             ;ALIGN TO LHS E.G. 01.23 -> 1.23
DA11 4F                  LD C,A
DA12
DA12 11A05B      PFEXIT: LD DE,PRNBUFF
DA15 3A9D5B              LD A,(DIGITS)
DA18 219F5B              LD HL,DECPNTED
DA1B B6                  OR (HL)
DA1C 6F                  LD L,A           ;L=0 IF NINE DIGIT INTEGER
DA1D 3A9E5B              LD A,(EPOWER)
DA20 67                  LD H,A
DA21 A7                  AND A
DA22 7D                  LD A,L
DA23 2008                JR NZ,PFEFORM    ;JR IF EFORM (FRACTIONS ARE ALWAYS NOMINALLY EFORM)
DA25
DA25             ;NOW CHECK FOR 9-DIGIT INTEGERS WHICH SHOULD BE EFORM TOO
DA25
DA25 A7                  AND A
DA26 C0                  RET NZ           ;RET IF NOT 9 DIGIT INTEGER
DA27
DA27 3D                  DEC A
DA28 329E5B              LD (EPOWER),A    ;POWER FF IS CORRECT FOR E+8
DA2B 1892                JR PFNRND        ;PRINT EFORMAT NUMBER
DA2D
DA2D A7          PFEFORM: AND A
DA2E 2801                JR Z,PFPOWOK     ;IF A 9-DIGIT INTEGER WAS PRODUCED
DA30                                      ;BY MULTIPLYING BY A POWER - JUMP. IF NOT, FORM
DA30                                      ;IS EG 12345678.9 OR 12345678, BECAUSE POWER
DA30                                      ;CHOSEN WAS TOO -VE (IN THE CASE OF BIG NUMBERS)
DA30                                      ;OR TOO SMALL (IN THE CASE OF SMALL NUMBERS)
DA30                                      ;IN WHICH CASE RESULT IS EFFECTIVELY 0.XE+/-H
DA30 24                  INC H            ;CHANGE TO X.XE+/-H
DA31
DA31 3A9A5B      PFPOWOK: LD A,(FRACLIM)   ;NORMALLY 6, TO ALLOW UP TO 4 LEADING ZEROS
DA34 6F                  LD L,A
DA35 7C                  LD A,H
DA36 D608                SUB 8
DA38 283A                JR Z,PFSPBH      ;JR IF ROUNDING MEANS *NO* LEADING ZEROS
DA3A                                      ;(0.999999->1)
DA3A BD                  CP L
DA3B 3845                JR C,PFFRACT     ;JR WITH 1 FOR 0.1, 2 FOR 0.01,... 5 FOR 0.00001
DA3D
DA3D CB7C                BIT 7,H
DA3F 2801                JR Z,NEGEFORM
DA41
DA41 2F                  CPL
DA42
DA42 2E2F        NEGEFORM: LD L,2FH        ;"0"-1
DA44
DA44 2C          CALCELP: INC L
DA45 D60A                SUB 10
DA47 30FB                JR NC,CALCELP
DA49
DA49 C63A                ADD A,3AH
DA4B F5                  PUSH AF
DA4C EB                  EX DE,HL         ;RESULT IN EA (POWER AS ASCII)
DA4D 09                  ADD HL,BC        ;HL=PRNBUFF, BC=LEN - GET PAST END
DA4E 3645                LD (HL),"E"
DA50 23                  INC HL
DA51 362B                LD (HL),"+"
DA53 CB7A                BIT 7,D
DA55 2002                JR NZ,POWSOK
DA57
DA57 362D                LD (HL),"-"
DA59
DA59 23          POWSOK: INC HL
DA5A 7B                  LD A,E
DA5B FE30                CP "0"
DA5D 2803                JR Z,POWL10
DA5F
DA5F 0C                  INC C            ;LEN IS GREATER BECAUSE "E"+/- 2 DIGITS
DA60 73                  LD (HL),E
DA61 23                  INC HL
DA62
DA62 F1          POWL10: POP AF
```

```
DA63 77                 LD (HL),A
DA64 0C                 INC C
DA65 0C                 INC C
DA66 0C                 INC C           ;ALLOW FOR E(SGN)1 DIGIT
DA67 21A05B             LD HL,PRNBUFF
DA6A 7E                 LD A,(HL)
DA6B FE2D               CP "-"
DA6D 2001               JR NZ,NOMINUS2
DA6F
DA6F 23                 INC HL          ;SKIP "-"
DA70
DA70 23         NOMINUS2:  INC HL       ;SKIP FIRST DIGIT
DA71 7E                 LD A,(HL)
DA72 FE45               CP "E"
DA74
DA74 282D       PFSPBH:    JR Z,PFSETPRB    ;END NOW IF E.G. 1E+7 - NO PT. INSERTION
DA76
DA76 0C                 INC C           ;ALLOW FOR GREATER LEN
DA77 3E01               LD A,1
DA79 CDA7DA             CALL PFSPACE    ;MAKE 1 SPACE AFTER ANY MINUS (CAN BE CONSIDERED
DA7C                                    ;AS SPACE 1 AFTER THIS, AS LOCNS MATCH)
DA7C 23                 INC HL          ;PT TO FIRST DIGIT
DA7D 23                 INC HL          ;PT TO SPACE AFTER 1ST DIGIT
DA7E 362E               LD (HL),"."
DA80 1821               JR PFSETPRB
DA82
DA82            ;PRINT FRACTION - INSERT LEADING 0./0.0/0.00 ETC.
DA82
DA82 3C         PFFRACT:   INC A
DA83 FE14               CP 20           ;LIMIT TO 17 LEADING ZEROS (FOR PRINT USING)
DA85 3802               JR C,PFFRACT2
DA87
DA87 3E13               LD A,19
DA89
DA89 47         PFFRACT2:  LD B,A       ;B=2-6, C=LEN
DA8A                                    ;2=0.1, 8=0.0000001
DA8A CDA7DA             CALL PFSPACE
DA8D 23                 INC HL          ;PT TO SPACE
DA8E 79                 LD A,C
DA8F 80                 ADD A,B
DA90 FE16               CP 22
DA92 3802               JR C,PFFRACT3
DA94
DA94 3E15               LD A,21
DA96
DA96 4F         PFFRACT3:  LD C,A       ;NEW LEN (MAX=21)
DA97 3630               LD (HL),"0"
DA99 3E2E               LD A,"."        ;DECIMAL PT. ON FIRST LOOP ONLY
DA9B 1803               JR FRACLPEN
DA9D
DA9D 77         FRACLZLP:  LD (HL),A
DA9E 3E30               LD A,"0"
DAA0
DAA0 23         FRACLPEN:  INC HL
DAA1 10FA               DJNZ FRACLZLP
DAA3
DAA3 11A05B     PFSETPRB:  LD DE,PRNBUFF    ;BC=LEN
DAA6 C9                 RET
DAA7
DAA7            ;ENTRY:    A=SPACE TO OPEN AFTER ANY MINUS SIGN IN PRNBUFF. (1-19D BYTES)
DAA7            ;EXIT: HL=BEFORE SPACE. BC SAVED
DAA7
DAA7 C5         PFSPACE:   PUSH BC
DAA8 26FF               LD H,0FFH
DAAA ED44               NEG
DAAC 6F                 LD L,A          ;HL=-SPACE
DAAD E5                 PUSH HL
DAAE 011500             LD BC,21        ;BUFF LEN (ACTUALLY, OVERLAPS BCD BUFF, BUT OK)
DAB1 09                 ADD HL,BC       ;MAX TO MOVE IS 15; LESS IF BIGGER GAP WANTED
DAB2                                    ;MIN IS 1
DAB2 44                 LD B,H
DAB3 4D                 LD C,L
DAB4 E1                 POP HL
DAB5 3AA05B             LD A,(PRNBUFF)
DAB8 FE2D               CP "-"
DABA 2001               JR NZ,PFSKMIN
DABC
DABC 0D                 DEC C           ;1 LESS IF LEADING MINUS - LEAVE IT ALONE
DABD
DABD 11B45B     PFSKMIN:   LD DE,PRNBUFF+20  ;DEST=END OF BUFFER
DAC0 19                 ADD HL,DE       ;SRC IS SOME BYTES BEFORE
DAC1 EDB8               LDDR
DAC3 C1                 POP BC
DAC4 C9                 RET
DAC5
DAC5            ;*************************************************************************
DAC5            ;PRFPBUF - "PRINT" A NUMBER (ON FPCS) TO PRNBUFF IN "RAW" STATE. (NO ROUNDING
DAC5            ;OR LEADING ZEROS, NO E+/-, ETC.)
DAC5            ;EXIT: (9-DIGITS)=SIGNIF DIGS IN PRNBUFF. EPOWER=POWER OF 10 USED IF EFORM
DAC5            ;NO DECIMAL PT IF EFORM.
DAC5            ;LEADING MINUS IF -VE, ALWAYS 1 LEADING ZERO.
DAC5            ;E.G. -0.123 OR 01234.5  OR 01.2345
DAC5
DAC5 219D5B     PRFPBUF:   LD HL,DIGITS
DAC8 360A               LD (HL),10      ;DIGITS ALLOWED
DACA 23                 INC HL
DACB AF                 XOR A
DACC 77                 LD (HL),A       ;POWER=0 "NOT EFORMAT"
DACD 23                 INC HL
DACE 77                 LD (HL),A       ;"NO DECIMAL PT USED YET"
DACF 23                 INC HL
DAD0 229B5B             LD (NPRPOS),HL  ;PRINT POSN=BUFFER START
DAD3
```

```
DAD3 EF                     DB CALC
DAD4 31                     DB RESTACK
DAD5 07                     DB DROP
DAD6 33                     DB EXIT             ;DE=DROPPED NUMBER
DAD7
DAD7 210500                 LD HL,5
DADA 19                     ADD HL,DE
DADB
DADB EB                     EX DE,HL            ;HL=NUMBER, DE=OLD STKEND
DADC 23                     INC HL
DADD CB7E                   BIT 7,(HL)          ;NZ IF -VE
DADF 2B                     DEC HL
DAE0
DAE0 3E2D                   LD A,"-"
DAE2 C473DB                 CALL NZ,NPRINT      ;MINUS PRINTED BEFORE -VE NUMBERS
DAE5 CD67DB                 CALL NPRZERO        ;LEADING ZERO IS USEFUL FOR ROUNDING UP LATER
DAE8 7E                     LD A,(HL)
DAE9 A7                     AND A
DAEA C8                     RET Z               ;EXIT WITH "0" IF ZERO
DAEB
DAEB FE81                   CP 81H
DAED 384D                   JR C,PRBORS         ;JR IF NO BITS BEFORE BINARY POINT (<1)
DAEF
DAEF D680                   SUB 80H             ;=BITS BEFORE BINARY POINT (1-7F)
DAF1 FE1E                   CP 30
DAF3 3047                   JR NC,PRBORS        ;IF MORE THAN 29 BITS - MIGHT NOT FIT IN 9 DIG. BCD
DAF5
DAF5 23           PRMEDN:   INC HL
DAF6 56                     LD D,(HL)
DAF7 CBFA                   SET 7,D             ;TRUE NUMERIC BIT
DAF9 23                     INC HL
DAFA 5E                     LD E,(HL)
DAFB 23                     INC HL
DAFC D5                     PUSH DE
DAFD D9                     EXX
DAFE E1                     POP HL              ;MSW TO HL"
DAFF D9                     EXX
DB00 56                     LD D,(HL)
DB01 23                     INC HL
DB02 5E                     LD E,(HL)
DB03 EB                     EX DE,HL            ;LSW TO HL
DB04 CD95DB                 CALL DECIMIZE       ;CONVERT "A" BITS OF HL"HL TO BCD
DB07 EB                     EX DE,HL            ;PROTECT LSW
DB08 CDB5DB                 CALL PRBCD          ;PRINT BCD TO PRINT BUFFER
DB0B EB                     EX DE,HL
DB0C D8                     RET C               ;RET IF BUFFER FULL
DB0D
DB0D 7C                     LD A,H
DB0E B5                     OR L
DB0F D9                     EXX
DB10 B4                     OR H
DB11 B5                     OR L
DB12 C8                     RET Z               ;RET IF NO SIGNIF BITS LEFT
DB13
DB13              ;BITS REMAINING IN HL"HL ARE THE BITS AFTER THE BINARY POINT, EXP 80H FORM
DB13              ;EXCEPT THERE MAY BE LEADING ZERO BITS.
DB13              ;NOW PRINT HL"HL BY MULTIPLYING REPEATEDLY BY 10 TILL BUFFER FULL
DB13
DB13
DB13 CD60DB       PRFRACT:  CALL NPRPNT         ;"."
DB16 EB                     EX DE,HL
DB17 D9                     EXX
DB18 EB                     EX DE,HL            ;TRANSFER NUMBER TO DE"DE
DB19
DB19 AF           PRFRLP:   XOR A               ;START WITH CARRY OF ZERO
DB1A CD2ADB                 CALL TENX           ;LE=DE*10+A
DB1D 55                     LD D,L              ;DE=DE*10. A=CY
DB1E D9                     EXX
DB1F CD2ADB                 CALL TENX           ;LE=DE*10+A
DB22 55                     LD D,L              ;DE=DE*10. A=CY
DB23 D9                     EXX
DB24 CD68DB                 CALL NPRINTC
DB27 30F0                   JR NC,PRFRLP        ;LOOP UNTIL BUFFER FULL
DB29
DB29 C9                     RET
DB2A
DB2A              ;LE=DE*10+A. CY IN A
DB2A
DB2A 4B           TENX      LD C,E
DB2B CD30DB                 CALL TENMUL
DB2E 5D                     LD E,L
DB2F 4A                     LD C,D
DB30
DB30              ;HL=C*10+A.
DB30
DB30 69           TENMUL:   LD L,C
DB31 2600                   LD H,0
DB33 44                     LD B,H
DB34 29                     ADD HL,HL
DB35 29                     ADD HL,HL
DB36 09                     ADD HL,BC           ;*5
DB37 29                     ADD HL,HL           ;*10
DB38 4F                     LD C,A
DB39 09                     ADD HL,BC
DB3A 7C                     LD A,H              ;A=CARRY FOR NEXT TIME, L=RESULT
DB3B C9                     RET
DB3C
DB3C              ;NUMBERS LESS THAN 1 HAVE EXPONENTS OF 1-80H. EXP 80H MEANS THAT THERE ARE
DB3C              ;NO ZERO BITS AFTER THE BINARY POINT, EXP 71H MEANS THERE ARE 15 BITS, ETC.
DB3C
DB3C ED53655C     PRBORS:   LD (STKEND),DE      ;PUT NUMBER BACK ON FPCS
DB40 3007                   JR NC,PRBIGN        ;JR IF BIG
DB42
```

```
DB42 CD7FDB               CALL DECDIGN      ;GET MAX LEADING ZEROS (DECIMAL). 0 OR MORE
DB45 C609                 ADD A,9           ;GET AN INTEGER OF 8 OR 9 DIGITS
DB47 1806                 JR SHIFTDEC
DB49
DB49
DB49            ;NUMBER WITH EXP 9E-FFH (NOW 1E-7FH) - AT LEAST 9 DEC DIGITS
DB49
DB49 CD83DB     PRBIGN:   CALL DECDIGP      ;FIND MAX DIGITS BEFORE DEC PT, -1. (9 OR MORE)
DB4C D609                 SUB 9             ;0 OR MORE.
DB4E 2F                   CPL               ;-1 OR LESS. GET AN INTEGER OF 8 OR 9 DIGITS
DB4F
DB4F 3D         SHIFTDEC: DEC A
DB50 329E5B               LD (EPOWER),A     ;..FE=E+9,FF=E+8. GAP=0. 9=E-1,10=E-2...
DB53 3C                   INC A
DB54 CDD4DB               CALL POFTEN       ;GET N1**A (SIGNED). EXIT WITH RESULT ON FPCS
DB57
DB57 CD121D               CALL FDELETE
DB5A 7E                   LD A,(HL)         ;A=EXP OF DELETED RESULT
DB5B D680                 SUB 80H
DB5D C3F5DA               JP PRMEDN         ;HL PTS TO DROPPED NUMBER (NOW LARGE INTEGER)
DB60
DB60 3E2E       NPRPNT:   LD A,"."
DB62 329F5B               LD (DECPNTED),A   ;SHOW NON-INTEGER
DB65 180C                 JR NPRINT
DB67
DB67 AF         NPRZERO:  XOR A
DB68
DB68            ;PRINT "A" AND DEC "DIGITS TO COME". CY IF BUFFER FULL (CHAR STILL ENTERED, THO)
DB68
DB68 C630       NPRINTC:  ADD A,30H
DB6A E5                   PUSH HL
DB6B 219D5B               LD HL,DIGITS
DB6E 35                   DEC (HL)
DB6F 2003                 JR NZ,NPRINT2     ;JR IF SPACE OK
DB71
DB71 37                   SCF               ;CY=BUFFER FULL
DB72 21                   DB 21H            ;"JR+2"
DB73
DB73            ;THIS ENTRY FOR CHARS THAT DON"T COUNT - LIKE "-" AND "."
DB73
DB73 E5         NPRINT:   PUSH HL
DB74
DB74 A7         NPRINT2:  AND A             ;NC
DB75
DB75 2A9B5B     NPRINT3:  LD HL,(NPRPOS)
DB78 77                   LD (HL),A
DB79 23                   INC HL
DB7A 229B5B               LD (NPRPOS),HL
DB7D E1                   POP HL
DB7E C9                   RET
DB7F
DB7F            ;FIND HOW MANY DECIMAL DIGITS-1 THERE ARE IN A POWER OF TWO (MAXIMUM)
DB7F            ;ENTRY WITH A=POWER OF 2 (-VE IF ENTRY AT DECDIGN)
DB7F            ;EXIT WITH A=DECIMAL DIGITS BEFORE POINT-1 IF DECDIGP OR ZEROS AFTER
DB7F            ;PT. IF DECDIGN
DB7F
DB7F 4F         DECDIGN:  LD C,A
DB80 3E80                 LD A,80H
DB82 91                   SUB C             ;1-80H->7F-00H
DB83
DB83 CDDA1C     DECDIGP:  CALL STACKA
DB86
DB86 EF                   DB CALC
DB87 27                   DB FIVELIT
DB88 7F1A20               DB 7FH,1AH,20H
DB8B 9A85                 DB 9AH,85H        ;0.30103
DB8D 00                   DB MULT
DB8E 30                   DB TRUNC
DB8F 07                   DB DROP
DB90 33                   DB EXIT
DB91
DB91 13                   INC DE
DB92 13                   INC DE
DB93 1A                   LD A,(DE)
DB94 C9                   RET
DB95
DB95            ;CONVERT A BITS IN HL"HL TO ASCI IN PRNBUFF. RESULT MAY HAVE UP TO 10 BCD DIGITS
DB95
DB95 4F         DECIMIZE: LD C,A
DB96
DB96 AF                   XOR A
DB97 0605                 LD B,5
DB99 11B05B               LD DE,BCDBUFF     ;ZERO 5-BYTE BCD BUFFER.
DB9C
DB9C 12         ZBCDBLP:  LD (DE),A
DB9D 13                   INC DE
DB9E 10FC                 DJNZ ZBCDBLP
DBA0
DBA0 29         DECIMCLP: ADD HL,HL
DBA1 D9                   EXX
DBA2 ED6A                 ADC HL,HL
DBA4 D9                   EXX               ;SHIFT LEFT BITS IN HL"HL, TO CY
DBA5
DBA5 11B45B               LD DE,BCDBUFF+4   ;PT TO RHS BYTE IN BCD BUFFER
DBA8 0605                 LD B,5
DBAA
DBAA 1A         DECIMBLP: LD A,(DE)
DBAB 8F                   ADC A,A           ;FOR FIRST BYTE, ADD IN CY FROM HL"HL SHIFT
DBAC                                        ;FOR LATER BYTES, ADD IN CY FROM PREVIOUS DAA
DBAC 27                   DAA
DBAD 12                   LD (DE),A
DBAE 1B                   DEC DE
DBAF 10F9                 DJNZ DECIMBLP
```

```
DBB1
DBB1 0D                 DEC C
DBB2 20EC               JR NZ,DECIMCLP    ;SHIFT C BITS
DBB4
DBB4 C9                 RET
DBB5
DBB5            ;PRINT CONTENTS OF 10-DIGIT BCD BUFFER TO PRINT BUFFER AS ASCII.
DBB5            ;EXITS IF ALL BCD DIGITS USED (NC), OR, IF PRINT BUFFER IS PART-FILLED ON ENTRY,
DBB5            ;WHEN DIGIT COUNT REACHES 00 (CY)
DBB5            ;USES HL,BC,AF
DBB5
DBB5 0E01       PRBCD:    LD C,1          ;NZ C SIGNALS "LEADING ZEROS - IGNORE"
DBB7 3E0F                 LD A,0FH        ;CLEAR LHS OF A, AND MARK RHS WITH NON-BCD DIGIT
DBB9
DBB9 0605       BCDDIG:   LD B,5          ;BCD ROTATE OF 5 BYTES
DBBB 21B45B               LD HL,BCDBUFF+4
DBBE
DBBE ED6F       BCDROTL:  RLD             ;RHS NIBBLE OF (HL) TO A, LHS NIBBLE TO RHS,
DBC0                                      ;RHS NIBBLE OF A TO RHS OF (HL)
DBC0 2B                   DEC HL
DBC1 10FB                 DJNZ BCDROTL
DBC3
DBC3 2003                 JR NZ,PRDIGIT     ;JR IF LAST RLD GAVE A REG A NZ DIGIT
DBC5
DBC5 B1                   OR C              ;NZ IF "LEADING ZERO" STATUS
DBC6 20F1                 JR NZ,BCDDIG      ;IGNORE ANY. ELSE A=0 FOR NON-LEADING ZERO
DBC8
DBC8 48         PRDIGIT:  LD C,B            ;C=0 - NO LEADING ZEROS NOW
DBC9 E60F                 AND 0FH
DBCB FE0A                 CP 0AH
DBCD D0                   RET NC            ;RET IF TERMINATOR HIT
DBCE
DBCE CD68DB               CALL NPRINTC
DBD1 30E6                 JR NC,BCDDIG       ;LOOP UNTIL DIGIT COUNT SHOWS FULL
DBD3
DBD3 C9                   RET
DBD4
DBD4            ;POWER-OF-TEN. MULT. TOP OF FPCS BY 10**"A" REG. WORKS WITH -VE POWERS (A>80H)
DBD4
DBD4 4F         POFTEN:   LD C,A            ;BIT 7, C=1 IF -VE POWER
DBD5 CB79                 BIT 7,C
DBD7 2802                 JR Z,POF10L1
DBD9
DBD9 ED44                 NEG
DBDB
DBDB 47         POF10L1:  LD B,A            ;B=ABSOLUTE POWER (CORRUPTS!) BC IS SAVED BY SAM
DBDC                                        ;FPCS
DBDC EF                   DB CALC
DBDD EC                   DB STKTEN         ;N1,10 (CURRENT POWER. CALL IT P)
DBDE 33                   DB EXIT
DBDF
DBDF 1804                 JR POF10LPE       ;JUMP INTO THE LOOP
DBE1
DBE1 EF         POF10LP:  DB CALC
DBE2 25                   DB DUP            ;N1*P,P,P
DBE3 00                   DB MULT           ;N1*P,P*P. DOUBLE POWER (WE ARE MOVING MORE SIGNIF
DBE4                                        ;BITS RIGHT IN B)
DBE4 33                   DB EXIT
DBE5
DBE5 CB38       POF10LPE: SRL B
DBE7 3010                 JR NC,POF10L3     ;JR IF NO MULT/DIV ON THIS BIT
DBE9
DBE9 CB79                 BIT 7,C
DBEB 2007                 JR NZ,POF10L2     ;JR IF -VE POWER - USE DIV
DBED
DBED EF                   DB CALC
DBEE D0                   DB STO0           ;N1,P
DBEF 00                   DB MULT           ;N1*P
DBF0 D8                   DB RCL0           ;N1*P,P
DBF1 33                   DB EXIT
DBF2
DBF2 1805                 JR POF10L3
DBF4
DBF4 EF         POF10L2:  DB CALC
DBF5 D0                   DB STO0           ;N1,P
DBF6 05                   DB DIVN           ;N1/P
DBF7 D8                   DB RCL0           ;N1/P,P
DBF8 33                   DB EXIT
DBF9
DBF9 04         POF10L3:  INC B
DBFA 10E5                 DJNZ POF10LP      ;JUMP IF B NOT SHIFTED TO ZERO YET
DBFC
DBFC 22655C               LD (STKEND),HL    ;HL PTS TO LAST VALUE (HL IS SET BY "EXIT")
DBFF C9                   RET               ;DELETE LAST POWER, LEAVE N1**A ON FPCS
DC00
DC00
DC00                      INCLUDE TPRINT.SAM
DC00            ;TPRINT.SAM - PRINT ASCII AND CR.
DC00
DC00 FE20       PROM1:    CP 20H
DC02 DAC4DD               JP C,PRCRLCDS
DC05
DC05 FE80                 CP 80H
DC07 D2ABDC               JP NC,PRGR80
DC0A
DC0A ED5B365C   PRASCII:  LD DE,(CHARS)
DC0E 213B5C               LD HL,FLAGS
DC11 CBC6                 SET 0,(HL)        ;'SPACE WAS LAST CHAR'
DC13 FE20                 CP 20H
DC15 2802                 JR Z,PRINTMN1
DC17
DC17 CB86                 RES 0,(HL)        ;'LAST CHAR WAS NOT SPACE'
DC19
```

```
DC19 6F          PRINTMN1:  LD L,A
DC1A 2600                   LD H,0
DC1C 29                     ADD HL,HL
DC1D 29                     ADD HL,HL
DC1E 29                     ADD HL,HL
DC1F 19                     ADD HL,DE          ;HL PTS TO CHAR DATA
DC20
DC20 32725A      IOPENT:    LD (OPCHAR),A      ;USED BY LPRINT
DC23 47                     LD B,A
DC24
DC24             ;FROM POUDGS, AND AFTER CR
DC24
DC24 CD893D      NLENTRY:   CALL POFETCH       ;GET DE=COL/ROW, A=RHS LIMIT
DC27 BB                     CP E               ;CURRENT COL
DC28 3022                   JR NC,PRNONWLN     ;JR IF COL=LESS THAN OR EQUAL TO RHS LIMIT
DC2A
DC2A 1D                     DEC E
DC2B 93                     SUB E              ;Z IF JUST 1 PAST RHS LIMIT - LINE FULL; ELSE CR
DC2C 4F                     LD C,A             ;WAS USED JUST BEFORE
DC2D C5                     PUSH BC
DC2E 1EFF                   LD E,0FFH          ;COL FF SHOWS 'RECURSIVE' CR
DC30 3A735A                 LD A,(DEVICE)
DC33 C6FE                   ADD A,0FEH         ;CY IF PRINTER
DC35 E5                     PUSH HL
DC36 CDA2DE                 CALL PRENTER
DC39 E1                     POP HL             ;GET CHAR PATTERN BACK
DC3A C1                     POP BC             ;B=CHAR CODE
DC3B 3ABD5A                 LD A,(INDOPFG)
DC3E A7                     AND A
DC3F 28E3                   JR Z,NLENTRY       ;JR IF NO INDENT O/P WANTED - NOT LISTING.
DC41
DC41 79                     LD A,C
DC42 A7                     AND A
DC43 20DF                   JR NZ,NLENTRY
DC45
DC45 C5                     PUSH BC
DC46 CD6AF4                 CALL INDOPEN         ;O/P SPACES IF POSN WAS JUST 1 PAST RHS LIMIT -
DC49 F1                     POP AF               ;LINE WAS FULL. ELSE CR WAS USED.
DC4A D7                     RST 10H
DC4B C9                     RET
DC4C
DC4C D5          PRNONWLN:  PUSH DE            ;CURRENT PRINT POSN
DC4D 3A535A                 LD A,(OVERT)       ;0-3
DC50 FE01                   CP 1               ;0->CY
DC52 9F                     SBC A,A            ;0->FF
DC53 2F                     CPL                ;0->00, 1,2,3->FF
DC54 47                     LD B,A
DC55 3A545A                 LD A,(INVERT)      ;0=INVERSE 0, FF=INVERSE 1
DC58 4F                     LD C,A
DC59 DD2AD25B               LD IX,(PATOUT)     ;USUALLY=ENDOUTP
DC5D CD2D00                 CALL IXJUMP
DC60 E1                     POP HL             ;POSN
DC61 2C                     INC L              ;MOVE RIGHT
DC62
DC62 3A735A      POSTORE:   LD A,(DEVICE)      ;0=UPPER SCREEN, 1=LOWER, 2=PRINTER OR OTHER
DC65 A7                     AND A
DC66 280B                   JR Z,POSUSCRN
DC68
DC68 3D                     DEC A
DC69 2804                   JR Z,POSLSCRN
DC6B
DC6B 22705A                 LD (PRPOSN),HL     ;L=PRINTER COL
DC6E C9                     RET
DC6F
DC6F 226E5A      POSLSCRN:  LD (SPOSNL),HL
DC72 C9                     RET
DC73
DC73 226C5A      POSUSCRN:  LD (SPOSNU),HL
DC76 C9                     RET
DC77
DC77 3AB75A      ENDOUTP:   LD A,(DMPFG)
DC7A A7                     AND A
DC7B C0                     RET NZ
DC7C
DC7C 3A735A                 LD A,(DEVICE)
DC7F FE02                   CP 2
DC81 2822                   JR Z,ENDOP2        ;JR IF PRINTER
DC83
DC83 3A365A                 LD A,(CSIZE)
DC86 FE10                   CP 16
DC88 DA393C                 JP C,EPSUB         ;JR IF NOT DOUBLE HEIGHT
DC8B
DC8B CDEDDF                 CALL DBCHAR        ;EXITS WITH HL PTING TO DOUBLE HEIGHT CHAR MATRIX
DC8E C5                     PUSH BC
DC8F D5                     PUSH DE
DC90 CD393C                 CALL EPSUB         ;PRINT TOP HALF
DC93 3E08                   LD A,8
DC95 32825B                 LD (DHADJ),A       ;ADJUST ADDRESSES TO PRINT 8 SCANS LOWER
DC98 D1                     POP DE
DC99 C1                     POP BC
DC9A 213151                 LD HL,MEMVAL+16    ;PTR TO BOTTOM HALF
DC9D CD393C                 CALL EPSUB
DCA0 AF                     XOR A
DCA1 32825B                 LD (DHADJ),A
DCA4 C9                     RET
DCA5
DCA5 3A725A      ENDOP2:    LD A,(OPCHAR)
DCA8 C3D1DE                 JP CHBOP           ;SEND CHAR ON 'B' CHANNEL
DCAB
DCAB             ;*********************************************************************************
DCAB
DCAB 4F          PRGR80:    LD C,A
DCAC 3ABA5A                 LD A,(INQUFG)      ;BIT 0=1 IF IN QUOTES
```

```
DCAF 0F                   RRCA
DCB0 67                   LD H,A
DCB1 3A715C               LD A,(FLAGX)      ;BIT 7=1 IF INPUT LINE
DCB4 B4                   OR H
DCB5 17                   RLA
DCB6 3803                 JR C,POUDGH       ;JR IF IN QUOTES, OR INPUT LINE - PRINT UDGS,
DCB8                                        ;NOT TOKENS
DCB8 79                   LD A,C
DCB9 FE85                 CP 85H
DCBB
DCBB DA7DDD     POUDGH:   JP C,POUDG        ;80-84H ARE ALWAYS UDGS
DCBE
DCBE            ;USED BY CHANNEL 'R'
DCBE
DCBE 2ADE5A     PRGR802:  LD HL,(PRTOKV)
DCC1 CD10F2               CALL JPOPT
DCC4
DCC4 3C                   INC A
DCC5 CAE8DC               JP Z,POFN         ;JP IF FF FN LEADER
DCC8
DCC8 D686                 SUB 86H           ;(PLUS 1 TO COMPENSATE FOR INC)
DCCA 11BCF9               LD DE,KWDS85      ;SPLIT CMDS LIST INTO 4 SUB-LISTS FOR SPEED
DCCD FE1B                 CP 1BH
DCCF 3857                 JR C,POBTL        ;85-9FH
DCD1
DCD1 D61B                 SUB 1BH
DCD3 2620                 LD H,20H
DCD5 112FFA               LD DE,KWDSA0      ;A0-BF
DCD8 BC                   CP H
DCD9 384D                 JR C,POBTL
DCDB
DCDB 94                   SUB H
DCDC 11CBFA               LD DE,KWDSC0      ;C0-DF
DCDF BC                   CP H
DCE0 3846                 JR C,POBTL
DCE2
DCE2 94                   SUB H
DCE3 1167FB               LD DE,KWDSE0      ;E0-FE
DCE6 1840                 JR POBTL
DCE8
DCE8            ;*************************************************************************
DCE8            ;ROUTINES FOR PRINTING ERROR MSGS, CMDS, FNS, MSGS
DCE8
DCE8 11AD3D     POFN:     LD DE,POSTFF
DCEB C3EBDE               JP SVSETOP        ;SAVE CURRENT O/P ADDR AND SET O/P ADDR TO DE
DCEE
DCEE CD7204     PSTFF2:   CALL RESTOP       ;RESTORE O/P ADDR. A= CHAR PRINTED AFTER FF
DCF1 D63B                 SUB PITOK
DCF3 FE18                 CP SINTOK-PITOK
DCF5 380D                 JR C,POIMFN       ;JR IF IMMEDIATE FN
DCF7
DCF7 D63F                 SUB MODTOK-PITOK
DCF9 3816                 JR C,POFPCFN      ;JR IF FPC FN
DCFB
DCFB 11A3F9               LD DE,BINFNTL     ;BINARY FN TOKEN LIST
DCFE FE07                 CP ANDTOK-MODTOK+1
DD00 3826                 JR C,POBTL        ;MOD-AND (BIN OPS) HAVE LEADING AND TRAILING SPACE
DD02
DD02 1822                 JR POMSP2         ;<>, >=, <= HAVE NO SPACES
DD04
DD04
DD04 11C9F8     POIMFN:   LD DE,IMFNTL      ;IMMEDIATE FN TOKEN LIST
DD07 FE07                 CP FNTOK-PITOK
DD09 280B                 JR Z,POTRNL
DD0B
DD0B FE08                 CP BINTOK-PITOK
DD0D 2807                 JR Z,POTRNL       ;FN AND BIN HAVE TRAILING BUT NO LEADING SPACES
DD0F
DD0F 1815                 JR POMSP2         ;REST HAVE NEITHER
DD11
DD11 C627       POFPCFN:  ADD A,MODTOK-SINTOK
DD13 112AF9               LD DE,FPCFNTL     ;FPC FN TOKEN LIST
DD16
DD16 A7         POTRNL:   AND A
DD17 F5                   PUSH AF           ;TRAILING SPACE
DD18 37                   SCF               ;NO LEADING SPACE
DD19 180F                 JR POGEN1
DD1B
DD1B 37         POMSPX:   SCF
DD1C F5                   PUSH AF
DD1D F5                   PUSH AF
DD1E 21CB50               LD HL,MSGBUFF+11
DD21 CD4FDD               CALL POMSR2
DD24 1808                 JR POGEN2
DD26
DD26 37         POMSP2:   SCF               ;NO LEADING OR TRAILING SPACES
DD27 26                   DB 26H            ;"JR+1". FOR MSGS OR E.G. PI/ITEM/POINT
DD28
DD28 A7         POBTL:    AND A             ;BOTH TRAILING AND LEADING SPACES
DD29
DD29 F5         POGEN:    PUSH AF           ;TRAILING STATUS
DD2A
DD2A F5         POGEN1:   PUSH AF           ;LEADING  STATUS
DD2B CD4CDD               CALL POMSR        ;GET MSG TO BUFFER
DD2E
DD2E F1         POGEN2:   POP AF            ;LEADING STATUS
DD2F 3809                 JR C,POMSG3       ;NO LEADING SPACE FOR MSG OR FNS (EXCEPT AND/OR)
DD31
DD31 3A3B5C               LD A,(FLAGS)
DD34 1F                   RRA
DD35 3E20                 LD A," "
DD37 D41000               CALL NC,0010H     ;PRINT LEADING SPACE IF PREVIOUS CHAR WASN'T ONE
DD3A
```

```
DD3A CD1300     POMSG3:     CALL PRINTSTR     ;OUTPUT BC FROM (DE)
DD3D F1                     POP AF
DD3E D8                     RET C             ;RET IF NO TRAILING SPACE WANTED
DD3F
DD3F 1B                     DEC DE
DD40 1A                     LD A,(DE)         ;LAST CHAR
DD41 FE41                   CP "A"
DD43 3003                   JR NC,POMSG4      ;JR IF LETTER - DO TRAILING SPACE
DD45
DD45 FE24                   CP "$"
DD47 C0                     RET NZ            ;'$' IS ALSO FOLLOWED BY A SPACE, BUT NOT =/>/#
DD48
DD48 3E20       POMSG4:     LD A," "
DD4A D7                     RST 10H
DD4B C9                     RET
DD4C
DD4C            ;A=MSG NUMBER 00->, DE=START OF MSG LIST
DD4C
DD4C 21C050     POMSR:      LD HL,MSGBUFF
DD4F
DD4F E5         POMSR2:     PUSH HL           ;BUFFER START
DD50 CD5ADD                 CALL POMSR3
DD53 D1                     POP DE            ;BUFFER ST
DD54 A7                     AND A
DD55 ED52                   SBC HL,DE
DD57 44                     LD B,H
DD58 4D                     LD C,L            ;LEN
DD59 C9                     RET
DD5A
DD5A 47         POMSR3:     LD B,A
DD5B 04                     INC B
DD5C 1805                   JR POMSR4
DD5E
DD5E 1A         LKHIBTLP:   LD A,(DE)         ;WITH ABOUT 4 CHARS/TOKEN, TAKES ABOUT 0.7 MS
DD5F 13                     INC DE            ;(ON SAM) TO FIND 30TH WORD.
DD60 17                     RLA
DD61 30FB                   JR NC,LKHIBTLP
DD63
DD63 10F9       POMSR4:     DJNZ LKHIBTLP
DD65
DD65 1A         MVWORDLP:   LD A,(DE)
DD66 E67F                   AND 7FH
DD68 FE20                   CP 20H
DD6A 3009                   JR NC,MVWORD2     ;JR WITH ALL EXCEPT COMPRESSION CODES 00-1F
DD6C
DD6C D5                     PUSH DE           ;SAVE MAIN MSG PTR
DD6D 1119F8                 LD DE,COMPLIST    ;LIST OF COMPRESSION CODED WORDS
DD70 CD5ADD                 CALL POMSR3       ;RECURSIVE CALL
DD73 D1                     POP DE            ;MAIN MSG PTR
DD74 01                     DB 01H            ;'JR+2'
DD75
DD75 77         MVWORD2:    LD (HL),A
DD76 23                     INC HL
DD77
DD77 1A         MVWORD3:    LD A,(DE)
DD78 13                     INC DE
DD79 17                     RLA
DD7A 30E9                   JR NC,MVWORDLP    ;MAIN MSG CAN HAVE COMP CODES AND ASCII. BIT 7 IS
DD7C                                          ;HI IN LAST CHAR. COMPRESSED WORD CANNOT HAVE COMP
DD7C                                          ;CODES.
DD7C C9                     RET
DD7D
DD7D
DD7D            ;*************************************************************************
DD7D
DD7D 3A735A     POUDG:      LD A,(DEVICE)
DD80 FE02                   CP 2
DD82 200A                   JR NZ,PUDGS       ;JR IF NOT PRINTER
DD84
DD84 79                     LD A,C
DD85 32725A                 LD (OPCHAR),A     ;USED BY LPRINT
DD88 2AF85A                 LD HL,(LPRTV)
DD8B CD10F2                 CALL JPOPT
DD8E
DD8E 3A345A     PUDGS:      LD A,(BGFLG)      ;BLOCK GRAPHICS FLAG
DD91 A7                     AND A
DD92 79                     LD A,C
DD93 201C                   JR NZ,POFUDG      ;JR IF FOREIGN SET/UDGS WANTED, NOT BLOCKS
DD95
DD95 FE90                   CP 90H
DD97 3018                   JR NC,POFUDG      ;>=90H ARE FOREIGN/UDGS ANYWAY
DD99
DD99 CD3DD8                 CALL QUADBITS     ;GET LOWER 4 BITS OF A INTO DE, QUADRUPLED
DD9C 212151                 LD HL,MEMVAL
DD9F E5                     PUSH HL
DDA0 0604                   LD B,4
DDA2
DDA2 73         BKGRL1:     LD (HL),E
DDA3 23                     INC HL
DDA4 10FC                   DJNZ BKGRL1
DDA6
DDA6 0604                   LD B,4
DDA8
DDA8 72         BKGRL2:     LD (HL),D
DDA9 23                     INC HL
DDAA 10FC                   DJNZ BKGRL2
DDAC
DDAC E1                     POP HL
DDAD 41                     LD B,C
DDAE C324DC                 JP NLENTRY
DDB1
DDB1 2A7D5C     POFUDG:     LD HL,(HUDG)      ;HI UDG FOR CHARS. >A8H
DDB4 D6A9                   SUB 0A9H
```

```
DDB6 3008                JR NC,POUDG1      ;JR IF HI UDG
DDB8
DDB8 79                  LD A,C
DDB9 1180FB              LD DE,0FB80H      ;COMP. FOR A BEING 80H-A8H AND UDG VAR PTING
DDBC 2A7B5C              LD HL,(UDG)       ;TO CHR$ 144
DDBF 19                  ADD HL,DE
DDC0
DDC0 EB          POUDG1: EX DE,HL
DDC1 C319DC              JP PRINTMN1
DDC4
DDC4             ;PRINT CONTROL CODES
DDC4             ;*************************************************************************
DDC4             ;ENTRY: JR FROM MAIN PRINT ROUTINE. A=00-1FH
DDC4
DDC4 FE18        PRCRLCDS: CP 24
DDC6 3053                JR NC,PRQUERY
DDC8
DDC8 FE06                CP 6
DDCA 384F                JR C,PRQUERY
DDCC
DDCC 5F                  LD E,A
DDCD 1600                LD D,0
DDCF 21D4DD              LD HL,CCPTB-6       ;CONTROL CODE PTR TABLE
DDD2 19                  ADD HL,DE
DDD3 5E                  LD E,(HL)
DDD4 19                  ADD HL,DE
DDD5 4F                  LD C,A
DDD6 CD893D              CALL POFETCH      ;D=ROW, E=COL, A=RHS MAX, CY IF LPRINTING
DDD9 E9                  JP (HL)
DDDA
DDDA             CCPTB:
DDDA
DDDA             ;         DB PRQUERY-CCPT       ;0
DDDA             ;         DB PRQUERY-CCPT-1     ;1
DDDA             ;         DB PRQUERY-CCPT-2     ;2
DDDA             ;         DB PRQUERY-CCPT-3     ;3  CLS?
DDDA             ;         DB PRQUERY-CCPT-4     ;4
DDDA             ;         DB PRQUERY-CCPT-5     ;5
DDDA 12                  DB PRCOMMA-CCPTB      ;6  PRINT COMMA
DDDB 40                  DB PRQUERY-CCPTB-1    ;7  (EDIT)
DDDC 43                  DB CURLF-CCPTB-2      ;8  CURSOR LEFT
DDDD 64                  DB CURRT-CCPTB-3      ;9  CURSOR RIGHT
DDDE BA                  DB CURDN-CCPTB-4      ;10 CURSOR DOWN
DDDF 58                  DB CURUP-CCPTB-5      ;11 CURSOR UP
DDE0 70                  DB PRDELL-CCPTB-6     ;12 DELETE LEFT
DDE1 C1                  DB PRENTER-CCPTB-7    ;13 ENTER
DDE2 92                  DB PRDELR-CCPTB-8     ;14 DELETE RIGHT
DDE3 38                  DB PRQUERY-CCPTB-9    ;15
DDE4 F7                  DB CC1OP-CCPTB-10     ;16 INK
DDE5 F6                  DB CC1OP-CCPTB-11     ;17 PAPER
DDE6 F5                  DB CC1OP-CCPTB-12     ;18 FLASH
DDE7 F4                  DB CC1OP-CCPTB-13     ;19 BRIGHT
DDE8 F3                  DB CC1OP-CCPTB-14     ;20 INVERSE
DDE9 F2                  DB CC1OP-CCPTB-15     ;21 OVER
DDEA F6                  DB CC2OPS-CCPTB-16    ;22 AT
DDEB F5                  DB CC2OPS-CCPTB-17    ;23 TAB
DDEC
DDEC             ;*************************************************************************
DDEC             ;D=LINE, E=COL
DDEC
DDEC 7B          PRCOMMA: LD A,E            ;COL
DDED 21575A              LD HL,WINDLHS
DDF0 96                  SUB (HL)          ;A=DIST FROM WIND LHS
DDF1 F5                  PUSH AF
DDF2 3A2F5A              LD A,(TABVAR)
DDF5 A7                  AND A
DDF6 0110F0              LD BC,0F010H      ;FOR 16-COLUMN TAB
DDF9 2803                JR Z,PCOM2
DDFB
DDFB 0108F8              LD BC,0F808H      ;FOR 8-COLUMN TAB
DDFE
DDFE 3A565A      PCOM2:  LD A,(WINDRHS)
DE01 BB                  CP E
DE02 3813                JR C,PC25         ;JR IF LINE FULL
DE04
DE04 F1                  POP AF
DE05 A0                  AND B             ;GET DIST DIV 16 OR DIV 8
DE06 81                  ADD A,C           ;ADD 16 OR 8
DE07 86                  ADD A,(HL)        ;ABS DESIRED COLUMN
DE08 2B                  DEC HL            ;PT TO RHS
DE09 BE                  CP (HL)
DE0A 3802                JR C,PCOM3        ;JR IF NEW POSN WOULD NOT REACH RHS MAX
DE0C
DE0C 7E                  LD A,(HL)
DE0D 3C                  INC A
DE0E
DE0E 93          PCOM3:  SUB E
DE0F
DE0F C8          OPAORZ: RET Z
DE10
DE10 47          OPASPACES: LD B,A
DE11
DE11 3E20        OPSPLP: LD A," "
DE13 D7                  RST 10H
DE14 10FB                DJNZ OPSPLP
DE16
DE16 C9                  RET
DE17
DE17 F1          PC25:   POP AF
DE18 41                  LD B,C
DE19 18F6                JR OPSPLP
DE1B
DE1B
```

```
DE1B 3E3F       PRQUERY:    LD A,"?"
DE1D D7                     RST 10H
DE1E C9                     RET
DE1F
DE1F            ;***************************************************************************
DE1F            ;CURSOR LEFT/UP/RIGHT
DE1F
DE1F 3A575A     CURLF:      LD A,(WINDLHS)
DE22 BB                     CP E
DE23 3A565A                 LD A,(WINDRHS)
DE26 2809                   JR Z,CURLF2         ;JR IF AT LHS OF WINDOW
DE28
DE28 1D                     DEC E               ;DEC COL.
DE29 BB                     CP E
DE2A 3001                   JR NC,POSTOREH      ;JR IF NOT PAST RHS ('LINE FULL')
DE2C
DE2C 5F                     LD E,A              ;ELSE COL=RHS
DE2D
DE2D EB         POSTOREH:   EX DE,HL
DE2E C362DC                 JP POSTORE
DE31
DE31
DE31 5F         CURLF2:     LD E,A              ;COL=RHS
DE32 3A735A                 LD A,(DEVICE)
DE35 C6FE                   ADD A,0FEH          ;CY IF PRINTER (DEVICE 2)
DE37
DE37 38E2       CURUP:      JR C,PRQUERY        ;PRINT '?' IF LPRINTING - POSN UNCHANGED
DE39
DE39 3A585A                 LD A,(WINDTOP)
DE3C BA                     CP D
DE3D C8                     RET Z               ;RET IF WE ARE AT TOP OF WINDOW
DE3E
DE3E 15                     DEC D               ;UP A ROW. C=RHS+1
DE3F 18EC                   JR POSTOREH
DE41
DE41
DE41 2A535A     CURRT:      LD HL,(OVERT)       ;OVERT/INVERT
DE44 E5                     PUSH HL
DE45 3A515A                 LD A,(M23PAPT)
DE48 F5                     PUSH AF
DE49 210100                 LD HL,0001H         ;OVER 1;INVERSE 0
DE4C 0601                   LD B,1
DE4E 1837                   JR SPOX
DE50
DE50            ;DELETE LEFT
DE50
DE50 38C9       PRDELL:     JR C,PRQUERY        ;JR IF LPRINTING
DE52
DE52 3E08                   LD A,8
DE54 D7                     RST 10H             ;BACKSPACE
DE55 CD74DE                 CALL SPO0           ;PRINT OVER 0;' ';
DE58 3E08                   LD A,8              ;BACKSPACE
DE5A D7                     RST 10H
DE5B C9                     RET
DE5C
DE5C            ;ERASE OLD CURSOR, GET H=TABLE MSB, GET C=LNPTR
DE5C
DE5C 3A585A     EROC2:      LD A,(WINDTOP)      ;ALLOW FOR WINDOW
DE5F 4F                     LD C,A
DE60 3A8D5B                 LD A,(LNPTR)
DE63 91                     SUB C
DE64 3001                   JR NC,EROC3
DE66
DE66 AF                     XOR A
DE67
DE67 4F         EROC3:      LD C,A
DE68 1E05                   LD E,5
DE6A CD1A08                 CALL ATSR2          ;PRINT AT A,5;
DE6D CD74DE                 CALL SPO0           ;ERASE OLD CURSOR
DE70 79                     LD A,C
DE71 265B                   LD H,LPT/256
DE73 C9                     RET
DE74
DE74            ;DELETE RIGHT=SPACE (OVER 0)
DE74
DE74            PRDELR:
DE74
DE74            ;PRINT OVER 0;INVERSE 0;" "; (IN PERMANENT PAPER IF MODES 2 AND 3)
DE74
DE74 0601       SPO0:       LD B,1              ;1 SPACE
DE76
DE76            ;O/P B SPACES (OVER 0;INVERSE 0)
DE76
DE76 2A535A     OPBSP:      LD HL,(OVERT)       ;OVERT/INVERT
DE79 E5                     PUSH HL
DE7A 3A515A                 LD A,(M23PAPT)
DE7D F5                     PUSH AF
DE7E 3A485A                 LD A,(M23PAPP)
DE81 32515A                 LD (M23PAPT),A      ;PERM PAPER
DE84 210000                 LD HL,0000H         ;OVER 0, INVERSE 0
DE87
DE87 22535A     SPOX:       LD (OVERT),HL
DE8A
DE8A 3E20       OPSL:       LD A,20H
DE8C D7                     RST 10H             ;PRINT SPACE
DE8D 10FB                   DJNZ OPSL
DE8F
DE8F F1                     POP AF
DE90 32515A                 LD (M23PAPT),A
DE93 E1                     POP HL
DE94 22535A                 LD (OVERT),HL
DE97 C9                     RET
DE98
```

```
DE98 D5          CURDN:     PUSH DE            ;SAVE COL.
DE99 1EFE                   LD E,0FEH          ;FORCE ACTUAL MOVE DOWN
DE9B CDA2DE                 CALL PRENTER       ;EXIT WITH HL=POSN
DE9E D1                     POP DE
DE9F 6B                     LD L,E             ;UNCHANGED COL.
DEA0 181E                   JR PRENT5
DEA2
DEA2 381F        PRENTER:   JR C,LPRENT        ;JR IF LPRINT
DEA4
DEA4 7B                     LD A,E             ;COL
DEA5 FEFE                   CP 0FEH
DEA7 3813                   JR C,PRENT3        ;IF PREVIOUS CHAR WAS NOT CR, JUST SET COL 0FEH.
DEA9                                           ;('LINE FULL') ELSE DROP DOWN A LINE
DEA9 3A595A                 LD A,(WINDBOT)
DEAC BA                     CP D
DEAD 2006                   JR NZ,PRENT2       ;JR IF NOT ON BOTTOM LINE ALREADY
DEAF
DEAF D5                     PUSH DE
DEB0 CD51DF                 CALL SCRLSCR       ;SCROLL WINDOW UP
DEB3 D1                     POP DE
DEB4 15                     DEC D              ;LINE VALUE WILL END AS BOTTOM LINE STILL
DEB5
DEB5 14          PRENT2:    INC D              ;DOWN A LINE
DEB6 1C                     INC E              ;Z IF COL WAS FF (RECURSIVE)
DEB7 3A575A                 LD A,(WINDLHS)
DEBA 2802                   JR Z,PRENT4        ;JR IF WAS RECURSIVE CR FROM 'LINE FULL' - LHS COL
DEBC                                           ;FOR CHAR COMING NEXT
DEBC
DEBC 3EFE        PRENT3:    LD A,0FEH          ;'LINE FULL, NOT RECURSIVE'
DEBE
DEBE 5F          PRENT4:    LD E,A
DEBF EB                     EX DE,HL
DEC0
DEC0 C362DC      PRENT5:    JP POSTORE
DEC3
DEC3 3E0D        LPRENT:    LD A,0DH
DEC5 CDD1DE                 CALL CHBOP         ;O/P CR ON CHANNEL 'B'
DEC8 AF                     XOR A
DEC9 32705A                 LD (PRPOSN),A
DECC 3A0F5A                 LD A,(AFTERCR)
DECF A7                     AND A
DED0 C8                     RET Z
DED1                                           ;ELSE LPRINT LF IF (AFTERCR)= 0AH
DED1 2A4F5C      CHBOP:     LD HL,(CHANS)
DED4 111900                 LD DE,25
DED7 19                     ADD HL,DE          ;PT TO CHANNEL B
DED8 C3C701                 JP HLJPI
DEDB             ;           LD E,(HL)
DEDB              ;          INC HL
DEDB               ;         LD D,(HL)
DEDB                ;        EX DE,HL          ;HL=O/P ADDR (USUALLY 'SENDA')
DEDB                 ;       JP (HL)
DEDB
DEDB             ;CONTROL CODES WITH OPERANDS
DEDB             ;ONE OPERAND - INK, PAPER, OVER, INVERSE, BRIGHT, FLASH
DEDB
DEDB 11A73D      CC1OP:     LD DE,CCRESTOP
DEDE 1803                   JR SETADCOM
DEE0
DEE0             ;TWO OPERANDS - TAB OR 'AT' (TAB JUNKS THE SECOND OPERAND!)
DEE0
DEE0 117D04      CC2OPS:    LD DE,PRERESTOP
DEE3
DEE3 CDEBDE      SETADCOM:  CALL SVSETOP
DEE6 79                     LD A,C             ;CONTROL CODE
DEE7 32BE5B                 LD (TVDATA),A
DEEA C9                     RET
DEEB
DEEB             ;SAVE CURRENT O/P ADDR AND SET IT TO DE
DEEB
DEEB D5          SVSETOP:   PUSH DE
DEEC CD46DF                 CALL SVCUROP
DEEF D1                     POP DE
DEF0 C37604                 JP POCHNG
DEF3
DEF3 2ABE5B      CCRP2:     LD HL,(TVDATA)     ;H=1ST OPERAND (IF 2),L=CONTROL CODE
DEF6
DEF6 57                     LD D,A             ;LAST OPERAND (ONLY OPERAND IF 1, ELSE 2ND)
DEF7 7D                     LD A,L             ;CONTROL CODE
DEF8 FE16                   CP 16H
DEFA DA8113                 JP C,PRCOITEM
DEFD
DEFD 2817                   JR Z,POATCC        ;JR IF 'AT'
DEFF
DEFF             ;DEAL WITH TAB. H=ONLY RELEVANT OPERAND (FIRST - COLUMN)
DEFF
DEFF CD893D                 CALL POFETCH       ;D=ROW, E=COL, A=RHS MAX, CY IF LPRINTING
DF02 4F                     LD C,A
DF03 BB                     CP E
DF04 7C                     LD A,H             ;PARAM
DF05 380B                   JR C,TAB2          ;JR IF LINE FULL
DF07
DF07 3A575A                 LD A,(WINDLHS)
DF0A 84                     ADD A,H            ;ABS TAB POSN
DF0B 93                     SUB E              ;FIND SPACES REQUIRED TO REACH TAB POSN.
DF0C C8                     RET Z
DF0D
DF0D 3003                   JR NC,TAB2         ;JR IF NOT ALREADY PAST TAB POSN
DF0F
DF0F 7C                     LD A,H             ;CALCULATE SPACES NEEDED TO FILL LINE
DF10 89                     ADC A,C            ;AND REACH TAB POSN.
DF11 93                     SUB E
DF12
```

```
DF12 A7          TAB2:      AND A
DF13 C30FDE                 JP OPAORZ
DF16
DF16             ;*************************************************************************
DF16             ;'AT' ROUTINE
DF16
DF16 EB          POATCC:    EX DE,HL
DF17 5C                     LD E,H              ;D=1ST OPERAND (ROW), E=2ND (COLUMN)
DF18 21575A                 LD HL,WINDLHS
DF1B 7E                     LD A,(HL)
DF1C 83                     ADD A,E             ;ADD COL
DF1D 5F                     LD E,A              ;ABS SCREEN COL
DF1E 2B                     DEC HL
DF1F 7E                     LD A,(HL)           ;WINDRHS
DF20 BB                     CP E
DF21 3821                   JR C,PATER          ;ERROR IF TOO FAR RIGHT
DF23
DF23 23                     INC HL              ;WINDLHS
DF24 23                     INC HL              ;WINDTOP
DF25 7E                     LD A,(HL)
DF26 82                     ADD A,D             ;ABS SCREEN ROW
DF27 57                     LD D,A
DF28 23                     INC HL
DF29 7E                     LD A,(HL)           ;WINDBOT
DF2A 92                     SUB D
DF2B 3013                   JR NC,POSTORH       ;JR IF 'AT' NOT OFF BOTTOM
DF2D
DF2D ED44                   NEG
DF2F 47                     LD B,A              ;NUMBER OF EXTRA ROWS NEEDED IN LS
DF30 3A735A                 LD A,(DEVICE)
DF33 A7                     AND A
DF34 280E                   JR Z,PATER          ;ERROR IF UPPER SCREEN, BUT LS CAN INCREASE SIZE
DF36
DF36 56                     LD D,(HL)
DF37 D5                     PUSH DE             ;D=WINDBOT, E=ABS 'AT' COLUMN
DF38
DF38 C5          SLSLP:     PUSH BC
DF39 CDB8DF                 CALL SCRLS
DF3C C1                     POP BC
DF3D 10F9                   DJNZ SLSLP          ;SCROLL LS UP 'B' TIMES TILL POSN IS ON RIGHT ROW
DF3F                                            ;(I.E. ADD ROWS TO TOP OF LOWER SCREEN)
DF3F D1                     POP DE
DF40
DF40 EB          POSTORH:   EX DE,HL
DF41 C362DC                 JP POSTORE
DF44
DF44 CF          PATER:     RST 08H
DF45 20                     DB 32               ;'OFF SCREEN'
DF46
DF46             ;SAVE CURRENT O/P ADDR
DF46
DF46 2A515C      SVCUROP:   LD HL,(CURCHL)
DF49 5E                     LD E,(HL)
DF4A 23                     INC HL
DF4B 56                     LD D,(HL)
DF4C ED53B55A               LD (OPSTORE),DE
DF50 C9                     RET
DF51
DF51             ;SCROLL SCREEN
DF51
DF51 3A3C5C      SCRLSCR:   LD A,(TVFLAG)
DF54 E610                   AND 10H
DF56 2810                   JR Z,SCRLSCR2       ;JR IF NOT AUTO-LIST
DF58
DF58 3A865B                 LD A,(BCREG)
DF5B 3D                     DEC A
DF5C 204D                   JR NZ,DOSCRL        ;JR IF CURRENT LINE NOT PRINTED YET
DF5E
DF5E CD6311                 CALL SETSTRM        ;(STREAM ZERO)
DF61 ED7B3F5C               LD SP,(LISTSP)      ;CLEAR STACK
DF65 C32C06                 JP AULX             ;JP TO SET 'AUTOLIST FINISHED'
DF68
DF68 3A735A      SCRLSCR2:  LD A,(DEVICE)
DF6B 3D                     DEC A
DF6C 284A                   JR Z,SCRLS          ;JR IF USING LOWER SCREEN
DF6E
DF6E CD5D0E                 CALL BRKCR
DF71 3ABB5A                 LD A,(SPROMPT)
DF74 A7                     AND A
DF75 2034                   JR NZ,DOSCRL        ;JR IF PROMPTS TURNED OFF
DF77
DF77 218C5C                 LD HL,SCRCT
DF7A 35                     DEC (HL)
DF7B 202E                   JR NZ,DOSCRL        ;JR IF NOT TIME FOR PROMPT YET
DF7D
DF7D CDAEDF                 CALL SETSCRCT
DF80 CDE0DF                 CALL SVTEMPS        ;SAVE TEMPORARY ATTRIBUTES ETC
DF83 2A515C                 LD HL,(CURCHL)
DF86 E5                     PUSH HL             ;PROBABLY CHANNEL 'S'
DF87 D9                     EXX
DF88 D5                     PUSH DE
DF89 3E01                   LD A,1              ;'Scroll?'
DF8B CD01E0                 CALL WTBRK
DF8E D1                     POP DE
DF8F D9                     EXX
DF90 FE20                   CP " "
DF92 2804                   JR Z,BRCH
DF94
DF94 E6DF                   AND 0DFH
DF96 FE4E                   CP "N"
DF98
DF98 CCCD08      BRCH:      CALL Z,IOPOF        ;INDENT O/P OFF. EXIT WITH Z
DF9B CA610E                 JP Z,BRCERR
```

```
DF9E
DF9E E1                 POP HL
DF9F 22515C             LD (CURCHL),HL
DFA2 CD7011             CALL CHANFLAG
DFA5 CDDEDF             CALL RSTTEMPS     ;RESTORE TEMPORARY ATTRIBUTES
DFA8 CDAC11             CALL COLEX
DFAB
DFAB C3950B     DOSCRL: JP EDRS1UP
DFAE
DFAE 2A585A     SETSCRCT: LD HL,(WINDTOP)
DFB1 7C                 LD A,H
DFB2 95                 SUB L
DFB3 3C                 INC A             ;A=WINDOW HEIGHT
DFB4 328C5C             LD (SCRCT),A
DFB7 C9                 RET
DFB8
DFB8
DFB8            ;SCROLL LOWER SCREEN. IF THE TOP OF THE LOWER SCREEN HITS UPPER SCREEN PRINT
DFB8            ;POSN THEN THE UPPER SCREEN HAS TO BE SCROLLED AS WELL, FROM TOP TO PRINT POSN
DFB8
DFB8 213E5A     SCRLS:  LD HL,LWTOP
DFBB 7E                 LD A,(HL)         ;LS TOP WILL MOVE UP BY ONE LINE
DFBC 3D                 DEC A
DFBD 2885               JR Z,PATER        ;OFF SCREEN ERROR IF WINDOW HIT SCREEN TOP
DFBF
DFBF 77                 LD (HL),A         ;NEW LWTOP
DFC0 32585A             LD (WINDTOP),A    ;ALSO RECORD NEW TOP FOR CURRENT (TEMP) WINDOW (LS)
DFC3 21C35A             LD HL,KPOS+1
DFC6 35                 DEC (HL)          ;CURSOR LINE KEPT UP-TO-DATE
DFC7 216D5A             LD HL,SPOSNU+1
DFCA BE                 CP (HL)
DFCB 20DE               JR NZ,DOSCRL      ;JR IF NO CLASH WITH UPPER SCREEN PRINT POSN
DFCD                                      ;(JR ALSO IF WINDTOP IS ABOVE US PRINT POSN - SIC)
DFCD F5                 PUSH AF
DFCE 35                 DEC (HL)          ;US PRINT POSN MOVES UP ONE LINE
DFCF CD5E11             CALL STREAMFE     ;MAIN WINDOW
DFD2 F1                 POP AF
DFD3 32595A             LD (WINDBOT),A    ;WIND BOT=FORMER PRINT POSN
DFD6 CD950B             CALL EDRS1UP      ;SCROLL MAIN WINDOW UP ONE
DFD9 CD6111             CALL STREAMFD     ;LOWER WINDOW
DFDC 18CD               JR DOSCRL         ;SCROLL LS
DFDE
DFDE A7         RSTTEMPS: AND A
DFDF 26                 DB 26H            ;'JR +1'
DFE0
DFE0 37         SVTEMPS: SCF
DFE1 214D5A             LD HL,THFATT
DFE4 11C85A             LD DE,TEMPW1      ;8 BYTES OF TEMP STORE AVAILABLE HERE
DFE7 3801               JR C,SVRSTTMPS
DFE9
DFE9 EB                 EX DE,HL
DFEA
DFEA C3A413     SVRSTTMPS: JP LDIR8       ;LD BC,8:LDIR:RET
DFED
DFED
DFED            ;COPY 8 BYTES FROM HL TO MEM1 AND MEM2, DOUBLING
DFED            ;EXIT WITH HL PTING TO MEM1
DFED
DFED C5         DBCHAR: PUSH BC
DFEE D5                 PUSH DE
DFEF 112951             LD DE,MEMVAL+8    ;(MEMVAL+0 MIGHT BE IN USE)
DFF2 D5                 PUSH DE
DFF3 0608               LD B,8
DFF5
DFF5 7E         DHLP:   LD A,(HL)
DFF6 23                 INC HL
DFF7 12                 LD (DE),A
DFF8 13                 INC DE
DFF9 12                 LD (DE),A
DFFA 13                 INC DE
DFFB 10F8               DJNZ DHLP         ;LOOP DOUBLES CHARACTER BYTES
DFFD
DFFD E1                 POP HL            ;PTR TO TOP HALF OF CHAR PAT
DFFE D1                 POP DE
DFFF C1                 POP BC
E000 C9                 RET
E001
E001            ;USED TO GIVE 'START TAPE..' PROMPT TO LOWER SCREEN, AND
E001            ;TO GIVE 'SCOLL?' PROMPT. ON ENTRY, A=UTMSG NUMBER
E001
E001 CD0DE0     WTBRK:  CALL WTB2
E004 CDB03D             CALL UTMSG
E007 CDB11C             CALL READKEY      ;CLEAR BUFFER
E00A CDE404             CALL GTKBK
E00D
E00D F5         WTB2:   PUSH AF
E00E CDB506             CALL CLSLOWER
E011 F1                 POP AF
E012 C9                 RET
E013
E013            ;USED TO PRINT CURSOR
E013
E013 ED4BDC5B   FONOP2: LD BC,(MNOP)
E017 C5                 PUSH BC
E018 C9                 RET
E019
E019                    INCLUDE TAPEMN.SAM      ;SAVE, LOAD, MERGE, VERIFY
E019            ;TAPEMN.SAM - SAVE/LOAD/MERGE/VERIFY CONTROL
E019            ;SYNTAX CHECK FOR ALL TAPE COMMANDS STARTS AT SLMVC
E019
E019            ;THERE ARE 2 BUFFERS - HDR (HEADER - REQUESTED) AND HDL (HEADER - LOADED)
E019            ;HEADER BUFFER FORMAT:
E019
```

```
E019            ;0       - TYPE (16=BAS, 17=NUM ARRAY, 18=STR ARRAY, 19=CODE, 20=SCREEN$)
E019            ;1-10    - (10) FILE NAME, PADDED WITH SPACES IF NEEDED
E019            ;11-14   - (4)  ALLOWS LONGER FILE NAME IF SLDEV<>T
E019            ;15      - (1)  FLAGS. BIT 0=INVIS. NAME, BIT 1=PROTECTED CODE
E019            ;16-26   - (11) IF TYPE 17 OR 18=ARRAY/STRING TLBYTE/NAME (11)
E019            ;16              IF TYPE 20     =SCREEN MODE
E019            ;16-18           IF TYPE 16     =PROG LEN EXCLUDING VARS (3)
E019            ;19-21           IF TYPE 16     =PROG LEN PLUS NVARS, EXCLUDING GAP AND SAVARS (3)
E019            ;22-24           IF TYPE 16     =PROG LEN PLUS NVARS, EXCLUDING SAVARS (3)
E019            ;27        (1) DIRE - DIRECTORY ENTRY NUMBER (HDR ONLY) NOT USED
E019            ;28-30   - (4)  SPARE
E019
E019            ;DISP OF HDN FROM BUFFER START:
E019
E019            ;31-33   - (3)  REL PAGE FORM START ADDR IF CODE, ACTUAL ADDR IF PROG OR DATA.
E019            ;               (OR FFXXXX IF ARRAY DOESN'T EXIST)
E019            ;34-36   - (3)  DATA LENGTH  (PAGEFORM) OR FFFFFF IF E.G. LOAD "" CODE
E019            ;               PROG+VARS LENGTH, OR ARRAY LEN (OR FFXXXX IF NON-EXISTENT)
E019            ;37-39   - (3)  EXECUTE ADDR. IF CODE, REL PAGE FORM, OR FFXXXX IF NO EXEC.
E019            ;               IF BAS AND NO AUTO-RUN, FIRST BYTE=FF, ELSE FIRST=0 AND THE
E019            ;               AUTO-RUN LINE NUMBER IS IN THE NEXT TWO BYTES.
E019
E019            ;40-79           COMMENT. NOT INITIALISED - CAN BE POKED, PEEKED.
E019
E019            ;SAVE, LOAD, VERIFY, MERGE ALL ENTER HERE
E019
E019 D6A6       SLMVC:      SUB 0A6H            ;OVERTOK
E01B 32B95B                 LD (OVERF),A
E01E 2001                   JR NZ,NOVE
E020
E020 E7                     RST 20H             ;SKIP "OVER"
E021
E021 2A065A     NOVE:       LD HL,(PSLD)
E024 22B75B                 LD (SLDEV),HL       ;PERM DEVICE TO TEMP
E027 CDA13A                 CALL EXPTSTR        ;NAME
E02A 3045                   JR NC,HDR2          ;JR IF NOT RUNNING
E02C
E02C 21014B                 LD HL,HDR+1         ;HEADER BUFFER ADDR
E02F 0619                   LD B,25
E031
E031 3620       HDCLP:      LD (HL),20H         ;CLEAR NAMES AREAS WITH SPACES
E033 23                     INC HL
E034 10FB                   DJNZ HDCLP
E036
E036 060E                   LD B,14
E038
E038 36FF       HDCLP2:     LD (HL),0FFH        ;CLEAR REST WITH FFH
E03A 23                     INC HL
E03B 10FB                   DJNZ HDCLP2
E03D
E03D CDEA3E                 CALL SBFSR          ;BC=NAME LEN, DE=START, IN SYS PAGE BUFFER **
E040 260B                   LD H,NMLEN+1
E042 CD26E5                 CALL TCHK
E045 2802                   JR Z,NLTP
E047
E047 260F                   LD H,NMLEN+5        ;ALLOW EXTRA CHARS IF NON-T
E049
E049 79         NLTP:       LD A,C
E04A BC                     CP H
E04B 300F                   JR NC,IFNER         ;LIMIT NAME LEN
E04D
E04D 21014B                 LD HL,HDR+1         ;DEST FOR FIRST NAME CHAR
E050 36FF                   LD (HL),0FFH        ;ASSUME NAME IS NULL
E052 A7                     AND A
E053 3A745B                 LD A,(CURCMD)
E056 2006                   JR NZ,TNMOK         ;JR IF NAME LEN IS 1-MAX CHARS LONG
E058
E058 FE94                   CP SAVETOK
E05A 2015                   JR NZ,HDR2          ;JR IF NOT SAVE (WITH A NULL NAME)
E05C
E05C CF         IFNER:      RST 08H
E05D 12                     DB 18               ;'Invalid file name'
E05E
E05E FE94       TNMOK:      CP SAVETOK
E060 200C                   JR NZ,MVTNM
E062
E062 1A                     LD A,(DE)           ;FIRST CHAR OF SAVE NAME CAN BE
E063 FE04                   CP 4                ;0 FOR NOT INVIS, NOT PROT
E065 3007                   JR NC,MVTNM         ;1 FOR     INVIS, NOT PROT
E067                                            ;2 FOR NOT INVIS, PROT
E067                                            ;3 FOR INVIS, PROT
E067 320F4B                 LD (HDR+HFG),A
E06A 13                     INC DE              ;(INVIS. MEANS NAME NOT PRINTED DURING LOADING,
E06B 0D                     DEC C               ;PROT. MEANS AUTO-RUN CODE FILE CANNOT BE STOPPED)
E06C 2803                   JR Z,HDR2           ;FIRST CHAR IS NOT INCLUDED IN NAME PROPER.
E06E
E06E EB         MVTNM:      EX DE,HL
E06F EDB0                   LDIR                ;COPY NAME TO BUFFER
E071
E071 DF         HDR2:       RST 18H
E072 FEFF                   CP 0FFH
E074 2870                   JR Z,HDR4           ;JR IF FN LEADER FOUND (SHOULD BE SCREEN$ OR CODE)
E076
E076 FE8C       HDR3:       CP LINETOK
E078 C279E1                 JP NZ,HDR6          ;IF NOT 'LINE', CHECK FOR 'DATA', OR CR/COLON
E07B
E07B 3A745B                 LD A,(CURCMD)
E07E FE97                   CP VERIFYTOK
E080 CA290D                 JP Z,NONSENSE       ;VERIFY "NAME" LINE n NOT ALLOWED
E083
E083 CDE33A                 CALL SEXPT1NUM
E086 300C                   JR NC,HDRLNOK       ;JR IF NOT RUNNING
E088
```

```
E088 CD2E1D               CALL GETINT        ;LINE
E08B 21254B               LD HL,HDR+HDN+6    ;PTR TO AUTO-RUN LINE AREA
E08E 3600                 LD (HL),0          ;FLAG 'AUTORUN'
E090 23                   INC HL
E091 71                   LD (HL),C
E092 23                   INC HL
E093 70                   LD (HL),B          ;PLACE LINE NO.
E094
E094            ;S/L/V/M "NAME":   ENTERS HERE, AND S/L/M "NAME" LINE n
E094
E094 CD153B     HDRLNOK:  CALL CHKEND
E097
E097 CD321F               CALL ADDRPROG
E09A 321F4B               LD (HDR+HDN),A
E09D 22204B               LD (HDR+HDN+1),HL ;START
E0A0 EB                   EX DE,HL
E0A1 4F                   LD C,A             ;CDE=PROG
E0A2 CD351F               CALL ADDRELN
E0A5 2B                   DEC HL             ;DEC AHL SO PROG LEN-1 USED. END OF SAVARS MARKER
E0A6 CB7C                 BIT 7,H            ;NOT SAVED. (SO ELINE NEVER = PROG WHEN PROG
E0A8 2001                 JR NZ,HDRPLC       ;SPACE IS RECLAIMED, MAKEROOM AT (PROG) ALWAYS OK)
E0AA
E0AA 3D                   DEC A
E0AB
E0AB CDE71F     HDRPLC:   CALL SUBAHLCDE     ;GET AHL=PAGE FORM OF DIFFERENCE
E0AE 32224B               LD (HDR+HDN+3),A
E0B1 22234B               LD (HDR+HDN+4),HL ;FILE LEN=ELINE-PROG
E0B4
E0B4 0603                 LD B,3             ;DO NVARS, NUMEND, SAVARS
E0B6 DD21895A             LD IX,NVARS+1
E0BA FD21104B             LD IY,HDR+16       ;STORE IN HDR+16/+19/+22
E0BE
E0BE DD6600     RNTVL:    LD H,(IX+0)
E0C1 DD2B                 DEC IX
E0C3 DD6E00               LD L,(IX+0)
E0C6 DD2B                 DEC IX
E0C8 DD7E00               LD A,(IX+0)
E0CB DD2B                 DEC IX
E0CD CDE71F               CALL SUBAHLCDE
E0D0 FD7700               LD (IY+0),A
E0D3 FD23                 INC IY
E0D5 FD7500               LD (IY+0),L
E0D8 FD23                 INC IY
E0DA FD7400               LD (IY+0),H
E0DD FD23                 INC IY
E0DF 10DD                 DJNZ RNTVL
E0E1                                         ;PROGRAM (EX. STR/ARRAYS AND GAP) LEN
E0E1                                         ;PROGRAM (EX. STR/ARRAYS) LEN
E0E1
E0E1 3E10                 LD A,16            ;TYPE=PROG
E0E3 C3E2E1               JP SLVMC           ;JP TO COMMON ROUTINE
E0E6
E0E6 E7         HDR4:     RST 20H            ;SKIP FFH
E0E7 FE6C                 CP CODETOK
E0E9 2049                 JR NZ,HDR5
E0EB
E0EB            ;HANDLE "NAME" CODE <START><,LENGTH><,EXECUTION ADDR>
E0EB
E0EB 111F4B               LD DE,HDR+HDN      ;PT TO HDR AREA FOR NUMBERS
E0EE 0603                 LD B,3             ;MAX OF 3 NUMS FOLLOW 'CODE'
E0F0
E0F0 E7                   RST 20H            ;SKIP 'CODE'
E0F1 CD7A3A               CALL CRCOLON
E0F4 200E                 JR NZ,HDRNMS       ;JR IF NOT JUST "NAME" CODE
E0F6
E0F6 3A745B               LD A,(CURCMD)
E0F9 FE94                 CP SAVETOK
E0FB 2803                 JR Z,HDNSHP        ;SAVE "NAME" CODE NEEDS PARAMS
E0FD
E0FD 182D                 JR HDRNOK
E0FF
E0FF 05         HDRNLP:   DEC B
E100
E100 CA290D     HDNSHP:   JP Z,NONSENSE      ;JR IF MORE THAN 3 NUMBERS USED
E103
E103 E7                   RST 20H
E104
E104 C5         HDRNMS:   PUSH BC            ;COUNT
E105 D5                   PUSH DE            ;PTR
E106 CDE43A               CALL EXPT1NUM
E109 300E                 JR NC,HDRN1        ;JR IF NOT RUNNING
E10B
E10B CD8C3F               CALL UNSTLEN       ;GET NUMBER IN AHL IN PAGE/ADDR FORM
E10E D1                   POP DE
E10F EB                   EX DE,HL           ;HL=PTR, ADE=ADDR
E110 77                   LD (HL),A
E111 23                   INC HL
E112 73                   LD (HL),E
E113 23                   INC HL
E114 CBFA                 SET 7,D
E116 72                   LD (HL),D
E117 23                   INC HL
E118 E5                   PUSH HL            ;PTR
E119
E119 D1         HDRN1:    POP DE
E11A C1                   POP BC             ;NO. COUNT
E11B DF                   RST 18H
E11C FE2C                 CP ","
E11E 28DF                 JR Z,HDRNLP        ;LOOP FOR ANOTHER NO. IF A COMMA FOLLOWED THE LAST
E120
E120 3A745B               LD A,(CURCMD)
E123 FE94                 CP SAVETOK
E125 2005                 JR NZ,HDRNOK
```

```
E127
E127 78                  LD A,B
E128 FE03                CP 3
E12A 28D4                JR Z,HDNSHP       ;SAVE CANNOT HAVE JUST SAVE "NAME" START
E12C
E12C CD153B     HDRNOK:  CALL CHKEND
E12F
E12F            ;EXECUTION ROUTINE FOR SAVE/LOAD/VERIFY "NAME" CODE ETC.
E12F
E12F 3E13                LD A,19           ;TYPE=CODE
E131 C3E2E1              JP SLVMC          ;JP TO COMMON ROUTINE
E134
E134 FE4C       HDR5:    CP SCRNTOK
E136 C2290D              JP NZ,NONSENSE
E139
E139 CD143B              CALL SABORTER     ;SKIP 'SCREEN$'
E13C
E13C CDB2EE              CALL FLITE        ;GET BC=LINE INT TABLE LEN
E13F 212800              LD HL,0028H
E142 09                  ADD HL,BC         ;ALLOW FOR PALTAB
E143 3A405A              LD A,(MODE)
E146 32104B              LD (HDR+16),A     ;MODE
E149 CD0CE5              CALL SCRLEN       ;GET SCREEN LEN FOR MODE IN ADE
E14C 4F                  LD C,A
E14D E5                  PUSH HL           ;LEN OF PALTAB AND LINITAB
E14E D5                  PUSH DE           ;SCREEN LEN
E14F 19                  ADD HL,DE
E150 EB                  EX DE,HL          ;ADE=TOT LEN
E151 211F4B              LD HL,HDR+HDN
E154 3A785A              LD A,(CUSCRNP)
E157 77                  LD (HL),A
E158 23                  INC HL
E159 3600                LD (HL),0
E15B 23                  INC HL
E15C 3680                LD (HL),80H       ;START IS AT 8000H IN SCREEN PAGE
E15E 23                  INC HL
E15F 71                  LD (HL),C         ;LEN MSB
E160 23                  INC HL
E161 73                  LD (HL),E
E162 23                  INC HL
E163 72                  LD (HL),D         ;LEN
E164 D1                  POP DE
E165 CBFA                SET 7,D           ;DE=SCREEN END IF M0/1
E167 0C                  INC C
E168 0D                  DEC C
E169 2801                JR Z,HDRSC2       ;JR IF M0 OR M1
E16B
E16B 3C                  INC A             ;SECOND PAGE OF M2/M3 SCREEN
E16C
E16C C1         HDRSC2:  POP BC            ;PALTAB+LINITAB LEN
E16D 21D855              LD HL,PALTAB
E170 CDDF3F              CALL TSURPG
E173 EDB0                LDIR              ;COPY TO END OF SCREEN
E175 3E14                LD A,20           ;'SCREEN$' TYPE
E177 1869                JR SLVMC          ;JR TO COMMON ROUTINE
E179
E179 FEB9       HDR6:    CP 0B9H           ;DATATOK
E17B C294E0              JP NZ,HDRLNOK
E17E
E17E 3A745B              LD A,(CURCMD)
E181 FE96                CP MERGETOK
E183 2823                JR Z,SVDNH        ;MERGE "NAME" DATA NOT ALLOWED
E185
E185 E7                  RST 20H           ;SKIP 'DATA'
E186 CD0102              CALL R1OFFCL
E189 733F                DW LENGSR         ;** BUG FIX
E18B F5                  PUSH AF           ;FOUND/NOT FOUND STATUS
E18C 0612                LD B,18           ;TYPE ASSUMED TO BE STRING
E18E 3A3B5C              LD A,(FLAGS)
E191 87                  ADD A,A
E192 FAA5E1              JP M,SVDNU        ;JP IF NUMERIC
E195
E195 3008                JR NC,SVDS1       ;JR IF NOT RUNNING
E197
E197 DF                  RST 18H
E198 FE29                CP ")"
E19A 2013                JR NZ,SVDTT       ;MUST BE NAME$
E19C
E19C E7                  RST 20H           ;SKIP ')'
E19D 1810                JR SVDTT          ;MUST BE NAME$()
E19F
E19F CB71       SVDS1:   BIT 6,C
E1A1 280C                JR Z,SVDTT        ;JR IF NO OPENING BRACKET USED
E1A3
E1A3 1806                JR SVNSC
E1A5
E1A5 05         SVDNU:   DEC B             ;TYPE=17
E1A6 CB69                BIT 5,C
E1A8
E1A8 CA290D     SVDNH:   JP Z,NONSENSE     ;ERROR IF NON-ARRAY NUMERIC
E1AB
E1AB DF         SVNSC:   RST 18H
E1AC CD953A              CALL INSISCBRK    ;INSIST ON ')', SKIP
E1AF
E1AF F1         SVDTT:   POP AF
E1B0 C5                  PUSH BC           ;B=TYPE, 17 (NUMERIC) OR 18 (STRING)
E1B1 2818                JR Z,SVNWAR       ;JR IF NOT FOUND
E1B3
E1B3 3A2351              LD A,(MEMVAL+2)
E1B6 321F4B              LD (HDR+HDN),A
E1B9 ED53204B            LD (HDR+HDN+1),DE ;START OF VARIABLE
E1BD 212451              LD HL,MEMVAL+3    ;TEXT LEN DATA STORED HERE
E1C0 CD6C2C              CALL ADD14        ;ALLOW FOR T/L,NAME,PG,MOD 16K
```

```
E1C3 32224B               LD (HDR+HDN+3),A
E1C6 22234B               LD (HDR+HDN+4),HL ;LEN OF VARIABLE INC. HEADER
E1C9 1808                 JR SVDTCE
E1CB
E1CB            ;NEW ARRAY
E1CB
E1CB 3A745B     SVNWAR:   LD A,(CURCMD)
E1CE FE95                 CP LOADTOK
E1D0 C2082D               JP NZ,VNFERR      ;ERROR IF VERIFYING OR SAVING A NEW ARRAY
E1D3
E1D3 F1         SVDTCE:   POP AF            ;TYPE
E1D4 CD153B               CALL CHKEND
E1D7
E1D7 213F51               LD HL,TLBYTE
E1DA 11104B               LD DE,HDR+16
E1DD 010B00               LD BC,11
E1E0 EDB0                 LDIR              ;COPY DETAILS OF VAR NAME TO HDR
E1E2
E1E2 32004B     SLVMC:    LD (HDR),A
E1E5
E1E5 3A745B     SLVM1:    LD A,(CURCMD)
E1E8 FE94                 CP SAVETOK
E1EA CABFE4               JP Z,SAMAIN       ;SAVE PROGRAM
E1ED
E1ED            ;LOAD/VERIFY/MERGE MAIN
E1ED
E1ED CD1AE5     LVMMAIN:  CALL TORN
E1F0 2808                 JR Z,LKTH         ;JR IF TAPE, OR N AND NO DOS
E1F2
E1F2 DD21004B             LD IX,HDR
E1F6 CF                   RST 08H
E1F7 81                   DB FOPHK          ;CALL DOS TO LOAD HDR (OPEN FILE FOR READ)
E1F8 1803                 JR LDFL
E1FA
E1FA            ;JUMP FROM DOS (E=3) - LOAD HEADER, THEN FILE FROM TAPE/NET
E1FA
E1FA CDF1E3     LKTH:     CALL LKHDR        ;GET THE CORRECT HEADER
E1FD
E1FD            ;JUMP FROM DOS (E=1) - LOAD JUST FILE FROM TAPE/NET
E1FD
E1FD 3A504B     LDFL:     LD A,(HDL)
E200 D613                 SUB 19
E202 2014                 JR NZ,LVM2        ;JR IF NOT CODE
E204
E204 3A745B               LD A,(CURCMD)
E207 FE96                 CP MERGETOK
E209 201E                 JR NZ,CDSCVE      ;JR IF LOAD/VERIFY CODE
E20B
E20B 3A5F4B               LD A,(HDL+HFG)
E20E E602                 AND 02H
E210 2017                 JR NZ,CDSCVE      ;JR IF PROT - CANNOT STOP AUTO-RUN CODE
E212
E212 3D                   DEC A
E213 32754B               LD (HDL+HDN+6),A  ;PREVENT AUTO-RUN IF MERGE UNPROTECTED CODE
E216 1811                 JR CDSCVE
E218
E218 3D         LVM2:     DEC A
E219 CAC0E2               JP Z,LDSCRN       ;JR IF TYPE=SCREEN$ (20)
E21C
E21C 3A745B               LD A,(CURCMD)
E21F FE95                 CP LOADTOK
E221 CA23E3               JP Z,LDPRDT       ;JR IF LOAD (A PROGRAM OR ARRAY - TYPES 16-18)
E224
E224 FE96                 CP MERGETOK
E226 CA5738               JP Z,MEPROG       ;JP IF MERGE PROGRAM - CONT WITH VERIFY
E229
E229            ;VERIFY PROGRAM, DATA, CODE OR SCREEN, LOAD CODE
E229
E229 CD3820     CDSCVE:   CALL RDRLEN       ;CDE='DESIRED' LEN FROM HDR
E22C 41                   LD B,C
E22D C5                   PUSH BC
E22E D5                   PUSH DE
E22F CD3B20               CALL RDLLEN       ;CDE=ACTUAL LEN FROM LOADED HDL (PAGEFORM)
E232 E1                   POP HL
E233 F1                   POP AF            ;AHL=DESIRED LEN VALUE, CDE=ACTUAL LEN OR FFFFFF
E234 3C                   INC A
E235 2814                 JR Z,HDLNM        ;JR IF DESIRED LEN UNSPECIFIED (ONLY IF CODE)
E237
E237 3D                   DEC A
E238 A7                   AND A
E239 ED52                 SBC HL,DE
E23B 99                   SBC A,C
E23C DABEE2               JP C,TERROR       ;ERROR IF LEN OF FILE IS LONGER THAN REQUESTED
E23F
E23F B4                   OR H
E240 B5                   OR L
E241 2808                 JR Z,HDLNM        ;EQUAL LENS OK FOR VERIFY OR LOAD
E243
E243 3A745B               LD A,(CURCMD)
E246 FE97                 CP VERIFYTOK
E248 CABEE2               JP Z,TERROR       ;IF LOAD, LEN CAN BE LESS THAN 'DESIRED'
E24B                                        ;BUT VERIFY INSISTS ON EQUALITY
E24B
E24B 211F4B     HDLNM:    LD HL,HDR+HDN
E24E CD3F20               CALL RDTHREE      ;CDE=DESIRED START
E251 3A004B               LD A,(HDR)
E254 FE13                 CP 19
E256 79                   LD A,C
E257 2010                 JR NZ,STSPEC3     ;FOR ANYTHING APART FROM CODE, 'START' IS ALWAYS
E259                                        ;'DESIRED' (E.G. CUR. PROG, SCREEN OR VAR. ADDR)
E259 3C                   INC A
E25A 2008                 JR NZ,STSPEC      ;JR IF DESIRED START WAS SPECIFIED: E.G.
E25C                                        ;VERIFY/LOAD "NAME" CODE 40000
```

```
E25C 216F4B                LD HL,HDL+HDN
E25F CD3F20                CALL RDTHREE       ;CDE=LOADED START
E262 1801                  JR STSPEC2
E264
E264             STSPEC:   ;  IN A,(LRPORT)     ;PAGE AT 0000
E264                       ;  ADD A,C          ;CONVERT REL LOAD ADDR TO ABS
E264                       ;  LD C,A
E264 0D                    DEC C
E265
E265 3AB35A      STSPEC2:  LD A,(LDCO)
E268 81                    ADD A,C
E269
E269 D5          STSPEC3:  PUSH DE
E26A CDDF3F                CALL TSURPG        ;START SWITCHED IN AT HL
E26D CD3B20                CALL RDLLEN        ;CDE=ACTUAL LEN FROM LOADED HDL (PAGE FORM)
E270 E1                    POP HL             ;START
E271 CD9CE2                CALL LDVDBLK       ;LOAD OR VERIFY BLOCK
E274 3A745B                LD A,(CURCMD)
E277 FE97                  CP VERIFYTOK
E279 C8                    RET Z
E27A
E27A             ;MUST BE LOAD CODE
E27A
E27A 3A5F4B                LD A,(HDL+HFG)
E27D CB4F                  BIT 1,A
E27F 200A                  JR NZ,HDNSTP       ;JR IF PROT - CANNOT STOP AUTO-RUN CODE
E281
E281 3A254B                LD A,(HDR+HDN+6)
E284 2A264B                LD HL,(HDR+HDN+7)
E287 FEFF                  CP 0FFH
E289 2009                  JR NZ,HDLDEX       ;JR IF LOAD EXEC - OVER-RIDES ANY SAVED EXEC
E28B
E28B 3A754B      HDNSTP:   LD A,(HDL+HDN+6)
E28E 2A764B                LD HL,(HDL+HDN+7) ;AHL=LOADED EXEC ADDR (REL PAGE FORM) OR FFXXXX
E291 FEFF                  CP 0FFH
E293 C8                    RET Z              ;RET IF NO EXEC ADDR
E294
E294 CD7912      HDLDEX:   CALL PDPSR2        ;SWITCH IN CORRECT PAGE, FIDDLE HL IF NEEDED
E297 44                    LD B,H
E298 4D                    LD C,L
E299 C30702                JP R1OFFCLBC       ;CALL BC WITH ROM1 OFF
E29C
E29C             ;LOAD OR VERIFY DATA BLOCK ACCORDING TO CURCMD
E29C
E29C 3A745B      LDVDBLK:  LD A,(CURCMD)
E29F FE97                  CP VERIFYTOK
E2A1 2801                  JR Z,LDVD2         ;JR WITH NC IF VERIFY
E2A3
E2A3             ;CALLED TO LOAD A DATA BLOCK
E2A3
E2A3 37          LDDBLK:   SCF                ;LOAD
E2A4
E2A4 3EFF        LDVD2:    LD A,0FFH          ;DATA BLOCK
E2A6 F5                    PUSH AF
E2A7 CD1AE5                CALL TORN          ;Z IF 'T', OR 'N' AND NO DOS
E2AA 280D                  JR Z,LDVD3         ;IF NO DOS LET ROM HANDLE NET LOAD
E2AC
E2AC DD21004B    DOSLV:    LD IX,HDR
E2B0 F1                    POP AF             ;HL=DEST, CDE=LEN
E2B1 3803                  JR C,DOSLD
E2B3
E2B3 CF                    RST 08H
E2B4 83                    DB VFYHK           ;DOS VERIFY FOR DISC/NET ETC
E2B5 C9                    RET
E2B6
E2B6 CF          DOSLD:    RST 08H
E2B7 82                    DB LDHK            ;DOS LOAD FOR DISC/NET ETC
E2B8 C9                    RET
E2B9
E2B9 F1          LDVD3:    POP AF
E2BA CD0EE6                CALL LDBYTES
E2BD D8                    RET C
E2BE
E2BE CF          TERROR:   RST 08H
E2BF 13                    DB 19              ;'LOADING ERROR'
E2C0
E2C0
E2C0             ;LOAD/VERIFY SCREEN$ FILE
E2C0
E2C0 3A405A      LDSCRN:   LD A,(MODE)
E2C3 4F                    LD C,A
E2C4 21604B                LD HL,HDL+16
E2C7 7E                    LD A,(HL)
E2C8 FE20                  CP 20H             ;20H IF THIS IS A CODE FILE LOADED AS SCREEN$
E2CA 2002                  JR NZ,LDSC2        ;JR IF IT IS A SCREEN$ FILE LOADED AS SCREEN$
E2CC
E2CC 71                    LD (HL),C          ;SET MODE=CURRENT MODE FOR CODE FILE
E2CD 79                    LD A,C             ;SKIP MODET
E2CE
E2CE B9          LDSC2:    CP C
E2CF C45A01                CALL NZ,MODET      ;SELECT MODE IF A NEW ONE IS NEEDED
E2D2
E2D2 CDB63F                CALL SELSCRN
E2D5 21724B                LD HL,HDL+HDN+3
E2D8 CD3F20                CALL RDTHREE       ;CDE=LOADED LEN
E2DB 210080                LD HL,8000H        ;SCREEN START
E2DE D5                    PUSH DE
E2DF CD9CE2                CALL LDVDBLK
E2E2 D1                    POP DE
E2E3 3A745B                LD A,(CURCMD)
E2E6 FE97                  CP VERIFYTOK
E2E8 C8                    RET Z
E2E9
```

```
E2E9 3A604B               LD A,(HDL+16)     ;MODE
E2EC D5                   PUSH DE
E2ED CD0CE5               CALL SCRLEN       ;DE=SCREEN LEN MOD 16K FOR MODE. NC
E2F0 E1                   POP HL            ;FILE LEN MOD 16K
E2F1 CBBC                 RES 7,H
E2F3 ED52                 SBC HL,DE         ;HL=PALTAB/LINICOLS
E2F5 C8                   RET Z             ;RET IF EXACT LEN CODE FILE
E2F6
E2F6 D8                   RET C             ;OR SHORT
E2F7
E2F7 CBFA                 SET 7,D           ;DE=9B00 IF MODE 0, BB00 IF MODE 1, ELSE A000 (AND
E2F9                                        ;LOAD BLOCK MEANS C/D PAGED).
E2F9 012800               LD BC,40          ;SUB PALTAB LEN
E2FC ED42                 SBC HL,BC
E2FE E5                   PUSH HL           ;HL=LINICOL LEN IN FILE
E2FF EB                   EX DE,HL          ;HL PTS TO PALTAB AND LINICOLS DATA
E300 3A785A               LD A,(CUSCRNP)
E303 CD2CE5               CALL CUS2         ;Z IF CURRENT SCREEN IS DISPLAYED
E306 11D855               LD DE,PALTAB
E309 2803                 JR Z,PMV          ;MOVE LOADED PALTAB TO ACTIVE PALTAB, OR STORED
E30B                                        ;AREA, IF CURRENT SCRN IS INVIS.
E30B 11D8BF               LD DE,PALBUF-4000H
E30E
E30E EDB0       PMV:      LDIR              ;COPY PALTAB (40 BYTES)
E310
E310 7E                   LD A,(HL)
E311 FEC3                 CP 195
E313 12                   LD (DE),A
E314 C1                   POP BC            ;LINICOL LEN
E315 D0                   RET NC            ;RET IF NO LINICOLS
E316
E316 78                   LD A,B
E317 FE02                 CP 2
E319 D0                   RET NC            ;LIMIT LINICOL LEN TO 01FFH FOR SAFETY
E31A
E31A F3                   DI
E31B EDB0                 LDIR
E31D 3EFF                 LD A,0FFH
E31F D3F9                 OUT (STATPORT),A  ;PREVENT ANY LINE INTS TILL AFTER NEXT FRAME
E321 FB                   EI
E322 C9                   RET
E323
E323            ;LOAD PROGRAM OR DATA FILE
E323
E323 CD3B20     LDPRDT:   CALL RDLLEN       ;CDE=LEN OF FILE TO LOAD (PAGEFORM)
E326 D5                   PUSH DE
E327 79                   LD A,C
E328 CD3820               CALL RDRLEN       ;CDE=LEN OF CURRENT ('DESIRED') AREA
E32B E1                   POP HL            ;AHL=LOADING LEN
E32C 0C                   INC C
E32D 2806                 JR Z,LDNAR        ;JR IF LOADING A NEW ARRAY (CUR LEN=FFXXXX)
E32F
E32F 0D                   DEC C
E330 CDE71F               CALL SUBAHLCDE
E333 3803                 JR C,LDSZOK       ;JR IF CURRENT FILE IS BIGGER THAN LOADING
E335
E335 CD1E3F     LDNAR:    CALL TSTRMAHL     ;CHECK AHL BYTES OK. (PAGEFORM)
E338
E338 211F4B     LDSZOK:   LD HL,HDR+HDN
E33B CD3F20               CALL RDTHREE      ;START OF EXISTING ARRAY (PAGE FORM) OR FFXXXX
E33E 79                   LD A,C
E33F 0C                   INC C
E340 282D                 JR Z,LDCRA        ;JR IF NEW ARRAY
E342
E342 F5                   PUSH AF
E343 D5                   PUSH DE           ;START
E344 CD3820               CALL RDRLEN       ;CURRENT LEN TO CDE (PAGEFORM)
E347 D5                   PUSH DE
E348 51                   LD D,C
E349 C1                   POP BC            ;DBC=LEN
E34A E1                   POP HL
E34B F1                   POP AF            ;AHL=START
E34C CDDF3F               CALL SELURPG
E34F 3A004B               LD A,(HDR)
E352 D610                 SUB 16
E354 2015                 JR NZ,LDCRX       ;JR IF NOT PROGRAM (IT'S AN ARRAY)
E356
E356 32875A               LD (NVARSP),A
E359 32895A               LD (NVARS+1),A    ;SINCE PROG AND NVARS HAVE BEEN DELETED,
E35C                                        ;PREVENT XOINTERS ALTERING NVARS, CAUSING
E35C                                        ;SADJ WITH NO FOR-NEXTS, OR ODD BSTK
E35C 7A                   LD A,D            ;ABC=LEN
E35D CD531E               CALL RECL2BIG     ;DELETE CURRENT PROGRAM
E360 2AC65B               LD HL,(BASSTK)
E363 22C45B               LD (BSTKEND),HL   ;CLEAR BASIC STACK - DATA OBSOLETE
E366 CD321F               CALL ADDRPROG     ;ADDR PROG
E369 180A                 JR LDCR3
E36B
E36B 7A         LDCRX:    LD A,D            ;ABC=LEN
E36C CD531E               CALL RECL2BIG     ;DELETE CURRENT ARRAY
E36F
E36F CD351F     LDCRA:    CALL ADDRELN      ;ADDRESS ELINE
E372 CDB91F               CALL DECPTR       ;DEC HL AND CHANGE PAGE IF BELOW 8000H
E375
E375 E5         LDCR3:    PUSH HL
E376 CD3B20               CALL RDLLEN       ;CDE=LOADING LEN (PAGEFORM)
E379 E1                   POP HL
E37A C5                   PUSH BC
E37B D5                   PUSH DE
E37C 79                   LD A,C
E37D 42                   LD B,D
E37E 4B                   LD C,E
E37F CD1C1E               CALL MKRBIG       ;OPEN ABC AT HL
```

```
E382 36FF                LD (HL),0FFH      ;IN CASE LOAD FAILS!!
E384 D1                  POP DE
E385 C1                  POP BC
E386 DBFB                IN A,(251)
E388 F5                  PUSH AF
E389 E5                  PUSH HL
E38A CDA3E2              CALL LDDBLK       ;LOAD MAIN BLOCK
E38D D1                  POP DE
E38E C1                  POP BC            ;BDE=START OF MAIN BLK
E38F 3A004B              LD A,(HDR)
E392 FE10                CP 16
E394 2815                JR Z,LDPROG       ;JR IF PROGRAM WAS LOADED
E396
E396 78                  LD A,B
E397 CDDF3F              CALL TSURPG       ;DE PTS TO LOADED ARRAY TYPE/LEN BYTE
E39A 21104B              LD HL,HDR+16
E39D 1A                  LD A,(DE)
E39E AE                  XOR (HL)
E39F E6F0                AND 0F0H
E3A1 AE                  XOR (HL)
E3A2 12                  LD (DE),A         ;TYPE FROM LOADED ARRAY, NAME LEN FROM REQUESTED
E3A3 13                  INC DE            ;PT TO NAME
E3A4 23                  INC HL
E3A5 010A00              LD BC,10
E3A8 EDB0                LDIR              ;COPY REQUESTED NAME TO SAVARS
E3AA C9                  RET
E3AB
E3AB            ;BASIC FILE LOADED - SET VARS
E3AB
E3AB 0603       LDPROG:  LD B,3            ;DO NVARS, NUMEND, SAVARS
E3AD 21604B              LD HL,HDL+16
E3B0 FD21895A            LD IY,NVARS+1
E3B4
E3B4 CD3F20     SNTVL:   CALL RDTHREE      ;CDE=HDL DATA (LEN OF PROG ALONE, WITH NVARS, ETC)
E3B7 23                  INC HL
E3B8 E5                  PUSH HL
E3B9 CD321F              CALL ADDRPROG
E3BC CDDE1F              CALL ADDAHLCDE
E3BF FD7400              LD (IY+0),H
E3C2 FD2B                DEC IY
E3C4 FD7500              LD (IY+0),L
E3C7 FD2B                DEC IY
E3C9 FD7700              LD (IY+0),A
E3CC FD2B                DEC IY
E3CE E1                  POP HL            ;HDL PTR
E3CF 10E3                DJNZ SNTVL        ;SET NVARS, NUMEND, SAVARS
E3D1
E3D1 CDA234              CALL RESTOREZ     ;SET DATA PTR TO START OF PROG
E3D4 CD0102              CALL R1OFFCL
E3D7 DB33                DW DOCOMP         ;DEF PROCS/DEF FNS AND LABELS NEED DOING -
E3D9                                       ;ANY PRE-PASS OF LABELS ETC IS OBSOLETE.
E3D9 3A254B              LD A,(HDR+HDN+6)
E3DC 2A264B              LD HL,(HDR+HDN+7)
E3DF A7                  AND A
E3E0 2808                JR Z,LDUSLN       ;JR IF THERE'S A DESIRED START LINE (LD "N" LINE n)
E3E2
E3E2 3A754B              LD A,(HDL+HDN+6)
E3E5 2A764B              LD HL,(HDL+HDN+7)
E3E8 A7                  AND A
E3E9 C0                  RET NZ
E3EA
E3EA 3D         LDUSLN:  DEC A             ;A=FF
E3EB 32465C              LD (PPC+1),A      ;ENSURE GOTO SEARCH STARTS FROM PROG, NOT CLA
E3EE C39A19              JP GOTO3          ;SET VARS SO GOTO LINE HAPPENS NEXT
E3F1
E3F1            ;LOOK FOR A HEADER FROM TAPE OR NET.
E3F1            ;LOOK FOR A MATCHING HEADER (USING HDR AREA), LOADING HEADERS TO HDL AREA
E3F1            ;PRINT NAMES IF BIT 0,(TPROMPTS)=0. LOOP TILL MATCH FOUND.
E3F1
E3F1 CD66E4     LKHDR:   CALL LDHDR
E3F4 30FB                JR NC,LKHDR        ;LOOP IF ERROR
E3F6
E3F6 CD5E11              CALL STREAMFE     ;"S"
E3F9 3A504B              LD A,(HDL)        ;TYPE
E3FC 4F                  LD C,A
E3FD F610                OR 10H
E3FF FE15                CP 15H
E401 30EE                JR NC,LKHDR
E403
E403 215F4B              LD HL,HDL+HFG
E406 3A335A              LD A,(TPROMPTS)
E409 E601                AND 1             ;ISOLATE SYS VAR 'PRINT NAMES' BIT.
E40B B6                  OR (HL)           ;OR WITH HDR VERSION
E40C 77                  LD (HL),A
E40D 1F                  RRA
E40E 3812                JR C,LKHNP        ;JR IF NO PRINTING OF NAMES WANTED
E410
E410 3E0D                LD A,0DH
E412 D7                  RST 10H
E413 3E2A                LD A,"*"
E415 CB61                BIT 4,C
E417 CC1000              CALL Z,0010H      ;PRINT "*" BEFORE NAME IF ZX TYPE
E41A 79                  LD A,C            ;A=16-20
E41B F610                OR 10H
E41D D60C                SUB 12            ;A=4-8
E41F CDB03D              CALL UTMSG        ;PRINT TYPE MSG
E422
E422 01000A     LKHNP:   LD BC,NMLEN*256   ;C=0
E425 11504B              LD DE,HDL
E428 21004B              LD HL,HDR
E42B 1A                  LD A,(DE)         ;LOADED TYPE
E42C BE                  CP (HL)           ;CP DESIRED TYPE
E42D 280F                JR Z,LKHTM
```

```
E42F
E42F D614                   SUB 20              ;19->255, 20->0
E431 CE00                   ADC A,0             ;19->0,20->0
E433 2008                   JR NZ,SNMN          ;JR IF NOT CODE/SCREEN$
E435
E435 7E                     LD A,(HL)           ;ELSE
E436 12                     LD (DE),A           ;LOADED=DESIRED (DESPITE UTMSG)
E437 D614                   SUB 20
E439 CE00                   ADC A,0
E43B 2801                   JR Z,LKHTM          ;IF OTHER FILE IS CODE/SCREEN$, ACCEPT MATCH
E43D
E43D 0C           SNMN:     INC C               ;'NO MATCH' - TYPES DIFFER
E43E
E43E 23           LKHTM:    INC HL
E43F 13                     INC DE
E440 7E                     LD A,(HL)
E441 3C                     INC A
E442 2002                   JR NZ,PRHDLP
E444
E444 62                     LD H,D              ;IF DESIRED NAME=NULL (STARTS 0FFH) THEN
E445 6B                     LD L,E              ;ENSURE 'MATCH' OF LOADED NAME BY COMP VS. SELF!
E446
E446 1A           PRHDLP:   LD A,(DE)
E447 AE                     XOR (HL)
E448 E6DF                   AND 0DFH            ;IGNORE MISMATCH ON BIT 5 (LC/UC)
E44A 2801                   JR Z,PRHDC          ;JR IF MATCH BETWEEN LOADED NAME CHAR AND DESIRED
E44C
E44C 0C                     INC C               ;NZ C REG SHOWS A MISMATCH
E44D
E44D 3A5F4B       PRHDC:    LD A,(HDL+HFG)
E450 1F                     RRA
E451 1A                     LD A,(DE)           ;LOADED NAME CHAR
E452 D41000                 CALL NC,0010H       ;PRINT UNLESS TURNED OFF
E455 23                     INC HL
E456 13                     INC DE
E457 10ED                   DJNZ PRHDLP
E459
E459 79                     LD A,C
E45A A7                     AND A
E45B 2094                   JR NZ,LKHDR         ;JR IF ANY MISMATCH OCCURRED - LOOK FOR ANOTHER
E45D
E45D 3A5F4B                 LD A,(HDL+HFG)
E460 1F                     RRA
E461 3E0D                   LD A,0DH
E463 D8                     RET C
E464
E464 D7                     RST 10H             ;CR AFTER LAST (MATCHED) FILE NAME
E465 C9                     RET
E466
E466              ;LOAD A HEADER
E466
E466 21504B       LDHDR:    LD HL,HDL
E469 115000                 LD DE,80
E46C AF                     XOR A
E46D 4F                     LD C,A              ;CDE=80 BUT DE WILL CHANGE TO 17 IF ZX HDR FOUND
E46E 3C                     INC A               ;'HEADER'
E46F 37                     SCF                 ;'LOAD'
E470 CD0EE6                 CALL LDBYTES
E473 D0                     RET NC              ;RET IF LOAD FAILED
E474
E474 21504B                 LD HL,HDL
E477 7E                     LD A,(HL)           ;TYPE 0-3 IF ZX HEADER, 16-20 IF SAM
E478 CB67                   BIT 4,A
E47A C0                     RET NZ              ;RET IF A SAM HEADER. CY STILL SHOWS 'OK'
E47B
E47B D603                   SUB 3
E47D 3E00                   LD A,0
E47F 325F4B                 LD (HDL+HFG),A      ;ENSURE VISIBLE, UNPROTECTED
E482 C0                     RET NZ              ;RET IF OTHER THAN ZX CODE HDR (CY)
E483
E483              ;TRANSFORM 17-BYTE ZX CODE HEADER INTO SAM HEADER
E483
E483 3613                   LD (HL),19          ;TYPE=SAM CODE
E485 EB                     EX DE,HL
E486 AF                     XOR A
E487 2A5B4B                 LD HL,(HDL+11)      ;ZX FILE LEN
E48A CDF21F                 CALL PAGEFORM
E48D 32724B                 LD (HDL+HDN+3),A
E490 22734B                 LD (HDL+HDN+4),HL   ;SAM LENGTH FORMAT
E493 0E20                   LD C,20H            ;NORMAL 'SCREEN MODE' FOR SAM CODE FILE **
E495 2A5D4B                 LD HL,(HDL+13)      ;ZX START ADDR
E498 7C                     LD A,H              ;DEST ADDR MSB
E499 FE40                   CP 40H
E49B 3809                   JR C,HDTR5
E49D
E49D FE5B                   CP 5BH
E49F 3005                   JR NC,HDTR5
E4A1
E4A1              ;MUST BE LOADING TO SCREEN
E4A1
E4A1 3E14                   LD A,20
E4A3 12                     LD (DE),A           ;FILE TYPE=SCREEN$
E4A4 0E00                   LD C,0              ;MODE 0
E4A6
E4A6 79           HDTR5:    LD A,C
E4A7 32604B                 LD (HDL+16),A       ;SET SCREEN MODE OR 'NO MODE'
E4AA AF                     XOR A
E4AB 325F4B                 LD (HDL+HFG),A      ;NO PROTECT OR INVIS
E4AE CDF21F                 CALL PAGEFORM
E4B1 3D                     DEC A
E4B2 326F4B                 LD (HDL+HDN),A
E4B5 22704B                 LD (HDL+HDN+1),HL   ;SAM START ADDR FORMAT (IRREL IF SCREEN$)
E4B8 3EFF                   LD A,0FFH
```

```
E4BA 32754B               LD (HDL+HDN+6),A  ;NO EXEC ADDR
E4BD 37                   SCF               ;'OK'
E4BE C9                   RET
E4BF
E4BF 211F4B     SAMAIN:   LD HL,HDR+HDN
E4C2 3A004B               LD A,(HDR)
E4C5 FE13                 CP 19
E4C7 3E00                 LD A,0
E4C9 2002                 JR NZ,SAPA2       ;JR IF NOT CODE
E4CB
E4CB DBFA                 IN A,(250)        ;ADJUST START PAGE
E4CD
E4CD 86         SAPA2:    ADD A,(HL)
E4CE 77                   LD (HL),A
E4CF CD1AE5               CALL TORN
E4D2 2807                 JR Z,SVFL         ;LET ROM HANDLE T SAVE, OR NET SAVE IF NO DOS
E4D4
E4D4 DD21004B   SADOS:    LD IX,HDR
E4D8 CF                   RST 08H
E4D9 84                   DB SVHK           ;DOS SAVE
E4DA C9                   RET
E4DB
E4DB            ;JUMP FROM DOS (E=2) - SAVE ENTIRE FILE TO TAPE OR NET
E4DB
E4DB 3A335A     SVFL:     LD A,(TPROMPTS)
E4DE 1F                   RRA
E4DF 1F                   RRA
E4E0 3808                 JR C,SAHDDB       ;JR IF NO SAVE PROMPTS WANTED
E4E2
E4E2 CD26E5               CALL TCHK
E4E5 3E03                 LD A,3            ;'START TAPE..' MSG NO.
E4E7
E4E7 CC01E0               CALL Z,WTBRK      ;GIVE MSG IF TEMP DEVICE=T
E4EA
E4EA            ;SAVE HEADER, AND DATA BLOCK SPECIFIED BY HEADER
E4EA
E4EA            SAHDDB:  ;  IN A,(250)
E4EA                     ; INC A             ;PAGE WITH HDR IN
E4EA                     ;  CALL TSURPG      ;PAGE IN AT 8000H
E4EA AF                    XOR A
E4EB 4F                    LD C,A
E4EC 3C                    INC A             ;'HEADER' (A=1)
E4ED 115000                LD DE,80          ;CDE=80 (HEADER LEN)
E4F0                     ;  LD HL,HDR+4000H
E4F0 21004B                LD HL,HDR
E4F3 CD08E6                CALL SABYTES      ;SAVE HEADER
E4F6
E4F6 211F4B                LD HL,HDR+HDN
E4F9 7E                    LD A,(HL)
E4FA CDDF3F                CALL TSURPG
E4FD 23                    INC HL
E4FE 5E                    LD E,(HL)
E4FF 23                    INC HL
E500 56                    LD D,(HL)         ;DE=ADDR TO START SAVING FROM (SWITCHED IN)
E501 D5                    PUSH DE
E502 23                    INC HL
E503 CD3F20                CALL RDTHREE      ;CDE=LEN
E506 E1                    POP HL            ;START
E507 3EFF                  LD A,0FFH         ;MAIN BLOCK
E509 C308E6                JP SABYTES
E50C
E50C
E50C            ;GET SCREEN LEN
E50C            ;ENTRY: A=MODE. EXIT: ADE=001B00H IF MODE 0, 003800H IF MODE 1, ELSE 012000H
E50C
E50C 11001B     SCRLEN:   LD DE,1B00H       ;MODE 0 SCRN LEN
E50F A7                   AND A
E510 C8                   RET Z
E511
E511 1638                 LD D,38H          ;MODE 1 SCRN LEN=3800H
E513 3D                   DEC A
E514 C8                   RET Z
E515
E515 3E01                 LD A,1            ;A=16K PAGE. MODES 2 AND 3 LEN=4000H+2000H
E517 1620                 LD D,20H
E519 C9                   RET
E51A
E51A CD26E5     TORN:     CALL TCHK
E51D C8                   RET Z
E51E
E51E FE4E                 CP "N"
E520 C0                   RET NZ
E521
E521 3AC25B               LD A,(DOSFLG)
E524 A7                   AND A
E525 C9                   RET
E526
E526 3AB75B     TCHK:     LD A,(SLDEV)
E529 FE54                 CP "T"
E52B C9                   RET
E52C
E52C            ;CALLED BY VIDSEL AND TAPEMN
E52C
E52C E67F       CUS2:     AND 7FH           ;MIDI OUT BIT INACTIVE
E52E 57                   LD D,A
E52F DBFC                 IN A,(VIDPORT)
E531 E67F                 AND 7FH
E533 BA                   CP D
E534 C9                   RET
E535
E535                      INCLUDE TAPEX.SAM
E535            ;
E535            ;TAPEX.SAM - TAPE/NET EXECUTIVE ROUTINES
```

```
E535
E535            ;SAVE CDE BYTES FROM HL TO NET. TYPE ON STACK
E535
E535                                        ;HDR/DATA FLAG ON STACK
E535 C5         WNF:        PUSH BC
E536 0664                   LD B,100
E538 CD5F00                 CALL DELBC
E53B C1                     POP BC
E53C CD3DE7                 CALL CKNET      ;WAIT FOR FREE NET
E53F F1                     POP AF
E540 CD4DE5                 CALL NMOUT      ;SEND TYPE
E543
E543 DD2142E6               LD IX,NOSR
E547 CD6CE7                 CALL NIXR
E54A
E54A            ;NET SEND PARITY
E54A
E54A D9                     EXX
E54B 7C                     LD A,H
E54C D9                     EXX
E54D
E54D            ;NET/MIDI OUT
E54D
E54D F5         NMOUT:      PUSH AF
E54E
E54E CD5D0E     NMOUT1:     CALL BRKCR
E551 AF                     XOR A
E552 DBF8                   IN A,(CLUTPORT)
E554 E602                   AND 2
E556 20F6                   JR NZ,NMOUT1
E558
E558 CD5D0E                 CALL BRKCR
E55B F1                     POP AF
E55C D3FD                   OUT (MDIPORT),A
E55E C9                     RET
E55F
E55F            ;ENTRY: HL POINTS TO DATA TO SAVE, CDE=LEN TO SAVE (PAGEFORM) A=TYPE (AND FIRST
E55F            ;BYTE TO SAVE). 01=HEADER, FF=DATA
E55F            ;EXIT: NC IF ALL SAVED, CY IF SPACE PRESSED
E55F
E55F F3         SABLK:      DI
E560 F5                     PUSH AF
E561 CD26E5                 CALL TCHK
E564 20CF                   JR NZ,WNF
E566
E566 CD0420                 CALL CDENORM    ;CONVERT CDE TO 19-BIT
E569 F1                     POP AF
E56A F5                     PUSH AF
E56B D9                     EXX
E56C 47                     LD B,A
E56D 08                     EX AF,AF'       ;INITIALISE PARITY BYTE WITH TYPE
E56E 3AB85B                 LD A,(SLDEV+1)  ;SPEED. MAX SPEED IF 19 (GIVES ABOUT 5*ZX)
E571                                        ;HIGHER VALUES GIVE LOWER SPEEDS.
E571 6F                     LD L,A
E572 3C                     INC A
E573 87                     ADD A,A
E574 3002                   JR NC,SVLC
E576
E576 3EFE                   LD A,0FEH       ;HI VALUE IF CY
E578
E578 3C         SVLC:       INC A
E579 67                     LD H,A          ;H PRODUCES DELAY ABOUT TWICE AS LONG AS L
E57A E5                     PUSH HL
E57B
E57B 4F                     LD C,A          ;LEADER DELAY
E57C
E57C            ;NOW FIDDLE NO. OF LEADER CYCLES TO USE TO KEEP LEN ROUGHLY CONST. WHATEVER THE
E57C            ;SVSPEED. 128 OR MORE=LC*2, 86-127=LC*3, 64-85=LC*4, 52-63=LC*5, 43-51=LC*6
E57C            ;37-42=LC*8,.... 19=LC*14
E57C
E57C AF                     XOR A
E57D 67                     LD H,A
E57E 6F                     LD L,A
E57F 11B80B                 LD DE,3000
E582
E582 19         SVCLC:      ADD HL,DE
E583 81                     ADD A,C
E584 30FC                   JR NC,SVCLC
E586
E586 04                     INC B
E587 1001                   DJNZ SVTYP      ;JR IF DATA
E589
E589 29                     ADD HL,HL       ;LONGER LEADER IF HEADER
E58A
E58A 110202     SVTYP:      LD DE,0202H     ;D'=BORDER LS 3 BITS - RED
E58D
E58D 7A         SVHDR:      LD A,D
E58E EE0F                   XOR 0FH
E590 57                     LD D,A
E591 D3FE                   OUT (KEYPORT),A
E593 41                     LD B,C
E594
E594 05         SVDL0:      DEC B
E595 20FD                   JR NZ,SVDL0     ;DELAY REG*16
E597
E597 2B                     DEC HL
E598 7C                     LD A,H
E599 B5                     OR L
E59A 20F1                   JR NZ,SVHDR     ;LOOP TILL LEADER SAVED
E59C
E59C CB39                   SRL C
E59E CB39                   SRL C           ;QUARTER LENGTH PULSES NOW
E5A0 0C                     INC C
```

```
E5A1 0C                     INC C
E5A2 0C                     INC C               ;IN CASE VERY SMALL - HELPS AVOID SYNC BEING MISSED
E5A3 2E02                   LD L,2              ;2 HALF-CYCLES ONLY
E5A5 1D                     DEC E
E5A6 20E5                   JR NZ,SVHDR         ;JR IF SYNC PULSE TIME
E5A8
E5A8 E1                     POP HL              ;DELAY CONSTANTS FOR HI/LOW BITS
E5A9 F1                     POP AF              ;TYPE BYTE
E5AA 5F                     LD E,A              ;FIRST BYTE
E5AB 1601                   LD D,1              ;D'=BORDER LS 3 BITS - BLUE
E5AD 37                     SCF
E5AE CB13                   RL E
E5B0
E5B0 4C           SVBL:     LD C,H              ;ASSUME HI BIT
E5B1 3801                   JR C,SV2
E5B3
E5B3 4D                     LD C,L              ;GET DELAY FOR A LOW BIT
E5B4
E5B4 7A           SV2:      LD A,D              ;BORDER COLOUR.
E5B5 EE0F                   XOR 0FH             ;FLIP MIC AND BORDER BITS
E5B7 D3FE                   OUT (KEYPORT),A     ;38 OR 37 FROM SVBL TO END OF OUT INSTRUCTION
E5B9                                            ;55 OR 54 FROM PREV OUT, PLUS DELAY (IF WITHIN BYTE
E5B9 41                     LD B,C
E5BA
E5BA 10FE         SVDL:     DJNZ SVDL           ;THESE 2 INSTRUCTIONS GIVE DELAY OF 13*C, -1
E5BC
E5BC 0D                     DEC C
E5BD 0D                     DEC C               ;SUB 26 Ts FROM DELAY
E5BE EE0F                   XOR 0FH             ;NC
E5C0 CB13                   RL E
E5C2 D3FE                   OUT (KEYPORT),A     ;34 TO END OF OUT FROM PREV OUT, PLUS DELAY
E5C4
E5C4 CACCE5                 JP Z,SVBY           ;JP IF TIME FOR NEXT BYTE
E5C7                                            ;ELSE CY/NC SHOWS NEXT BIT
E5C7 41                     LD B,C
E5C8
E5C8 10FE         SVDL2:    DJNZ SVDL2
E5CA
E5CA 18E4                   JR SVBL
E5CC
E5CC              ;FROM SVBL: 38/37,OUT,13*R-1,34,OUT,13*R-27,22; 38/37,OUT ETC. R=REGISTER
E5CC              ;E.G. REG=20: 38/37,OUT,293,OUT,293/291,OUT,293   (0.098 MSEC AT 6MHZ)
E5CC              ;E.G. REG=43: 38/37,OUT,592,OUT,592/591,OUT,      (0.197 MSEC AT 6MHZ)
E5CC
E5CC D9           SVBY:     EXX
E5CD 7A                     LD A,D
E5CE B3                     OR E
E5CF 79                     LD A,C              ;FF IF PARITY SENT, FE IF LAST JUNK BYTE SENT
E5D0                                            ;(SO PARITY LAST BIT TERMINATED CORRECTLY)
E5D0                                            ;0 IF LAST 64K BLOCK SENT
E5D0 200C                   JR NZ,SV4           ;JR IF MOD 64K NOT SAVED YET
E5D2
E5D2 3C                     INC A
E5D3 3C                     INC A
E5D4 C8                     RET Z               ;RET IF ALL BYTES SAVED - NC, Z
E5D5
E5D5 0D                     DEC C               ;DEC NUMBER OF 64K BLOCKS
E5D6 FE03                   CP 3                ;
E5D8 3004                   JR NC,SV4           ;JR IF ORIG WASN'T FF OR 0 - SOME BLOCKS LEFT
E5DA
E5DA 08                     EX AF,AF'           ;PARITY
E5DB 47                     LD B,A
E5DC 1805                   JR SV45
E5DE
E5DE 1B           SV4:      DEC DE
E5DF 46                     LD B,(HL)           ;BYTE TO SAVE
E5E0 08                     EX AF,AF'
E5E1 A8                     XOR B
E5E2 08                     EX AF,AF'           ;PARITY
E5E3
E5E3 3EF7         SV45:     LD A,0F7H
E5E5 DBF9                   IN A,(STATPORT)
E5E7 E620                   AND 20H
E5E9 37                     SCF
E5EA C8                     RET Z               ;RET IF ESC KEY HIT - CY
E5EB
E5EB 23                     INC HL
E5EC 7C                     LD A,H
E5ED FEC0                   CP 0C0H
E5EF DBFB                   IN A,(251)
E5F1 3C                     INC A               ;READY IN CASE PASSED BFFF
E5F2 3804                   JR C,SV5            ;JR IF OK (ADDR HASN'T PASSED BFFF YET)
E5F4
E5F4 2680                   LD H,80H
E5F6 D3FB                   OUT (251),A         ;NEXT PAGE
E5F8
E5F8 78           SV5:      LD A,B
E5F9 D9                     EXX
E5FA
E5FA 5F                     LD E,A              ;NEXT BYTE TO SAVE
E5FB 79                     LD A,C              ;ORIG DELAY, MINUS 2
E5FC 04                     INC B               ;B=1
E5FD D611                   SUB 17              ;COMP FOR EXTRA TIME TAKEN GETTING NEXT BYTE
E5FF 3803                   JR C,SV6            ;USE B=1 IF DELAY COUNTER CARRIED
E601
E601 3C                     INC A               ;ENSURE NOT ZERO
E602 47                     LD B,A              ;DELAY 182 Ts LESS
E603 37                     SCF                 ;SHOWS BYTE ALL SENT WHEN ROTATED OUT.
E604
E604 CB13         SV6:      RL E
E606 18C0                   JR SVDL2            ;ABOUT 187 EXTRA TO HANDLE THIS BYTE
E608
E608 CD5FE5       SABYTES:  CALL SABLK
```

```
E60B 3F                   CCF               ;NOW CY=OK
E60C 1803                 JR SVLDCOM
E60E
E60E CD5DE6     LDBYTES:  CALL LDBLK
E611
E611 FB         SVLDCOM:  EI
E612 08                   EX AF,AF'         ;NC IF ERROR
E613 3A4B5C               LD A,(BORDCOL)
E616 D3FE                 OUT (KEYPORT),A
E618 CD5D0E               CALL BRKCR        ;CHECK BREAK, STOP IF REQUIRED
E61B 08                   EX AF,AF'
E61C C9                   RET               ;C IF OK
E61D
E61D
E61D            ;LOAD CDE BYTES TO HL FROM NET. TYPE ON STACK
E61D
E61D F1         WNH:      POP AF
E61E 47                   LD B,A
E61F 08                   EX AF,AF'         ;CY IF LOAD
E620 78                   LD A,B
E621
E621 F5         WNHL:     PUSH AF
E622 CD3DE7               CALL CKNET        ;WAIT FOR FREE NET
E625
E625 CD4FE7     WTNI:     CALL NMIN         ;WAIT FOR A BYTE
E628 30FB                 JR NC,WTNI
E62A
E62A 47                   LD B,A
E62B F1                   POP AF
E62C B8                   CP B
E62D 20F2                 JR NZ,WNHL        ;WAIT FOR HDR OR DATA MARKER
E62F
E62F DD2148E6             LD IX,NISR
E633 CD6CE7               CALL NIXR
E636
E636            ;NET PARITY CHECK
E636
E636 CD4FE7               CALL NMIN
E639
E639 D2BEE2     NERR:     JP NC,TERROR
E63C
E63C D9                   EXX
E63D AC                   XOR H
E63E D9                   EXX
E63F
E63F FE01                 CP 1
E641 C9                   RET
E642
E642            ;NET OUT SR. OUTPUTS THE BYTE AT (HL), INCLUDES IN PARITY
E642
E642 7E         NOSR:     LD A,(HL)
E643 CD4DE5               CALL NMOUT
E646 1810                 JR NTVL
E648
E648            ;NET INPUT SR. LOAD OR VERIFY NET BYTE WITH (HL), INCLUDE IN PARITY
E648
E648 CD4FE7     NISR:     CALL NMIN
E64B 30EC                 JR NC,NERR
E64D
E64D 08                   EX AF,AF'
E64E 3004                 JR NC,NTV         ;JR IF VERIFYING
E650
E650 08                   EX AF,AF'
E651 77                   LD (HL),A         ;LOAD BYTE
E652 1804                 JR NTVL
E654
E654 08         NTV:      EX AF,AF'
E655 BE                   CP (HL)           ;VERIFY BYTE
E656 20E1                 JR NZ,NERR
E658
E658 D9         NTVL:     EXX
E659 AC                   XOR H
E65A 67                   LD H,A            ;PARITY BYTE
E65B D9                   EXX
E65C
E65C C9                   RET
E65D
E65D            ;ENTRY: HL=DEST TO LOAD DATA TO, CDE=LEN TO SAVE (PGFORM), A=TYPE (CHECKED VS
E65D            ;FIRST BYTE ON TAPE). 01=HEADER, FF=DATA. (IF TAPE TYPE=ZX HEADER, 17 BYTES
E65D            ;WILL BE LOADED
E65D
E65D            ;EXIT: CY IF LOADED OK, NC IF ERROR
E65D
E65D F3         LDBLK:    DI
E65E F5                   PUSH AF
E65F CD26E5               CALL TCHK
E662 20B9                 JR NZ,WNH
E664
E664 F1                   POP AF
E665 CD0420               CALL CDENORM      ;CONVERT CDE TO 19-BIT
E668 ED43C85A             LD (TEMPW1),BC    ;SAVE C (64K BLOCKS)
E66C 22CA5A               LD (TEMPW2),HL    ;DEST ADDR
E66F 0C                   INC C             ;NZ
E670 08                   EX AF,AF'         ;NZ=FIRST BYTE, NC=VERIFY, CY=LOAD
E671 3E08                 LD A,8
E673 D3FE                 OUT (KEYPORT),A   ;ENSURE MIC BIT STARTS OFF OR WE CANNOT READ EAR!
E675                                        ;INIT EDGE TYPE. BORDER WILL GO WHITE
E675
E675 CD690E     LDERR:    CALL BRKTST
E678 C8                   RET Z             ;RET IF ESC PRESSED - NC
E679
E679 0608       LDSTRT:   LD B,8
E67B CD4520               CALL EDGE2        ;LOOK FOR ALTERED EAR SIGNAL
```

```
E67E 30F5                JR NC,LDERR       ;LOOP IF TIMEOUT OR ESC
E680
E680 CD4520              CALL EDGE2
E683 30F0                JR NC,LDERR
E685
E685 D9                  EXX
E686 6F                  LD L,A
E687 0600                LD B,0
E689 D9                  EXX
E68A
E68A 2E00                LD L,0            ;LOOK AT 256 SAMPLES
E68C
E68C            ;A SIGNAL HAS BEEN FOUND - SEE IF LEADER
E68C
E68C CD4520     LDLDR:   CALL EDGE2        ;LOOK FOR TWO EDGES
E68F 30E4                JR NC,LDERR       ;JR IF BREAK OR TIMEOUT
E691
E691 D9                  EXX
E692 60                  LD H,B            ;HL=AV. B=0
E693 4D                  LD C,L            ;BC=AV
E694 29                  ADD HL,HL
E695 29                  ADD HL,HL
E696 29                  ADD HL,HL
E697 29                  ADD HL,HL         ;*16
E698 ED42                SBC HL,BC         ;*15
E69A 4F                  LD C,A            ;BC=SAMPLE
E69B 09                  ADD HL,BC         ;*15+NEW SAMPLE
E69C 29                  ADD HL,HL
E69D 29                  ADD HL,HL
E69E 29                  ADD HL,HL
E69F 29                  ADD HL,HL         ;H=NEW AV (HL/16)
E6A0 6C                  LD L,H            ;KEEP RUNNING AVERAGE OF PULSE WIDTH IN L
E6A1 7D                  LD A,L
E6A2 91                  SUB C             ;LAST PULSE WIDTH
E6A3 3002                JR NC,BLWAV
E6A5
E6A5 ED44                NEG               ;GET ABS DIFF FROM AV.
E6A7
E6A7 87         BLWAV:   ADD A,A
E6A8 87                  ADD A,A
E6A9 BD                  CP L              ;4*ABS DIFF FROM AVERAGE SHOULD BE LESS THAN AV.
E6AA D9                  EXX               ;(+/-25%)
E6AB
E6AB 30CC                JR NC,LDSTRT
E6AD
E6AD 2D                  DEC L
E6AE 20DC                JR NZ,LDLDR       ;LOOK FOR 256 LEADER PULSES
E6B0                                       ;L ENDS AS ZERO - PARITY BYTE INIT STATE.
E6B0 78                  LD A,B
E6B1 E6F8                AND 0F8H
E6B3 F602                OR 02H
E6B5 47                  LD B,A            ;RED BORDER NOW
E6B6
E6B6 D9                  EXX
E6B7 7D                  LD A,L
E6B8 CB3F                SRL A
E6BA 47                  LD B,A
E6BB CB3F                SRL A
E6BD 8D                  ADC A,L           ;A=1.25*AV LDR PULSE LEN
E6BE 5F                  LD E,A            ;E'=MAX LEN FOR 'LONG' (HI) BIT
E6BF 53                  LD D,E
E6C0 15                  DEC D
E6C1 15                  DEC D             ;FIDDLE A BIT TO CENTRALIZE..
E6C2 CB3A                SRL D             ;D'=HALF THAT - SHORT/LONG DECISION VALUE
E6C4 78                  LD A,B
E6C5 85                  ADD A,L
E6C6 ED4BC75A            LD BC,(TEMPW1-1)
E6CA 04                  INC B             ;B'=64K BLOCKS, PLUS 1
E6CB D9                  EXX
E6CC
E6CC 67                  LD H,A
E6CD CB3C                SRL H             ;0.75*AV
E6CF
E6CF            ;NOTE: WAITING FOR SYNC THIS WAY ENSURES BOTH HALVES OF ALL PULSES KEPT TOGETHER
E6CF            ;WHATEVER POLARITY OF SIGNAL, AS WELL AS SHOWING START OF DATA
E6CF
E6CF 0E00       WTSYNC:  LD C,0
E6D1 CD4C20              CALL EDGSENS
E6D4 BC                  CP H              ;CP 1.50*AV HALF-CYCLE LEN
E6D5 30A2                JR NC,LDSTRT      ;ERROR IF TOO LONG (>0.75*AV)
E6D7
E6D7 87                  ADD A,A           ;2*HALF-CYCLE
E6D8 BC                  CP H              ;CP 1.50*AV HALF-CYCLE LEN
E6D9 30F4                JR NC,WTSYNC      ;LOOP UNTIL FIRST HALF OF SYNC PULSE*2<1.50*AV
E6DB                                       ;HALF CYCLE. IE WAIT UNTIL 0.75 OR LESS OF AV.
E6DB CD4C20              CALL EDGSENS      ;C IS CARRIED IN AS START VALUE
E6DE BC                  CP H              ;A=TIME FOR WHOLE POSSIBLE SYNC PULSE
E6DF 30EE                JR NC,WTSYNC      ;INSIST WHOLE SYNC PULSE<0.5*AV WHOLE CYCLE
E6E1
E6E1 78                  LD A,B
E6E2 EE03                XOR 3
E6E4 47                  LD B,A            ;NEW BORDER COLOURS ARE BLUE AND YELLOW
E6E5 D9                  EXX
E6E6 2ACA5A              LD HL,(TEMPW2)    ;DEST PTR
E6E9 D9                  EXX
E6EA 182A                JR LDSTART        ;JUMP TO LOAD TYPE BYTE
E6EC
E6EC CB11       LDTYPE:  RL C              ;SAVE CY
E6EE 24                  INC H
E6EF 25                  DEC H
E6F0 2003                JR NZ,LDTCP       ;JR IF NOT TYPE 0 (ZX HEADER)
E6F2
E6F2 24                  INC H             ;NOW TYPE=1 (SAM HEADER)
```

```
E6F3 1E11                   LD E,17           ;LEN=ZX HEADER LEN
E6F5
E6F5 AC         LDTCP:      XOR H             ;SEE IF TYPE FLAG A' MATCHES TYPE BYTE LOADED
E6F6 C0                     RET NZ            ;RET IF NOT - NC, NZ
E6F7
E6F7 79                     LD A,C
E6F8 1F                     RRA               ;GET CY BACK. CAN'T USE RR C - CHANGES Z
E6F9 08                     EX AF,AF'
E6FA 181A                   JR LDSTART
E6FC
E6FC D9         LDLOOP:     EXX
E6FD
E6FD 08         LDDE:       EX AF,AF'
E6FE 7C                     LD A,H
E6FF D9                     EXX
E700 3803                   JR C,LDNVER
E702
E702 AE         LDVERIF:    XOR (HL)
E703 C0                     RET NZ            ;RET IF MATCH FAILED - NC,NZ
E704
E704 3E                     DB 3EH            ;"JR+1"
E705
E705 77         LDNVER:     LD (HL),A         ;STORE LOADED BYTE
E706
E706 08         LDNEXT:     EX AF,AF'         ;PROTECT FLAGS
E707 23                     INC HL            ;INC DEST PTR
E708 7C                     LD A,H
E709 FEC0                   CP 0C0H
E70B DBFB                   IN A,(251)
E70D 3C                     INC A
E70E 3804                   JR C,LDDEC        ;JR IF NOT PAST BFFFH
E710
E710 2680                   LD H,80H
E712 D3FB                   OUT (251),A       ;NEXT PAGE
E714
E714 D9         LDDEC:      EXX
E715 1B                     DEC DE            ;DEC BYTE COUNT
E716
E716 2601       LDSTART:    LD H,1            ;MARKER BIT SET IN DEST SHIFT REG
E718 0E03                   LD C,3            ;COMP FOR TIME LOST BY HANDLING BYTE
E71A
E71A CD4720     LDBITS:     CALL EDGEC
E71D D0                     RET NC            ;RET IF TIMEOUT OR SPACE PRESSED
E71E
E71E D9                     EXX
E71F BB                     CP E
E720 D0                     RET NC            ;RET IF TOO LONG
E721
E721 BA                     CP D
E722 3F                     CCF               ;NC IF SHORTER THAN 'DECISION' REGISTER
E723 D9                     EXX
E724
E724 CB14                   RL H              ;ACCUMULATE BITS IN H
E726 0E00                   LD C,0
E728 30F0                   JR NC,LDBITS      ;LOOP FOR ALL 8 BITS (TILL THE MARKER ROTATED OUT)
E72A
E72A 7D                     LD A,L            ;PARITY BYTE
E72B AC                     XOR H             ;MODIFIED
E72C 6F                     LD L,A            ;WITH NEW DATA
E72D
E72D 08                     EX AF,AF'
E72E 20BC                   JR NZ,LDTYPE      ;NO DEC B IF TYPE BYTE
E730
E730 08                     EX AF,AF'
E731 7B                     LD A,E
E732 B2                     OR D
E733 20C8                   JR NZ,LDDE        ;LOOP TO LOAD DE BYTES
E735
E735 D9                     EXX
E736 10C4                   DJNZ LDLOOP
E738 D9                     EXX
E739
E739 7D                     LD A,L            ;PARITY
E73A FE01                   CP 1              ;CY SET IF OK (ZERO)
E73C C9                     RET
E73D
E73D
E73D            ;CHECK IF NET BUSY
E73D
E73D E5         CKNET:      PUSH HL
E73E
E73E 2604       CKNT1:      LD H,4
E740
E740 CD5D0E     CKNT2:      CALL BRKCR
E743 DBFC                   IN A,(VIDPORT)
E745 17                     RLA
E746 30F6                   JR NC,CKNT1       ;LOOP TILL FREE
E748
E748 2B                     DEC HL
E749 7C                     LD A,H
E74A B5                     OR L
E74B 20F3                   JR NZ,CKNT2       ;ENSURE FREE FOR A WHILE
E74D
E74D E1                     POP HL
E74E C9                     RET
E74F
E74F
E74F            ;NET/MIDI IN. GETS BYTE IN A (CY) OR NC AND A=0 IF NO BYTE
E74F
E74F E5         NMIN:       PUSH HL
E750 212C01                 LD HL,300
E753
E753 2B         NMIN1:      DEC HL
```

```
E754 7C                  LD A,H
E755 B5                  OR L
E756 2812                JR Z,NMIN3
E758
E758 CD5D0E              CALL BRKCR
E75B DBF9                IN A,(STATPORT)
E75D CB57                BIT 2,A
E75F 20F2                JR NZ,NMIN1
E761
E761 DBF9       NMIN2:   IN A,(STATPORT)
E763 CB57                BIT 2,A
E765 28FA                JR Z,NMIN2
E767
E767 DBFD                IN A,(MDIPORT)
E769 37                  SCF
E76A
E76A E1         NMIN3:   POP HL
E76B C9                  RET
E76C
E76C
E76C D9         NIXR:    EXX
E76D 67                  LD H,A            ;PARITY BYTE
E76E D9                  EXX
E76F
E76F CBBA                RES 7,D
E771 79                  LD A,C            ;16K BLOCKS
E772 A7                  AND A
E773 280D                JR Z,NIXJ
E775
E775 F5         NIXL:    PUSH AF
E776 D5                  PUSH DE
E777 110040              LD DE,4000H       ;DO 16K
E77A CD82E7              CALL NIXJ
E77D D1                  POP DE
E77E F1                  POP AF
E77F 3D                  DEC A
E780 20F3                JR NZ,NIXL
E782
E782            ;NET SAVE/LOAD/VERIFY 0-16K BYTES (DE BYTES FROM HL)
E782
E782 7A         NIXJ:    LD A,D
E783 B3                  OR E
E784 C8                  RET Z
E785
E785 7C                  LD A,H
E786 FEC0                CP 0C0H
E788 D4F23F              CALL NC,INCURPAGE ;** BUG FIX AND RE-WRITE
E78B CD2D00              CALL IXJUMP       ;INPUT OR OUTPUT
E78E 23                  INC HL
E78F 1B                  DEC DE
E790 18F0                JR NIXJ
E792
E792                     INCLUDE USING.SAM      ;INSTR, LENGTH, STRING$, XOINTERS
E792            ;USING.SAM
E792            ;POINTERS
E792            ;ENTRY: HL=LOCN (PAGED IN), CDE=ADJUSTMENT, CY IF SUBTRACT
E792            ;EXIT: PAGCOUNT/MODCOUNT=BLOCK SIZE TO MOVE, AHL=OLD WKEND
E792
E792            ; XOI3:     LD B,3
E792             ;         LD IY,NUMENDP
E792              ;        JR X31
E792
E792 060E       XOINTERS: LD B,14          ;14 SYS VARS TO DO
E794 FD21815A             LD IY,SAVARSP    ;FIRST SYS VAR TO CONSIDER
E798
E798 08         X31:      EX AF,AF'        ;CY'=ADD/SUB FLAG
E799 ED434D5B             LD (TEMPW4),BC
E79D ED534F5B             LD (TEMPW5),DE
E7A1 EB                   EX DE,HL
E7A2 DBFB                 IN A,(251)
E7A4 E61F                 AND 1FH
E7A6 4F                   LD C,A
E7A7 CB72                 BIT 6,D
E7A9 2803                 JR Z,PNT1
E7AB
E7AB 0C                   INC C
E7AC CBB2                 RES 6,D          ;MAKE LOCN IN SECTION C
E7AE
E7AE            PNT1:    ;  PUSH BC
E7AE            ;          PUSH DE          ;ADJUSTED LOCN
E7AE 3A8D5A               LD A,(WKENDP)
E7B1 2A8E5A               LD HL,(WKEND)
E7B4 CB74                 BIT 6,H
E7B6 2803                 JR Z,PNT15
E7B8
E7B8 3C                   INC A
E7B9 CBB4                 RES 6,H
E7BB
E7BB F5         PNT15:    PUSH AF          ;SAVE ORIG WKEND (END OF BLOCK TO MOVE)
E7BC E5                   PUSH HL
E7BD DBFB                 IN A,(251)
E7BF F5                   PUSH AF
E7C0 CD1F1F               CALL ADDRNV
E7C3 E5                   PUSH HL
E7C4 CDA11E               CALL PNLP        ;DO B SYS VARS
E7C7 79                   LD A,C
E7C8 C1                   POP BC
E7C9 2A885A               LD HL,(NVARS)
E7CC A7                   AND A
E7CD ED42                 SBC HL,BC
E7CF C425E8               CALL NZ,SADJ     ;IF NVARS HAVE MOVED, PROGRAM HAS ALTERED, SO
E7D2                                       ;CHECK ADDRESSES ON GOSUB/DO/PROC STACK.
E7D2                                       ;NOTE: DELETE OF 16K EXACTLY WILL BE MISSED..
```

```
E7D2 4F                  LD C,A
E7D3 F1                  POP AF
E7D4 D3FB                OUT (251),A
E7D6 3A935A              LD A,(ELINEP)
E7D9 CD0BE8              CALL SMBW          ;MAKE WORKSP AND
E7DC                                        ;WKEND SAME BASE PAGE AS ELINE
E7DC 6F                  LD L,A
E7DD 3AB05A              LD A,(CLA+1)
E7E0 A7                  AND A
E7E1 7D                  LD A,L
E7E2 2E96                LD L,>CHADP
E7E4 CC18E8              CALL Z,SMBS        ;IF RUNNING ELINE, MAKE CHAD SAME BASE AS ELINE
E7E7
E7E7 E1                  POP HL
E7E8 F1                  POP AF             ;AHL=OLD WKEND
E7E9            ;         POP DE
E7E9            ;         POP BC             ;CDE=ORIG LOCN
E7E9 F5                  PUSH AF
E7EA E5                  PUSH HL            ;OLD WORKEND
E7EB CDE71F              CALL SUBAHLCDE     ;AHL=BLOCK SIZE TO MOVE IF MAKEROOM
E7EE 08                  EX AF,AF'
E7EF 300D                JR NC,PNBS         ;JR IF MAKEROOM
E7F1
E7F1 08                  EX AF,AF'
E7F2 ED4B4D5B            LD BC,(TEMPW4)
E7F6 ED5B4F5B            LD DE,(TEMPW5)
E7FA CDE71F              CALL SUBAHLCDE     ;SUBTRACT BLOCK SIZE IF RECLAIM
E7FD 08                  EX AF,AF'
E7FE
E7FE 08          PNBS:   EX AF,AF'
E7FF CBBC                RES 7,H
E801 32835B              LD (PAGCOUNT),A
E804 23                  INC HL             ;ONE EXTRA (MODCOUNT MAY BE 4000H NOW)
E805 22845B              LD (MODCOUNT),HL
E808 E1                  POP HL
E809 F1                  POP AF             ;AHL=OLD WKEND
E80A C9                  RET
E80B
E80B
E80B 21905A      SMBW:   LD HL,WORKSPP
E80E CD18E8              CALL SMBS
E811 2E8D                LD L,>WKENDP
E813 CD18E8              CALL SMBS
E816 2E99                LD L,>KCURP
E818
E818 F5          SMBS:   PUSH AF
E819 AE                  XOR (HL)
E81A E61F                AND 1FH
E81C 2805                JR Z,SMBS2
E81E
E81E 35                  DEC (HL)           ;IF WORKSPACE IS ON PAGE ABOVE, ADJUST
E81F                                        ;TO SAME BASE AS ELINE
E81F 23                  INC HL
E820 23                  INC HL
E821 CBF6                SET 6,(HL)
E823
E823 F1          SMBS2:  POP AF
E824 C9                  RET
E825
E825             ;STACK ADJUST
E825             ;80H=DO, 40H=PROC, 00=GOSUB
E825
E825 C5          SADJ:   PUSH BC            ;OLD (STILL CORRECT!) NVARS
E826 4F                  LD C,A             ;CDE=LOCN
E827 2AC45B              LD HL,(BSTKEND)
E82A
E82A 7E          SADJL:  LD A,(HL)          ;TYPE/PAGE
E82B FEFF                CP 0FFH
E82D 2011                JR NZ,SADJ2
E82F
E82F E1                  POP HL             ;NVARS
E830 D5                  PUSH DE
E831 ED53CC5A            LD (TEMPW3),DE
E835 51                  LD D,C
E836 08                  EX AF,AF'
E837 CD0102              CALL R1OFFCL
E83A EE1E                DW AFLPS           ;ADJUST FOR-NEXT LOOPS
E83C 08                  EX AF,AF'
E83D D1                  POP DE
E83E 79                  LD A,C
E83F C9                  RET
E840
E840 F5          SADJ2:  PUSH AF            ;SAVE B BITS 7-5 OF PAGE
E841 E5                  PUSH HL
E842 CD9C1E              CALL ASSV          ;ADJUST SINGLE "SYS VAR"
E845 E1                  POP HL
E846 F1                  POP AF
E847 E6E0                AND 0E0H
E849 B6                  OR (HL)
E84A 77                  LD (HL),A          ;KEEP TYPE BITS 7-5
E84B 23                  INC HL             ;SKIP PAGE
E84C 23                  INC HL
E84D 23                  INC HL             ;SKIP OFFSET
E84E 23                  INC HL             ;SKIP STAT
E84F E6E0                AND 0E0H           ;ISOLATE TYPE BITS
E851 FE40                CP 40H
E853 20D5                JR NZ,SADJL        ;JR IF NOT PROC
E855
E855 7E          FPTLP:  LD A,(HL)
E856 23                  INC HL
E857 B6                  OR (HL)
E858 20FB                JR NZ,FPTLP
E85A
```

```
E85A 23                   INC HL
E85B 18CD                 JR SADJL        ;LOOP WHEN PROC DOUBLE ZERO TERMINATOR FOUND
E85D
E85D
E85D
E85D            ;SEND BYTE IN 'A' TO PRINTER
E85D
E85D C5         SNDA2:    PUSH BC
E85E F5                   PUSH AF
E85F ED4B105A             LD BC,(LPTPRT1)   ;C=CONTROL PORT, B=1
E863
E863 CD5D0E     SENDLP:   CALL BRKCR        ;STOP IF BREAK PRESSED
E866 ED78                 IN A,(C)
E868 0F                   RRCA              ;TEST BIT 0
E869 38F8                 JR C,SENDLP       ;LOOP TILL NOT BUSY
E86B
E86B 0D                   DEC C
E86C F1                   POP AF
E86D ED79                 OUT (C),A         ;DATA TO DATA PORT
E86F 0C                   INC C
E870 ED41                 OUT (C),B         ;STROBE SIGNAL TO CONTROL PORT
E872 05                   DEC B
E873 ED41                 OUT (C),B         ;CANCEL STROBE PULSE (ABOUT 2.6 USECS)
E875 C1                   POP BC
E876 C9                   RET
E877
E877            ;HEAP ROOM
E877            ;ENTRY: BC=BYTES TO RESERVE (IF +VE) OR BYTES TO RELEASE (IF -VE)
E877            ;EXIT: DE=OLD HEAPEND (ROOM START). RELEASE OF >=HEAP SIZE MAKES IT EMPTY.
E877            ;E.G. LD BC,0C000H:CALL HEAPRM WILL MAKE HEAP MT.
E877            ;HL=NEW HEAPEND (ROOM+1). CAN BE CALLED WITH BC=0. NC INDICATES NOT ENOUGH
E877            ;ROOM. (HL=DEGREE OF OVERFLOW, 0+ BYTES). BC IS PRESERVED.
E877
E877 2AC85B     HEAPROOM: LD HL,(HEAPEND)   ;(4200H TO ABOUT 4A00H)
E87A 09                   ADD HL,BC
E87B 7C                   LD A,H
E87C FE40                 CP 40H
E87E 3005                 JR NC,HEAPR2      ;JR IF HEAP NOT EMPTY AFTER ADJUSTMENT
E880
E880 2ACA5B               LD HL,(HPST)      ;EMPTY HEAP
E883 1808                 JR HEAPR3
E885
E885 ED5BC45B   HEAPR2:   LD DE,(BSTKEND)
E889 ED52                 SBC HL,DE
E88B D0                   RET NC            ;RET IF NEW HEAPEND WOULD BE >=BSTKEND
E88C
E88C 19                   ADD HL,DE
E88D
E88D ED5BC85B   HEAPR3:   LD DE,(HEAPEND)
E891 22C85B               LD (HEAPEND),HL
E894 C9                   RET
E895
E895            ;USING.SAM, USING AND INARRAY/INSTRING, LENGTH, STRING$
E895            ;*****************************************************************************
E895            ;INSTRING FUNCTION - E.G. PRINT INSTR(2,S$,T$) OR INSTR(S$,T$)
E895
E895
E895 CD8B3A     IMINSTR:      CALL SINSISOBRK     ;'('
E898 CD083B                   CALL EXPTEXPR       ;EXITS WITH Z IF $, NZ IF NUM, C IF RUNNING
E89B 2008                     JR NZ,INSTR2        ;JR IF FOUND NUMBER FOR SEARCH START
E89D
E89D 3009                     JR NC,INSTR3        ;AVOID STACKING DEFAULT OF 1 IF SYNTAX CHK
E89F
E89F EF                       DB CALC             ;S$
E8A0 E9                       DB STKONE           ;S$,1
E8A1 06                       DB SWOP             ;1,S$
E8A2 33                       DB EXIT
E8A3
E8A3 1803                     JR INSTR3
E8A5
E8A5 CD9B3A     INSTR2:       CALL EXPTCSTR        ;COMMA, SEARCH$
E8A8
E8A8 CDA93A     INSTR3:       CALL EXPTCSTRB      ;COMMA, TARGET$, ')'
E8AB D0                       RET NC              ;DON'T RET HERE IF NOT RUNNING
E8AC
E8AC DBFB                     IN A,(URPORT)
E8AE F5                       PUSH AF
E8AF
E8AF            ;FETCH TARGET$, COPY TO PAGE 0 BUFFER IF LENGTH 1-255,
E8AF            ;FETCH SEARCH$. GET DE=S$ ST, BC=S$ LEN, A=T$ LEN, NC. S$ PAGED IN
E8AF
E8AF CDEA3E                   CALL SBFSR          ;COPY $ ON FPCS TO INSTBUF
E8B2                          ;INV ARG IF T$ LEN >255. A=LEN, NC ON EXIT
E8B2 F5                       PUSH AF             ;T$ LEN, NC
E8B3 CDDC3F                   CALL GETSTRING      ;S$ ST TO DE, LEN TO BC, PAGE IN
E8B6 D5                       PUSH DE             ;S$ ST
E8B7 C5                       PUSH BC             ;S$ LEN
E8B8 CD2E1D                   CALL GETINT         ;START TO BC. A=C
E8BB B0                       OR B
E8BC CA312E                   JP Z,SWER2          ;ERROR IF POSN=0
E8BF
E8BF 3A055A                   LD A,(INSTHASH)
E8C2
E8C2 D9                       EXX
E8C3 4F                       LD C,A              ;C'=HASH OR EQUIVALENT
E8C4 D9                       EXX
E8C5
E8C5 0B                       DEC  BC             ;POSN >=0
E8C6 E1                       POP HL              ;S$ LEN
E8C7 D1                       POP DE              ;S$ ST
E8C8 F1                       POP AF              ;T$ LEN
E8C9 CDD0E8                   CALL INARRAYEN
E8CC F1                       POP AF
```

```
E8CD C32FD1                 JP OSBC             ;OUT (URPORT),A: JP STACKBC
E8D0                       ;  OUT (URPORT),A
E8D0                        ; JP STACKBC
E8D0
E8D0 C5         INARRAYEN:  PUSH BC             ;POSN
E8D1 4F                     LD C,A              ;A=T$ LEN
E8D2 0600                   LD B,0              ;BC=T$ LEN
E8D4                                            ;DE=S$ START
E8D4                                            ;HL=S$ LEN
E8D4                                            ;STACK=POSN
E8D4 A7                     AND A
E8D5 CA333C                 JP Z,NOTFND2        ;NOT FOUND IF T$ LEN=0
E8D8
E8D8 ED42                   SBC HL,BC           ;SBC S$ LEN,T$ LEN
E8DA C1                     POP BC              ;POSN
E8DB 3803                   JR C,NOTFND3H       ;NOT FOUND IF TARGET$ LEN > SEARCH$ LEN
E8DD
E8DD 23                     INC HL
E8DE ED42                   SBC HL,BC           ;SUB START POSN
E8E0
E8E0 DA343C     NOTFND3H:   JP C,NOTFND3        ;JR IF START POSN TOO FAR UP SEARCH$
E8E3
E8E3 EB                     EX DE,HL
E8E4 13                     INC DE              ;DE=BYTES TO CHECK
E8E5 08                     EX AF,AF'
E8E6 09                     ADD HL,BC           ;MODIFY START-OF-SEARCH PTR NOW IN HL
E8E7 DC9D1F                 CALL C,PGOVERF      ;IF OVERFLOW, GET HL IN 8000-BFFF AREA
E8EA
E8EA CDEF3F     INSTBKLP:   CALL CHKHL          ;IF NEEDED, GET HL IN 8000-BFFF AREA
E8ED 7A                     LD A,D
E8EE FE3F                   CP 3FH
E8F0 3815                   JR C,MINSR          ;JR IF OK FOR A SEARCH WITHOUT OVERFLOW
E8F2
E8F2 D5                     PUSH DE             ;LEN >=3F00
E8F3 11FF3E                 LD DE,3EFFH
E8F6 CD07E9                 CALL MINSR          ;SEARCH 3EFF BYTES
E8F9 3017                   JR NC,JUNKS         ;IF FOUND, JUNK PREV LEN, NC. BC=POSN
E8FB
E8FB            ;            POP DE              ;JUNK PREVIOUS LEN
E8FB            ;            RET                 ;FOUND - BC=POSN FOUND, NC
E8FB
E8FB EB                     EX DE,HL            ;ELSE A'=T$ LEN
E8FC                                            ;DE=S$ PTR, HL=START POSN
E8FC 01FF3E                 LD BC,3EFFH
E8FF 09                     ADD HL,BC           ;INCR START POSN FOR NEW SEARCH
E900 E3                     EX (SP),HL          ;HL=PREV. LEN, (SP)=START POSN
E901 ED42                   SBC HL,BC
E903 EB                     EX DE,HL            ;DE=REMAINING LEN TO SEARCH, HL=S$ PTR
E904 C1                     POP BC              ;START POSN
E905 18E3                   JR INSTBKLP
E907
E907            ;ENTRY: A'=T$ LEN, HL=S$ PTR, DE=BYTES TO SEARCH, BC=START POSN
E907
E907 08         MINSR:      EX AF,AF'
E908 F5                     PUSH AF
E909 ED43475B               LD (BCSTORE),BC
E90D CD0102                 CALL R1OFFCL
E910 FE3B                   DW R0INST
E912
E912 08         JUNKS:      EX AF,AF'
E913 F1                     POP AF
E914 08                     EX AF,AF'
E915 C9                     RET
E916
E916            ;E.G. LENGTH(1,A$) OR LENGTH(2,B())
E916
E916 CD8B3A     IMLENGTH:  CALL SINSISOBRK
E919 CDE43A                CALL EXPT1NUM
E91C CD853A                CALL INSISCOMA
E91F CD0102                CALL R1OFFCL
E922 733F                  DW LENGSR        ;SETS  BIT 6,(FLAGS) FOR NUM/STR
E924 CA082D                JP Z,VNFERR      ;ERROR IF NOT FOUND IN RUNTIME
E927
E927 DF                    RST 18H
E928 CDC33A                CALL EXCBRF      ;INSIST ON A$() OR XXX()
E92B 08                    EX AF,AF'        ;CY IF RUNNING
E92C CB69                  BIT 5,C
E92E 2807                  JR Z,IMLEN2      ;JR IF NOT NUMERIC ARRAY
E930
E930 DF                    RST 18H
E931 CDC33A                CALL EXCBRF      ;INSIST ON E.G. (1,AL() )
E934 D0                    RET NC           ;RET IF SYNTAX TIME
E935
E935 1815                  JR IMLEN3
E937
E937            ;E.G. ALPHA OR ASD$
E937
E937 08         IMLEN2:    EX AF,AF'
E938 D0                    RET NC           ;RET IF SYNTAX TIME
E939
E939 F24CE9                JP P,IMLEN3      ;JR IF STRING VAR
E93C
E93C CD331D                CALL GETBYTE
E93F A7                    AND A
E940 204A                  JR NZ,IMLERR
E942
E942 2A2151                LD HL,(MEMVAL)   ;ADDR OF NUMERIC VALUE
E945
E945 3A2351     IMLENC:    LD  A,(MEMVAL+2)
E948 47                    LD B,A
E949 C32F18                JP ASBHL         ;ADJUST BHL TO REL PAGE, STACK
E94C
E94C C5         IMLEN3:    PUSH BC          ;C=T/L
```

```
E94D CD331D               CALL GETBYTE
E950 C1                   POP BC
E951 212451               LD HL,MEMVAL+3    ;PTR TO LEN DATA
E954 F5                   PUSH AF
E955
E955            ;ENTRY: C=TYPE BYTE, HL PTS TO HEADER DATA
E955            ;EXIT: HL PTS TO START OF TEXT, DE=EL LEN,
E955            ;BC=NO OF ELS. SIMPLE STRINGS HAVE EL LEN OF 1.
E955
E955 79                   LD A,C
E956 E660                 AND 60H
E958 7E                   LD A,(HL)
E959 23                   INC HL
E95A 4E                   LD C,(HL)
E95B 23                   INC HL
E95C 2009                 JR NZ,ASSAR4      ;JR IF ARRAY
E95E
E95E 0F                   RRCA              ;A=PAGES OF SIMPLE STRING LEN
E95F 0F                   RRCA
E960 B6                   OR (HL)
E961 47                   LD B,A            ;BC=STRING LEN ('EL NUMBERS')
E962
E962 110100     ASSAR3:   LD DE,1           ;DE='ELEMENT' LEN
E965 180D                 JR ASSAR5
E967
E967 23         ASSAR4:   INC HL
E968 7E                   LD A,(HL)         ;DIMS
E969 23                   INC HL
E96A 4E                   LD C,(HL)
E96B 23                   INC HL
E96C 46                   LD B,(HL)
E96D 3D                   DEC A
E96E 28F2                 JR Z,ASSAR3       ;IF DIMS=1, USE BC AS EL. NOS, 1 AS EL LEN
E970
E970 23                   INC HL            ;IF DIMS=2, DE IS LOADED WITH EL. LEN
E971 5E                   LD E,(HL)
E972 23                   INC HL
E973 56                   LD D,(HL)
E974
E974 7D         ASSAR5:   LD A,L
E975 D623                 SUB >MEMVAL+2     ;GET BYTES FROM DIMS TO TEXT
E977 2A2151               LD HL,(MEMVAL)
E97A 85                   ADD A,L
E97B 6F                   LD L,A
E97C 3001                 JR NC,ASSAR6
E97E
E97E 24                   INC H
E97F
E97F             ; HL=ARRAY ST, DE=EL. LEN, BC=EL NOS
E97F
E97F F1         ASSAR6:   POP AF
E980 A7                   AND A
E981 28C2                 JR Z,IMLENC
E983
E983 3D                   DEC A
E984 CADD1C               JP Z,STACKBC
E987
E987 EB                   EX DE,HL
E988 3D                   DEC A
E989 CAD61C               JP Z,STACKHL
E98C
E98C CF         IMLERR:   RST 08H
E98D 1E                   DB 30             ;IOOR
E98E
E98E            ;*************************************************************************
E98E            ;STRING$(N,A$)
E98E
E98E CDCB3A     IMSTRINGS: CALL EXBNCSB     ;(N,$)
E991 D0                   RET NC            ;RET IF SYNTAX TIME
E992
E992 EF                   DB CALC           ;N,$
E993 C8                   DB STOD0          ;N
E994 25                   DB DUP            ;N,N
E995 D8                   DB RCL0           ;N,N,$
E996 25                   DB DUP            ;N,N,$,$
E997 51                   DB LEN            ;N,N,$, LEN $
E998 1D                   DB SWOP23         ;N,$,N, LEN $
E999 00                   DB MULT           ;N,$,RESULT LEN
E99A 33                   DB EXIT
E99B
E99B CD2E1D               CALL GETINT
E99E B0                   OR B
E99F 2820                 JR Z,STRINGSN     ;JR IF RESULT LEN ZERO
E9A1
E9A1 78                   LD A,B
E9A2 FE02                 CP 2
E9A4 D2283A               JP NC,STLERR      ;LIMIT LEN TO <512 CHARS TO FIT BUFFER
E9A7
E9A7 C5                   PUSH BC           ;RESULT LEN
E9A8 CDE43E               CALL SBUFFET      ;COPY $ TO BUFFER, ERROR IF LEN >255. BC=LEN
E9AB C5                   PUSH BC           ;$ LEN
E9AC D5                   PUSH DE           ;BUFFER START
E9AD CD331D               CALL GETBYTE      ;N IN A
E9B0 E1                   POP HL            ;BUFFER START
E9B1 54                   LD D,H
E9B2 5D                   LD E,L
E9B3 C1                   POP BC            ;$ LEN
E9B4
E9B4 C5         STRSL:    PUSH BC
E9B5 E5                   PUSH HL
E9B6 EDB0                 LDIR              ;MAKE REPEATED COPIES FROM BUFFER START
E9B8                                        ;(FIRST ONE ON TOP OF ITSELF!)
E9B8 E1                   POP HL
```

```
E9B9 C1                    POP BC
E9BA 3D                    DEC A
E9BB 20F7                  JR NZ,STRSL
E9BD
E9BD C1                    POP BC              ;RESULT LEN
E9BE C3673C                JP CWKSTK           ;COPY RESULT TO WKSPACE AND STACK PARAMS.
E9C1
E9C1             ;NULL STRING RESULT IF EG STRING$(0,"AA") OR STRING$(10,"")
E9C1             ;BC=0
E9C1
E9C1 EF          STRINGSN:  DB CALC             ;N,$
E9C2 07                     DB DROP             ;N
E9C3 07                     DB DROP             ;
E9C4 33                     DB EXIT
E9C5
E9C5 C3DD1C                 JP STACKBC
E9C8
E9C8             ;CIRCLE/FILL SR. UNSTACKS COORDS TO B,C (NC) OR FIDDLED COORDS IN BC (CY)
E9C8             ; AND TEMPW1 IS SET UP AS OFFSET TO X COORD IF THIN PIX.
E9C8
E9C8 CD0102      CIFILSR:   CALL R1OFFCL        ;v2.6
E9CB 9327                   DW  GTFCOORDS       ;Y COORD IN B, X IN C, OR X IN HL AND CY IF THIN
E9CD 380F                   JR C,CIFISR2        ;JR IF THINPIX
E9CF
E9CF 3A745B                 LD A,(CURCMD)
E9D2 21405A                 LD HL,MODE
E9D5 86                     ADD A,(HL)
E9D6 D6ED                   SUB 0EBH+2          ;FILLTOK+2
E9D8 37                     SCF
E9D9 3F                     CCF                 ;NC
E9DA C0                     RET NZ              ;RET UNLESS: MODE 2, FATPIX, FILL
E9DB                                            ;(CUR CMD MUST BE 9A,9D OR EB)
E9DB
E9DB 69                     LD L,C
E9DC 67                     LD H,A
E9DD 29                     ADD HL,HL           ;X=X*2. (ALWAYS USE THIN PIX SYSTEM FOR FILL)
E9DE
E9DE             ;HL=X, B=Y
E9DE
E9DE 7C          CIFISR2:   LD A,H
E9DF A7                     AND A
E9E0 7D                     LD A,L
E9E1 2006                   JR NZ,THINC2
E9E3
E9E3             ;X IS <=00FF
E9E3 FE80                   CP 80H
E9E5 3806                   JR C,THINC3       ;IF X IS <80H, JR, USE L (X LSB) AS X, ADD HL OF 0
E9E7
E9E7 1809                   JR THINC4
E9E9
E9E9             ;X>=0100H
E9E9 FE80        THINC2:    CP 80H
E9EB 3805                   JR C,THINC4       ;IF X>017FH, NO JR, USE L AS X, ADD HL OF 0100H
E9ED
E9ED             ;X>017FH
E9ED 4D          THINC3:    LD C,L
E9EE 2E00                   LD L,0
E9F0 1807                   JR THINC5
E9F2
E9F2             ;X IS BETWEEN 80H AND 017FH
E9F2
E9F2 118000      THINC4:    LD DE,80H
E9F5 4B                     LD C,E              ;PRETEND CENTRE X IS 128
E9F6 A7                     AND A
E9F7 ED52                   SBC HL,DE           ;HL=0-FF=DISP TO ADD TO 80H CENTRE TO GET REAL CRD
E9F9
E9F9 22C85A      THINC5:    LD (TEMPW1),HL
E9FC 37                     SCF                 ;"THINPIX"
E9FD C9                     RET
E9FE
E9FE             ;PIXEL ROLL/SCROLL SR
E9FE
E9FE D5          CRBBFN:    PUSH DE
E9FF E5                     PUSH HL
EA00 47                     LD B,A
EA01 112C67                 LD DE,672CH         ;2ND BYTE OF RRD/INC L
EA04 0D                     DEC C
EA05 2003                   JR NZ,CRBB2         ;JR IF MOVE RIGHT
EA07
EA07 112D6F                 LD DE,6F2DH         ;2ND BYTE OF RLD/DEC L
EA0A
EA0A 3EED        CRBB2:     LD A,0EDH           ;1ST BYTE OF RLD OR RRD
EA0C 21004D                 LD HL,CDBUFF
EA0F
EA0F 77          CRBBL:     LD (HL),A
EA10 23                     INC HL
EA11 72                     LD (HL),D
EA12 23                     INC HL
EA13 73                     LD (HL),E
EA14 23                     INC HL
EA15 10F8                   DJNZ CRBBL
EA17
EA17 36C9                   LD (HL),0C9H        ;"RET"
EA19 E1                     POP HL
EA1A D1                     POP DE
EA1B C9                     RET
EA1C
EA1C C5          CRTBF:     PUSH BC
EA1D 0D                     DEC C
EA1E 0EA8                   LD C,0A8H
EA20 2004                   JR NZ,CRTB2         ;JR IF LDD WANTED
EA22
EA22 0E                     DB 0EH              ;"JR+1"
```

```
EA23
EA23 C5          CRTBFI:    PUSH BC
EA24
EA24 0EA0        CRTB1:     LD C,0A0H          ;ELSE LDI
EA26
EA26 E5          CRTB2:     PUSH HL
EA27 47                     LD B,A
EA28 3EED                   LD A,0EDH
EA2A 21004D                 LD HL,CDBUFF
EA2D
EA2D 77          CRTBL:     LD (HL),A
EA2E 23                     INC HL
EA2F 71                     LD (HL),C
EA30 23                     INC HL
EA31 10FA                   DJNZ CRTBL
EA33
EA33 36C9                   LD (HL),0C9H       ;"RET"
EA35 E1                     POP HL
EA36 C1                     POP BC
EA37 C9                     RET
EA38
EA38             ;DRCURVE
EA38
EA38             ;DRAW TO X,Y,Z. MEM0 HOLDS Z, REGS HOLD FIDDLED COORDS (NOW DISPLACEMENTS).
EA38             ;PUT COORDS BACK ON STACK, REPLACE Z.
EA38             ;ENTRY: B=Y DIFF, C (OR HL IF THIN)=X DIFF, D=SGN Y, E=SGN X (01/FF)
EA38
EA38 3A4D5A      DRTCRV:    LD A,(THFATT)
EA3B A7                     AND A              ;Z IF THIN
EA3C 7B                     LD A,E             ;SGN X
EA3D ED44                   NEG
EA3F 58                     LD E,B
EA40 D5                     PUSH DE            ;SGN Y, Y DIFF
EA41 EB                     EX DE,HL
EA42 2803                   JR Z,DRT2          ;JR IF THIN PIX - DE=X DIFF
EA44
EA44 1600                   LD D,0
EA46 59                     LD E,C             ;DE=X DIFF
EA47
EA47 2A655C      DRT2:      LD HL,(STKEND)
EA4A CDBCCB                 CALL STORADE       ;FIDDLE SGN AND PUT DE ON FPCS
EA4D 23                     INC HL
EA4E 23                     INC HL             ;PT TO NEXT NUMBER AREA ON FPCS
EA4F D1                     POP DE             ;SGN Y, Y DIFF
EA50 7A                     LD A,D
EA51 1600                   LD D,0
EA53 CDBCCB                 CALL STORADE
EA56 23                     INC HL
EA57 23                     INC HL
EA58 22655C                 LD (STKEND),HL
EA5B
EA5B EF                     DB CALC
EA5C D8                     DB RCL0            ;GET CURVATURE BACK
EA5D 33                     DB EXIT
EA5E
EA5E EF          DRCURVE:   DB CALC            ;X,Y,-Z
EA5F 5D                     DB NEGATE          ;X,Y,Z (REVERSE ANGLE SO CURVES IN RIGHT DIRECTION)
EA60 D5                     DB STO5            ;X,Y,Z (M5=Z)
EA61 E0                     DB STKHALF         ;X,Y,Z,0.5
EA62 00                     DB MULT            ;X,Y,Z/2
EA63 39                     DB SIN             ;X,Y,SIN(Z/2)
EA64 25                     DB DUP             ;X,Y,SIN(Z/2),SIN(Z/2)
EA65 5C                     DB NOT             ;X,Y,SIN(Z/2),1/0
EA66 1E                     DB JPTRUE         ;JP IF APPROX. A STRAIGHT LINE, DROPEX
EA67 18          DRHLB:     DB DROPEX-DRHLB
EA68
EA68 C8                     DB STOD0           ;X,Y  (M0=SIN(Z/2)
EA69 5D                     DB NEGATE          ;X,Y  (REVERSED BECAUSE OF INVERSE Y AXIS)
EA6A 06                     DB SWOP            ;Y,X
EA6B 25                     DB DUP             ;Y,X,X
EA6C 41                     DB ABS             ;Y,X,ABS X
EA6D 06                     DB SWOP            ;Y,ABS X,X
EA6E 1C                     DB SWOP13          ;X,ABS X,Y
EA6F 25                     DB DUP             ;X,ABS X,Y,Y
EA70 41                     DB ABS             ;X,ABS X,Y,ABS Y
EA71 1D                     DB SWOP23          ;X,Y,ABS X,ABS Y
EA72 01                     DB ADDN            ;X,Y,ABS X+ABS Y
EA73 D8                     DB RCL0            ;X,Y,ABS X+ABS Y,SIN(Z/2)
EA74 05                     DB DIVN            ;X,Y,(ABS X+ABS Y)/SIN(Z/2)
EA75 41                     DB ABS             ;X,Y,FF
EA76 D8                     DB RCL0            ;X,Y,FF,SIN(Z/2)
EA77 06                     DB SWOP            ;X,Y,SIN(Z/2),FF
EA78 25                     DB DUP             ;X,Y,SIN(Z/2),FF,FF
EA79 E6                     DB STKFONE         ;X,Y,SIN(Z/2),FF,FF,1
EA7A 03                     DB SUBN            ;X,Y,SIN(Z/2),FF,FF-1
EA7B 2F                     DB GRTE0           ;X,Y,SIN(Z/2),FF,1/0 (1 IF FF>=1)
EA7C 1E                     DB JPTRUE          ;JP IF NOT A STRAIGHT LINE, DRCURV3
EA7D 07                     DB 07H
EA7E
EA7E 07                     DB DROP            ;X,Y,SIN(Z/2)
EA7F
EA7F 07          DROPEX:    DB DROP            ;X,Y
EA80 33                     DB EXIT
EA81
EA81 C33CEB                 JP LINEDRAW
EA84
EA84 25          DRCURV3:   DB DUP             ;X,Y,SIN(Z/2),FF,FF
EA85 43                     DB SQR             ;X,Y,SIN(Z/2),FF,SQR FF
EA86 26                     DB ONELIT
EA87 02                     DB 02H
EA88 06                     DB SWOP            ;"     "     ,FF,2,SQR FF
EA89 05                     DB DIVN            ;"     "     ,FF,2/SQR FF
EA8A DD                     DB RCL5            ;"     "     ,FF,2/SQR FF,Z
```

```
EA8B 06                  DB SWOP          ;"     "     ,FF,Z,2/SQR FF
EA8C 05                  DB DIVN          ;            ,FF,Z*SQR FF/2
EA8D 41                  DB ABS           ;            ,FF,ABS(Z*SQR FF/2)
EA8E 33                  DB EXIT
EA8F
EA8F CD651D              CALL FPTOA
EA92 06FC                LD B,0FCH
EA94 3806                JR C,DRCURV4     ;JR IF TOO BIG
EA96
EA96 A0                  AND B            ;A=00-FC
EA97 C604                ADD A,4          ;A=04-00H
EA99 2801                JR Z,DRCURV4     ;JR IF TOO BIG
EA9B
EA9B 47                  LD B,A
EA9C
EA9C C5         DRCURV4: PUSH BC          ;ARCS IN B
EA9D
EA9D EF                  DB CALC          ;            ,FF
EA9E DD                  DB RCL5          ;            ,FF,Z
EA9F 23                  DB STKBREG       ;            ,FF,Z,ARCS
EAA0 05                  DB DIVN          ;            ,FF,Z/ARCS
EAA1 25                  DB DUP           ;            ,FF,Z/ARCS,Z/ARCS
EAA2 39                  DB SIN           ;            ,FF,Z/ARCS,SIN(Z/ARCS)
EAA3 CC                  DB STOD4         ;            ,FF,Z/ARCS
EAA4 25                  DB DUP           ;            ,FF,Z/ARCS,Z/ARCS
EAA5 E0                  DB STKHALF       ;            ,FF,Z/ARCS,Z/ARCS,0.5
EAA6 00                  DB MULT          ;            ,FF,Z/ARCS,Z/2*ARCS
EAA7 39                  DB SIN           ;            ,FF,Z/ARCS,SIN(Z/2*ARCS)
EAA8 D1                  DB STO1          ;            ,FF,Z/ARCS,SIN(Z/2*ARCS)
EAA9 06                  DB SWOP          ;            ,FF,SIN(Z/2*ARCS),Z/ARCS
EAAA C8                  DB STOD0         ;            ,FF,SIN(Z/2*ARCS)
EAAB 25                  DB DUP           ;            ,FF,SIN(Z/2*ARCS),SIN(Z/2*ARCS)
EAAC 00                  DB MULT          ;            ,FF,SIN(Z/2*ARCS)*SIN(Z/2*ARCS)
EAAD E0                  DB STKHALF       ;            ,FF,SIN(Z/2*ARCS)*SIN(Z/2*ARCS),0.5
EAAE 05                  DB DIVN          ;            ,FF,SIN(Z/2*ARCS)*SIN(Z/2*ARCS)*2
EAAF E6                  DB STKFONE       ;            ,FF,SIN(Z/2*ARCS)*SIN(Z/2*ARCS)*2,1
EAB0 06                  DB SWOP          ;            ,FF,1,SIN(Z/2*ARCS)*SIN(Z/2*ARCS)*2
EAB1 03                  DB SUBN          ;            ,FF,1-SIN(Z/2*ARCS)*SIN(Z/2*ARCS)*2
EAB2 CB                  DB STOD3         ;            ,FF
EAB3
EAB3 07                  DB DROP          ;X,Y,SIN(Z/2)
EAB4 D9                  DB RCL1          ;X,Y,SIN(Z/2),SIN(Z/2*ARCS)
EAB5 06                  DB SWOP          ;X,Y,SIN(Z/2*ARCS),SIN(Z/2)
EAB6 05                  DB DIVN          ;X,Y,SIN(Z/2)/SIN(Z/2*ARCS) (CALL IT V)
EAB7 C9                  DB STOD1         ;X,Y          (M1=V)
EAB8 06                  DB SWOP          ;Y,X
EAB9 25                  DB DUP           ;Y,X,X
EABA D9                  DB RCL1          ;Y,X,X,V
EABB 00                  DB MULT          ;Y,X,X*V
EABC CA                  DB STOD2         ;Y,X                (M2=X*V)
EABD 06                  DB SWOP          ;X,Y
EABE 25                  DB DUP           ;X,Y,Y
EABF D9                  DB RCL1          ;X,Y,Y,V
EAC0 00                  DB MULT          ;X,Y,Y*V
EAC1 C9                  DB STOD1         ;X,Y                (M1=Y*V)
EAC2 DD                  DB RCL5          ;X,Y,Z
EAC3 D8                  DB RCL0          ;X,Y,Z,Z/A
EAC4 03                  DB SUBN          ;X,Y,Z-Z/A
EAC5 E0                  DB STKHALF       ;X,Y,Z-Z/A,0.5
EAC6 00                  DB MULT          ;X,Y,(Z-Z/A)/2 (CALL IT T)
EAC7 25                  DB DUP           ;X,Y,T,T
EAC8 39                  DB SIN           ;X,Y,T,SIN T
EAC9 D5                  DB STO5          ;X,Y,T,SIN T    (M5=SIN T)
EACA 06                  DB SWOP          ;X,Y,SIN T,T
EACB 3A                  DB COS           ;X,Y,SIN T,COS T
EACC C8                  DB STOD0         ;X,Y,SIN T          (M0=COS T)
EACD D9                  DB RCL1          ;X,Y,SIN T,Y*V
EACE 00                  DB MULT          ;X,Y,SIN T*Y*V
EACF D8                  DB RCL0          ;X,Y,SIN T*Y*V,COS T
EAD0 DA                  DB RCL2          ;X,Y,SIN T*Y*V,COS T,X*V
EAD1 00                  DB MULT          ;X,Y,SIN T*Y*V,COS T*X*V
EAD2 01                  DB ADDN          ;X,Y,SIN T*Y*V+COS T*X*V (CALL IT P)
EAD3 D9                  DB RCL1          ;X,Y,P,Y*V
EAD4 06                  DB SWOP          ;X,Y,Y*V,P
EAD5 C9                  DB STOD1         ;X,Y,Y*V            (M1=P)
EAD6 D8                  DB RCL0          ;X,Y,Y*V,COS T
EAD7 00                  DB MULT          ;X,Y,Y*V*COS T
EAD8 DD                  DB RCL5          ;X,Y,Y*V*COS T,SIN T
EAD9 DA                  DB RCL2          ;X,Y,Y*V*COS T,SIN T,X*V
EADA 00                  DB MULT          ;X,Y,Y*V*COS T,SIN T*X*V
EADB 03                  DB SUBN          ;X,Y,Y*V*COS T-SIN T*X*V (CALL IT J)
EADC D2                  DB STO2          ;X,Y,J
EADD 41                  DB ABS           ;X,Y,ABS J
EADE D9                  DB RCL1          ;X,Y,ABS J,P
EADF 41                  DB ABS           ;X,Y,ABS J,ABS P
EAE0 01                  DB ADDN          ;X,Y,ABS J+ABS P
EAE1 07                  DB DROP          ;X,Y
EAE2 33                  DB EXIT          ;(DE)=EXP OF ABS J+ABS P
EAE3
EAE3 C1                  POP BC
EAE4 1A                  LD A,(DE)
EAE5 FE81                CP 81H
EAE7 DA3CEB              JP C,LINEDRAW
EAEA
EAEA EF                  DB CALC
EAEB 06                  DB SWOP          ;Y,X
EAEC 2A                  DB LKADDRW
EAED 425A                DW XCOORD        ;Y,X,XCOORD
EAEF D0                  DB STO0          ;Y,X,XCOORD
EAF0 01                  DB ADDN          ;Y,X+XCOORD
EAF1 06                  DB SWOP          ;X+XCOORD,Y
EAF2 29                  DB LKADDRB
EAF3 415A                DW YCOORD        ;X+XCOORD,Y,YCOORD
```

```
EAF5 D5                     DB STO5
EAF6 01                     DB ADDN             ;X+XCOORD,Y+YCOORD
EAF7 D8                     DB RCL0             ;X+XCOORD,Y+YCOORD,XCOORD
EAF8 DD                     DB RCL5             ;X+XCOORD,Y+YCOORD,XCOORD,YCOORD (END/START COORDS)
EAF9 33                     DB EXIT
EAFA
EAFA 1014                   DJNZ CUENTRY
EAFC
EAFC 182F                   JR CURVEND
EAFE
EAFE EF           CURVLP:   DB CALC
EAFF D9                     DB RCL1             ;PREV X INC
EB00 25                     DB DUP
EB01 DB                     DB RCL3
EB02 00                     DB MULT
EB03 DC                     DB RCL4
EB04 DA                     DB RCL2             ;PREV Y INC
EB05 00                     DB MULT
EB06 03                     DB SUBN
EB07 C9                     DB STOD1            ;NEW X INC
EB08 DC                     DB RCL4
EB09 00                     DB MULT
EB0A DB                     DB RCL3
EB0B DA                     DB RCL2
EB0C 00                     DB MULT
EB0D 01                     DB ADDN
EB0E CA                     DB STOD2            ;NEW Y INC
EB0F 33                     DB EXIT
EB10
EB10 EF           CUENTRY:  DB CALC
EB11 C8                     DB STOD0
EB12 D9                     DB RCL1             ;X INC
EB13 01                     DB ADDN
EB14 25                     DB DUP
EB15 2A                     DB LKADDRW
EB16 425A                   DW XCOORD           ;PEEK X COORD
EB18 03                     DB SUBN
EB19 DA                     DB RCL2             ;Y INC
EB1A D8                     DB RCL0
EB1B 01                     DB ADDN
EB1C D0                     DB STO0
EB1D 06                     DB SWOP
EB1E 29                     DB LKADDRB
EB1F 415A                   DW YCOORD           ;PEEK Y COORD
EB21 D8                     DB RCL0
EB22 03                     DB SUBN
EB23 33                     DB EXIT
EB24
EB24 C5                     PUSH BC
EB25 CD0102                 CALL R1OFFCL
EB28 9321                   DW DRAWFD           ;DRAW A LINE USING DISPS. ON FPCS, WITH XRG/YRG
EB2A C1                     POP BC              ;ADJUSTMENTS
EB2B 10D1                   DJNZ CURVLP
EB2D
EB2D EF           CURVEND:  DB CALC
EB2E 07                     DB DROP
EB2F 07                     DB DROP
EB30 06                     DB SWOP
EB31 2A                     DB LKADDRW
EB32 425A                   DW XCOORD
EB34 03                     DB SUBN
EB35 06                     DB SWOP
EB36 29                     DB LKADDRB
EB37 415A                   DW YCOORD
EB39 06                     DB SWOP
EB3A 03                     DB SUBN
EB3B 33                     DB EXIT
EB3C
EB3C CD0102       LINEDRAW: CALL R1OFFCL
EB3F 9321                   DW DRAWFD
EB41 C3C63F                 JP TRCURP
EB44
EB44                        INCLUDE MISC31.SAM      ;CALL, BORDER, ZOOM ETC,
EB44              ;MISC31.SAM -  CALL, BGRAPHICS, KEY, STRM16, NEW
EB44
EB44
EB44              ;'CALL' CMD
EB44
EB44 CDE43A       CALLER:   CALL EXPT1NUM     ;GET ADDR
EB47 0E00                   LD C,0              ;PARAMS COUNTER TO ZERO
EB49
EB49 CD793A       FCALERLP: CALL RCRC           ;RST 18H, CRCOLON
EB4C 2816                   JR Z,FCALERCE     ;EXIT LOOP IF CR OR ':'
EB4E
EB4E CD853A                 CALL INSISCOMA
EB51 C5                     PUSH BC
EB52 CD083B                 CALL EXPTEXPR     ;Z=STR, NZ=NUM, CY IF RUNNING. (BIT 7,A GAVE Z/NZ)
EB55 C1                     POP BC
EB56 0C                     INC C
EB57 30F0                   JR NC,FCALERLP
EB59
EB59                  ;       RLA               ;NC=STR, CY=NUMB
EB59                    ;     SBC A,A           ;STR=00, NUM=FF BUG**A IS CHAR BY THIS TIME
EB59                      ;   INC A             ;STR=01, NUM=00
EB59 3A3B5C                 LD A,(FLAGS)
EB5C 47                     LD B,A              ;NOW STACK EXPR AND TYPE UNDER ADDR
EB5D                                            ;C IS SAVED
EB5D
EB5D EF                     DB CALC             ;ADDR,EXPR
EB5E 06                     DB SWOP             ;EXPR,ADDR
EB5F 23                     DB STKBREG          ;EXPR,ADDR,TYPE
EB60 06                     DB SWOP             ;EXPR,TYPE,ADDR
EB61 33                     DB EXIT
```

```
EB62
EB62 18E5                   JR FCALERLP
EB64
EB64 79          FCALERCE:  LD A,C
EB65 32D05A                 LD (TEMPB3),A
EB68 CD153B                 CALL ABORTER
EB6B
EB6B CDFA01                 CALL R1OFFJP
EB6E FF17                   DW CALLX
EB70
EB70
EB70             ;CREATE ERROR, STAT, LINO
EB70
EB70 3A3A5C      SETUPVARS: LD A,(ERRNR)
EB73 119EEB                 LD DE,ERVT          ;PTR TO NAME
EB76 CD88EB                 CALL CRTVAR2
EB79 2A455C                 LD HL,(PPC)
EB7C 11A4EB                 LD DE,ERVT+6
EB7F CD8BEB                 CALL CRTVAR3
EB82 3A475C                 LD A,(SUBPPC)
EB85 11A9EB                 LD DE,ERVT+11
EB88
EB88 2600        CRTVAR2:   LD H,0
EB8A 6F                     LD L,A
EB8B
EB8B D5          CRTVAR3:   PUSH DE             ;PTR
EB8C CDD61C                 CALL STACKHL
EB8F E1                     POP HL
EB90 113F51                 LD DE,TLBYTE
EB93 7E                     LD A,(HL)           ;TLBYTE FOR SEARCH
EB94 010600                 LD BC,6             ;ENOUGH FOR 'ERROR' VAR
EB97 EDB0                   LDIR
EB99 CDFA01                 CALL R1OFFJP
EB9C 2B2B                   DW CRTVAR4
EB9E
EB9E
EB9E 046572726F ERVT:       DB 4,"error"
EBA4 036C696E6F             DB 3,"lino"
EBA9 0373746174             DB 3,"stat"
EBAE
EBAE
EBAE AF          MNINIT:    XOR A
EBAF ED47                   LD I,A
EBB1 ED56                   IM 1
EBB3
EBB3 01F800      RMPS:      LD BC,CLUTPORT
EBB6 57                     LD D,A
EBB7 87                     ADD A,A
EBB8 87                     ADD A,A
EBB9 87                     ADD A,A
EBBA ED79                   OUT (C),A           ;SET COLOUR 0
EBBC 7A                     LD A,D
EBBD D3FB                   OUT (URPORT),A
EBBF 210080                 LD HL,8000H
EBC2 110180                 LD DE,8001H
EBC5 01FF3F                 LD BC,3FFFH
EBC8 75                     LD (HL),L
EBC9 EDB0                   LDIR                ;CLEAR A PAGE
EBCB 210080                 LD HL,8000H
EBCE 04                     INC B               ;BC=0100
EBCF 1E40                   LD E,40H
EBD1
EBD1 36FF        RMCK:      LD (HL),0FFH
EBD3 56                     LD D,(HL)
EBD4 14                     INC D
EBD5 2019                   JR NZ,RAMEX         ;CHECK ALL BITS CAN BE ONES
EBD7
EBD7 72                     LD (HL),D           ;CHECK CAN BE ZEROED
EBD8 56                     LD D,(HL)
EBD9 14                     INC D
EBDA 15                     DEC D
EBDB 2013                   JR NZ,RAMEX         ;CHECK HOLDS ONE
EBDD
EBDD 09                     ADD HL,BC
EBDE 1D                     DEC E
EBDF 20F0                   JR NZ,RMCK
EBE1
EBE1 3C                     INC A
EBE2 FE20                   CP 20H
EBE4 38CD                   JR C,RMPS
EBE6
EBE6 3EFE                   LD A,0FEH
EBE8 DBFE                   IN A,(0FEH)
EBEA 1F                     RRA
EBEB 3E10                   LD A,10H            ;256K
EBED 3001                   JR NC,RAMEX         ;FORCE 256K SYSTEM IF SHIFT PRESSED
EBEF
EBEF 87                     ADD A,A             ;20H=512K
EBF0
EBF0 47          RAMEX:     LD B,A
EBF1 3D                     DEC A
EBF2 5F                     LD E,A
EBF3 32B45C                 LD (PRAMTP),A       ;PHYSICAL RAMTOP PAGE (USUALLY 0F OR 1F)
EBF6 D3FC                   OUT (VIDPORT),A     ;ANY NZ VAL STOPS SYS VARS APPEARING AS ATTR
EBF8 31004F                 LD SP,ISPVAL
EBFB 210051                 LD HL,ALLOCT
EBFE E5                     PUSH HL
EBFF
EBFF 3600        ATIF:      LD (HL),0           ;'FREE'
EC01 23                     INC HL
EC02 10FB                   DJNZ ATIF
EC04
EC04 D621                   SUB 21H
```

```
EC06 2F                     CPL             ;IF PRAMTP WAS 0F, A=11H; IF 1F, A=01
EC07 47                     LD B,A
EC08
EC08 36FF       ATIX:       LD (HL),0FFH    ;'NON-EXISTENT/TERMINATOR'
EC0A 23                     INC HL
EC0B 10FB                   DJNZ ATIX       ;FILL REST OF TABLE WITH FF'S
EC0D
EC0D E1                     POP HL
EC0E 3E40                   LD A,40H        ;'IN USE, CONTEXT 0'
EC10 0604                   LD B,4          ;PAGES TO RESERVE
EC12
EC12 77         ATIU:       LD (HL),A       ;'USED BY CONTEXT A'
EC13 23                     INC HL
EC14 10FC                   DJNZ ATIU
EC16
EC16 7D                     LD A,L
EC17 3D                     DEC A
EC18 32B05C                 LD (LASTPAGE),A ;LAST PAGE USED BY BASIC
EC1B 32B15C                 LD (RAMTOPP),A  ;RAMTOP PAGE
EC1E
EC1E 6B                     LD L,E
EC1F 2D                     DEC L           ;HL PTS TO SCREEN PAGES IN ALLOCT
EC20 7D                     LD A,L
EC21 F660                   OR 60H          ;SCREEN PAGE=LAST POSSIBLE PAIR, MODE=3, MIDI
EC23                                        ;BIT LOW (INACTIVE)
EC23 329F5C                 LD (FISCRNP),A
EC26 3EC0                   LD A,0C0H       ;'SCREEN'
EC28 77                     LD (HL),A
EC29 23                     INC HL
EC2A 77                     LD (HL),A
EC2B 21FFBF                 LD HL,0BFFFH
EC2E 22B25C                 LD (RAMTOP),HL
EC31
EC31 2169FC                 LD HL,KSRC
EC34 11E158                 LD DE,KTAB+1
EC37 0ED2                   LD C,70*3
EC39 EDB0                   LDIR            ;INIT MAIN KEYBOARD TABLES
EC3B
EC3B EB                     EX DE,HL        ;DE PTS TO TABLE OF VALUES AND DISPS TO NEXT
EC3C                                        ;DEST, ENDING IN ZERO
EC3C 21CE59                 LD HL,KTAB+238  ;DEST FOR BRIGHT CC
EC3F CD72ED                 CALL PBSL       ;B SHOULD BE ZERO ON ENTRY. EXITS UNCHANGED
EC42
EC42 21D59C                 LD HL,5CB6H+31+4000H
EC45 22A05A                 LD (PROG),HL
EC48
EC48 11B65C                 LD DE,5CB6H
EC4B ED534F5C               LD (CHANS),DE
EC4F 214AFC                 LD HL,CHANTAB
EC52 0E1F                   LD C,31
EC54 EDB0                   LDIR            ;INIT 6 CHANNELS
EC56
EC56 21DEFB                 LD HL,DKSRC
EC59 110058                 LD DE,DKBU
EC5C 0E44                   LD C,DKEN-DKSRC+1
EC5E EDB0                   LDIR            ;INIT DEF KEYS
EC60
EC60                ;       LD HL,CHIT
EC60 11005A                 LD DE,LNCUR
EC63 0E12                   LD C,18
EC65 EDB0                   LDIR            ;INIT FIRST 18 BYTES OF SYS VARS
EC67
EC67                    ;   LD HL,MAIT
EC67 11C65B                 LD DE,BASSTK
EC6A 0E1A                   LD C,26
EC6C EDB0                   LDIR            ;13 IMPORTANT ADDRESSES INITED
EC6E
EC6E 219E1C                 LD HL,NMISTOP
EC71 22E05A                 LD (NMIV),HL
EC74
EC74 214900                 LD HL,ANYI
EC77 22705B                 LD (ANYIV),HL
EC7A
EC7A CD39ED                 CALL UPACK      ;UNPACK CHAR SET
EC7D 219050                 LD HL,CHARSVAL-256
EC80 22365C                 LD (CHARS),HL
EC83 211055                 LD HL,CHARSVAL+896
EC86 227B5C                 LD (UDG),HL
EC89 1804                   JR NEW2
EC8B
EC8B CD153B     NEW:        CALL CHKEND
EC8E F3                     DI
EC8F
EC8F 21A05C     NEW2:       LD HL,SCLIST
EC92 229D5C                 LD (SCPTR),HL
EC95 2B                     DEC HL          ;PT TO FISCRNP
EC96 7E                     LD A,(HL)
EC97 32785A                 LD (CUSCRNP),A
EC9A D3FC                   OUT (VIDPORT),A
EC9C 0610                   LD B,16
EC9E
EC9E 23         SCLI:       INC HL
EC9F 77                     LD (HL),A
ECA0 3EFF                   LD A,0FFH       ;'CLOSED'
ECA2 10FA                   DJNZ SCLI
ECA4
ECA4 3AB45C                 LD A,(PRAMTP)
ECA7 3D                     DEC A
ECA8 3D                     DEC A           ;SKIP PAST SCREEN 1
ECA9 6F                     LD L,A
ECAA 2651                   LD H,ALLOCT/256
ECAC
ECAC 7E         CSPL:       LD A,(HL)
```

```
ECAD FEC0                   CP 0C0H
ECAF 2001                   JR NZ,DCSP          ;JR IF NOT A SCREEN PAGE
ECB1
ECB1 70                     LD (HL),B           ;CLOSE SCREEN PAGE
ECB2
ECB2 2D           DCSP:     DEC L
ECB3 20F7                   JR NZ,CSPL
ECB5
ECB5 4D                     LD C,L
ECB6 210C5C                 LD HL,STREAMS-4
ECB9 118B02                 LD DE,STRMTAB       ;MUST BE IN ROM0!
ECBC 0609                   LD B,9              ;INITIALISE 9 STREAMS
ECBE
ECBE 1A           STRIL:    LD A,(DE)
ECBF 13                     INC DE
ECC0 77                     LD (HL),A
ECC1 23                     INC HL
ECC2 71                     LD (HL),C
ECC3 23                     INC HL
ECC4 10F8                   DJNZ STRIL
ECC6
ECC6 0618                   LD B,24             ;12 MORE STREAM PTRS TO ZAP
ECC8
ECC8 71           CLSTL:    LD (HL),C
ECC9 23                     INC HL
ECCA 10FC                   DJNZ CLSTL
ECCC
ECCC CD321F                 CALL ADDRPROG
ECCF 36FF                   LD (HL),0FFH        ;PROGRAM TERMINATOR
ECD1 23                     INC HL
ECD2 22885A                 LD (NVARS),HL
ECD5 32875A                 LD (NVARSP),A
ECD8 32935A                 LD (ELINEP),A
ECDB CDA234                 CALL RESTOREZ
ECDE 212103                 LD HL,0321H
ECE1 22095C                 LD (REPDEL),HL      ;REPDEL=33, REPPER=3
ECE4 212151                 LD HL,MEMVAL
ECE7 22685C                 LD (MEM),HL
ECEA 7C                     LD A,H
ECEB 32385C                 LD (RASP),A
ECEE 32C65A                 LD (SPEEDIC),A      ;ANY NON-ZERO VALUE WILL DO
ECF1 324D5A                 LD (THFATT),A       ;TEMP FAT
ECF4 AF                     XOR A
ECF5 32445A                 LD (THFATP),A       ;PERM THIN
ECF8 32405A                 LD (MODE),A         ;NOT MODE 3 SO XRG NOT HALVED
ECFB CD6B39                 CALL CLRSR          ;CLEAR FPCS, BASIC STACK, NVARS, SAVARS
ECFE 2A825A                 LD HL,(SAVARS)
ED01 23                     INC HL
ED02 22945A                 LD (ELINE),HL
ED05 CD711D                 CALL SETMIN
ED08 3E03                   LD A,3
ED0A CD5A01                 CALL MODET          ;SET MODE, CSIZE, WINDOWS
ED0D CD80ED                 CALL CLSHS2
ED10 31004F                 LD SP,ISPVAL
ED13 21ED0E                 LD HL,MAINER
ED16 E5                     PUSH HL
ED17 ED733D5C               LD (ERRSP),SP
ED1B
ED1B              ;RAINBOW SCREEN
ED1B
ED1B 11D955                 LD DE,PALTAB+1
ED1E 210056                 LD HL,LINICOLS      ;L=0
ED21 45                     LD B,L              ;INIT SCAN NUMBER IN B
ED22 4D                     LD C,L
ED23
ED23 70           RBOWL:    LD (HL),B
ED24 23                     INC HL
ED25 71                     LD (HL),C           ;PAL MEM ZERO
ED26 23                     INC HL
ED27 1A                     LD A,(DE)
ED28 13                     INC DE
ED29 77                     LD (HL),A
ED2A 23                     INC HL
ED2B 77                     LD (HL),A           ;ALT COLOUR=MAIN
ED2C 23                     INC HL
ED2D 78                     LD A,B
ED2E C60B                   ADD A,11
ED30 47                     LD B,A              ;NEXT SCAN TO ALTER AT
ED31 FEA6                   CP 166
ED33 38EE                   JR C,RBOWL
ED35
ED35 36FF                   LD (HL),0FFH
ED37
ED37 CF                     RST 08H
ED38 50                     DB 50H              ;COPYRIGHT MSG
ED39
ED39
ED39              ;UPACK - CALLED BY INITIALIZATION ROUTINE
ED39              ;UPACK CHARSET FROM 5-BIT TO 8-BIT WITH 2 ZEROS AT LHS, 1 AT RHS. MAKE BOTTOM
ED39              ;SCAN ZERO EXCEPT IN 13 CASES DEALT WITH SEPARATELY.
ED39              ;CHAR DEFS. THIS ROUTINE TAKES 101 BYTES, WHEREAS A NORMAL LDIR WOULD TAKE 11.
ED39              ;NET GAIN: 402 BYTES (NOTE:OUT OF DATE NOW FOREIGN SET DROPPED!)
ED39
ED39 210763       UPACK:    LD HL,99*256+7      ;H COUNTS 99 CHARS, L COUNTS 7 BYTES/CHAR
ED3C D9                     EXX
ED3D 214EFE                 LD HL,CHARSRC
ED40 119051                 LD DE,CHARSVAL
ED43 AF                     XOR A
ED44 0605                   LD B,5              ;5 BITS TO UNPACK
ED46
ED46 4E           UPKGC:    LD C,(HL)
ED47 23                     INC HL
ED48 37                     SCF                 ;AFTER FIRST PASS, C ROTATES IN NC. WHEN C=0, LAST
```

```
ED49                                        ;VALID DATA BIT HAS GONE, CY=JUNK (SET)
ED49 CB11       UPKBL:      RL C
ED4B 28F9                   JR Z,UPKGC      ;GET NEXT DATA BYTE IF C USED UP
ED4D
ED4D 17                     RLA
ED4E 10F9                   DJNZ UPKBL      ;GET 5 BITS IN 'A' REG
ED50
ED50 07                     RLCA            ;CENTRE DATA.
ED51 12                     LD (DE),A
ED52 13                     INC DE
ED53 AF                     XOR A           ;NC, A=0
ED54 0605                   LD B,5          ;GET 5 BITS
ED56 D9                     EXX
ED57 2D                     DEC L
ED58 D9                     EXX
ED59 20EE                   JR NZ,UPKBL     ;LOOP FOR 7 BYTES/CHAR
ED5B
ED5B 12                     LD (DE),A       ;8TH BYTE IS ZERO
ED5C 13                     INC DE
ED5D
ED5D D9                     EXX
ED5E 2E07                   LD L,7
ED60 25                     DEC H
ED61 D9                     EXX
ED62 20E5                   JR NZ,UPKBL     ;LOOP FOR 137 CHARS
ED64
ED64 218954                 LD HL,95*8+CHARSVAL+1  ;PT TO SECOND BYTE OF COPYRIGHT SIGN
ED67                                        ;(FIRST OF 5 WHICH NEED BIT 6 SET)
ED67 CBF6       SB5L:       SET 6,(HL)
ED69 23                     INC HL
ED6A 10FB                   DJNZ SB5L       ;DO 5 BYTES
ED6C
ED6C            ;NOW PATCH CHARS WHERE BOTTOM SCAN IS USED
ED6C
ED6C 1133FE                 LD DE,U8TAB
ED6F 21F751                 LD HL,CHARSVAL+103 ;SCAN 7 OF COMMA
ED72
ED72            ;CALLED TO UNPACK CONTROL KEY VALUES
ED72
ED72 1A         PBSL:       LD A,(DE)       ;BYTE TO PLACE
ED73 13                     INC DE
ED74 77                     LD (HL),A       ;TO DEST
ED75 1A                     LD A,(DE)       ;DISP TO NEXT DEST, OR 0
ED76 13                     INC DE
ED77 4F                     LD C,A          ;(B=0)
ED78 09                     ADD HL,BC       ;PT HL TO NEXT DEST
ED79 A7                     AND A
ED7A 20F6                   JR NZ,PBSL      ;LOOP TILL TERMINATOR HIT
ED7C
ED7C C9                     RET
ED7D
ED7D            ;CLS #
ED7D
ED7D CD143B     CLSHS:      CALL SABORTER
ED80
ED80            ;CALLED BY NEW
ED80
ED80 CD5E11     CLSHS2:     CALL STREAMFE   ;'S'
ED83 011005                 LD BC,0510H
ED86
ED86 78         CLHSL:      LD A,B
ED87 81                     ADD A,C
ED88 D7                     RST 10H         ;PRINT CHR$ 15H/14H/13H/12H/11H
ED89 AF                     XOR A
ED8A D7                     RST 10H         ;OVER/INVERSE/BRIGHT/FLASH/PAPER 0
ED8B 10F9                   DJNZ CLHSL
ED8D
ED8D 79                     LD A,C
ED8E D7                     RST 10H         ;CHR$ 10H - PEN
ED8F 3E07                   LD A,7
ED91 D7                     RST 10H         ;PEN 7
ED92 CD9E13                 CALL PER3       ;COPY TEMP EFFECTS OF COLOUR CODES TO PERMS.
ED95                ;       CALL ZBCI
ED95                ;       JP MCLS
ED95
ED95            ;ZBCI:       XOR A
ED95 AF                     XOR A
ED96 CD3AF1                 CALL SETBORD
ED99 CDA7ED                 CALL COLINIT
ED9C C39806                 JP MCLS
ED9F
ED9F            ;COLOUR.SAM
ED9F            ;E.G. PALETTE I,C
ED9F            ;     PALETTE I,B,C
ED9F            ;     PALETTE I,C LINE L or COLOUR I,B,C LINE L
ED9F            ;     PALETTE I LINE L=DELETE CHANGE OF I AT LINE L
ED9F            ;     PALETTE   CLEARS ALL INTERRUPT CHANGES, RE-INITS PAL MEMORIES
ED9F
ED9F            ;I=LOGICAL INK NUMBER (PALETTE MEMORY) 0-15.
ED9F            ;C IS A COLOUR NUMBER FROM 0-127. B AND C ARE 2 SUCH COLOURS TO ALTERNATE UNDER
ED9F            ;INTERRUPT CONTROL. VALUES ARE SIMPLY ENTERED INTO TWO TABLES USED BY THE
ED9F            ;FRAME INTERRUPT TO SET UP THE PALETTE 50 TIMES/SEC. IF TABLES MATCH, NO
ED9F            ;INKS FLASH, OTHERWISE FLASH HAPPENS AFTER (SPEEDINK) INTS.
ED9F            ;IF 'LINE L' IS USED, THE COLOUR(S) WILL ONLY BE SET FROM START OF SPECIFIED
ED9F            ;SCAN LINE (ACTUALLY RHS BORDER OF PRECEDING LINE). LINE CAN BE 174 TO -16.
ED9F            ;=SCANS 1-191. GIVES INT AT SCANS 0-190 (ACTUALLY END OF PRECEDING LINE), AND
ED9F            ;COLOUR CHANGE AT SCANS 1-191
ED9F            ;UP TO 127 CHANGES CAN BE MADE PER SCREEN
ED9F
ED9F CD7A3A     COLOUR:     CALL CRCOLON
EDA2 2031                   JR NZ,COLOUR1           ;JR UNLESS JUST 'PALETTE'
EDA4
EDA4 CD153B                 CALL CHKEND
```

```
EDA7
EDA7 3EFF       COLINIT:     LD A,0FFH
EDA9 320056                  LD (LINICOLS),A          ;CLEAR INTERRUPT COL CHNG LIST
EDAC 11D855                  LD DE,PALTAB
EDAF CDCCED                  CALL COLINIT1            ;DO IT TWICE
EDB2 CDCCED                  CALL COLINIT1
EDB5 3A405A                  LD A,(MODE)
EDB8 FE02                    CP 2
EDBA C0                      RET NZ
EDBB
EDBB 11D855     PALSW:     LD DE,PALTAB      ;IF CHANGING TO MODE 2, SAVE NON-MODE-2 PALTAB 0-3
EDBE 21E855                LD HL,PALTAB+16   ;TO STORE, AND GET BACK MODE 2 PALTAB 0-3
EDC1 CDC7ED                CALL PL4S         ;SWOP MAIN COLOURS. EXIT WITH HL=PALTAB+20
EDC4 11FC55                LD DE,PALTAB+36
EDC7
EDC7 0604       PL4S:      LD B,4
EDC9 C39918                JP FPSWOPLP     ;SWOP ALTERNATE COLOURS
EDCC
EDCC 011400     COLINIT1:    LD BC,20
EDCF 2151FD                  LD HL,INITCOLS
EDD2 EDB0                    LDIR
EDD4 C9                      RET
EDD5
EDD5 CDE43A     COLOUR1:     CALL EXPT1NUM            ;I
EDD8 FE8C                    CP LINETOK
EDDA 2033                    JR NZ,COLOUR5
EDDC
EDDC CD573A                  CALL SSYNTAX6            ;L OF 'PALETTE I LINE L'
EDDF
EDDF CDC5EE                  CALL COLATSR
EDE2 F5                      PUSH AF                  ;LINE
EDE3 111710                  LD DE,1000H+23           ;LIMIT OF 16, INKVALERR
EDE6 CD621F                  CALL LIMBYTE             ;I
EDE9 D1                      POP DE                   ;D=LINE
EDEA 5F                      LD E,A                   ;E=I
EDEB 210056                  LD HL,LINICOLS
EDEE
EDEE 7E         COLDELP:     LD A,(HL)
EDEF 23                      INC HL
EDF0 BA                      CP D
EDF1 2806                    JR Z,COLDEL2             ;EXIT IF FOUND RIGHT SCAN ENTRY
EDF3
EDF3 D0                      RET NC                   ;RET IF PAST ANY ENTRIES FOR LINE D
EDF4
EDF4 23         COLDEL1:     INC HL
EDF5 23                      INC HL
EDF6 23                      INC HL
EDF7 18F5                    JR COLDELP
EDF9
EDF9 7E         COLDEL2:     LD A,(HL)
EDFA BB                      CP E
EDFB 20F7                    JR NZ,COLDEL1            ;JR IF WRONG I
EDFD
EDFD 54                      LD D,H                   ;ELSE DELETE ENTRY
EDFE 5D                      LD E,L
EDFF 1B                      DEC DE                   ;DE PTS TO LINE VALUE
EE00 23                      INC HL
EE01 23                      INC HL
EE02 23                      INC HL                   ;HL PTS TO LINE VALUE OF NEXT ENTRY
EE03 E5                      PUSH HL
EE04 CDB5EE                  CALL FLITD               ;GET LEN TO TABLE END
EE07 E1                      POP HL
EE08 F3                      DI
EE09 EDB0                    LDIR                     ;DELETE ENTRY. IF LINIPTR PTS BEFORE
EE0B FB                      EI                       ;DELETED ENTRY, NO PROBLEM. IF PTS TO
EE0C C9                      RET                      ;
EE0D
EE0D CF         INKVALERR:   RST 08H
EE0E 17                      DB 23                    ;'Invalid colour'
EE0F
EE0F CDDF3A     COLOUR5:     CALL EXPTCNUM            ;,C OF 'PALETTE I,C....'
EE12 FE2C                    CP ","
EE14 281B                    JR Z,COLOURFL            ;GET THIRD PARAM IF THERE IS ONE
EE16
EE16 FE8C                    CP LINETOK
EE18 2008                    JR NZ,COLSING            ;JR IF 'PALETTE I,C'
EE1A
EE1A CD573A                  CALL SSYNTAX6            ;LINE OF 'PALETTE I,C LINE L
EE1D
EE1D CDC5EE                  CALL COLATSR
EE20 1805                    JR COLOUR10
EE22
EE22 CD153B     COLSING:     CALL CHKEND
EE25
EE25 3EFF                    LD A,0FFH
EE27
EE27 F5         COLOUR10:    PUSH AF                  ;NUL LINE
EE28 111880                  LD DE,8000H+24           ;LIMIT TO <128
EE2B CD621F                  CALL LIMBYTE             ;GET SINGLE COLOUR
EE2E F5                      PUSH AF                  ;STACK IT AS THE 'SECOND' COLOUR TOO
EE2F 1820                    JR COLOUR2               ;FPCS=I, (SP)=COL, (SP+2)=LINE, A=COL
EE31
EE31 CDE33A     COLOURFL:    CALL SEXPT1NUM           ;SECOND COLOUR
EE34 FE8C                    CP LINETOK
EE36 2808                    JR Z,COLATL
EE38
EE38 CD153B                  CALL CHKEND
EE3B
EE3B 3EFF                    LD A,0FFH                ;NUL LINE
EE3D F5                      PUSH AF
EE3E 1807                    JR COLOUR15              ;FPCS=I,B,C. (SP)=LINE
EE40
EE40 CD573A     COLATL:      CALL SSYNTAX6            ;LINE
```

```
EE43
EE43 CDC5EE     COLATL2:    CALL COLATSR
EE46 F5                     PUSH AF                 ;LINE
EE47
EE47 111880     COLOUR15:   LD DE,8000H+24          ;LIMIT TO <128
EE4A CD621F                 CALL LIMBYTE            ;GET SECOND COLOUR TO A
EE4D
EE4D F5                     PUSH AF                 ;SECOND COLOUR
EE4E CD621F                 CALL LIMBYTE            ;FIRST COLOUR
EE51
EE51 F5         COLOUR2:    PUSH AF                 ;FIRST COLOUR
EE52 111710                 LD DE,1000H+23          ;LIMIT TO <16
EE55 CD621F                 CALL LIMBYTE            ;PALETTE ENTRY (I)
EE58
EE58 5F                     LD E,A                  ;E=I
EE59 C1                     POP BC                  ;B=FIRST COLOUR
EE5A F1                     POP AF
EE5B 4F                     LD C,A                  ;C=SECOND
EE5C F1                     POP AF                  ;LINE
EE5D
EE5D            ;A=LINE (FF IF NONE), B=FIRST COLOUR, C=SECOND, E=PALETTE ENTRY
EE5D
EE5D FEFF       JPALET:     CP 0FFH
EE5F 200E                   JR NZ,COLRLINE
EE61
EE61 21D855                 LD HL,PALTAB            ;PT TO PALETTE TABLE
EE64 1600                   LD D,0
EE66 19                     ADD HL,DE
EE67 70                     LD (HL),B               ;FIRST COLOUR
EE68 1E14                   LD E,20
EE6A 19                     ADD HL,DE               ;POINT TO ALTERNATE PALETTE TABLE
EE6B 71                     LD (HL),C               ;SECOND COLOUR (OFTEN THE SAME)
EE6C C9                     RET
EE6D
EE6D CF         COLFULERR:  RST 08H
EE6E 19                     DB 25                   ;'Too many palette changes'
EE6F
EE6F 210056     COLRLINE:   LD HL,LINICOLS
EE72 57                     LD D,A                  ;D=LINE
EE73
EE73 7E         COLRLP:     LD A,(HL)
EE74 23                     INC HL
EE75 BA                     CP D
EE76 3005                   JR NC,COLRL2            ;JR IF WE FOUND END (FF) OR LINE>=L
EE78
EE78 23         COLRLP2:    INC HL
EE79 23                     INC HL
EE7A 23                     INC HL
EE7B 18F6                   JR COLRLP
EE7D
EE7D 2007       COLRL2:     JR NZ,COLRL3            ;JR IF LINE HAS NO ENTRIES ALREADY
EE7F
EE7F 7E                     LD A,(HL)
EE80 BB                     CP E                    ;CP I
EE81 20F5                   JR NZ,COLRLP2           ;JR IF LINE DOES NOT CHANGE THIS I
EE83                                                ;ELSE OVERWRITE ENTRY
EE83
EE83 F3                     DI                      ;(IN CASE INTERRUPT COLOUR SWOP OCCURS)
EE84 1826                   JR LD2COL
EE86
EE86            ;WE NEED TO OPEN 4 BYTES AT (HL-1) - MIGHT BE TERMINATOR POSN
EE86
EE86 2B         COLRL3:     DEC HL
EE87 C5                     PUSH BC                 ;COLOURS
EE88 D5                     PUSH DE                 ;LINE, I
EE89 E5                     PUSH HL                 ;LINE OR TERMINATOR POSN
EE8A CDB2EE                 CALL FLITE              ;GET HL=TABLE END, BC=LEN
EE8D 03                     INC BC
EE8E 03                     INC BC
EE8F 03                     INC BC
EE90 03                     INC BC
EE91 78                     LD A,B
EE92 C6FE                   ADD A,0FEH
EE94 38D7                   JR C,COLFULERR          ;ERROR IF LIST WITH NEW ENTRY WOULD BE
EE96                                                ;MORE THAN 01FFH LONG
EE96 D1                     POP DE                  ;LOCN FOR NEW ENTRY
EE97 E5                     PUSH HL
EE98 ED52                   SBC HL,DE               ;GET TERMINATOR-LOCN
EE9A 44                     LD B,H
EE9B 4D                     LD C,L
EE9C 03                     INC BC                  ;ALLOW FOR INCLUSIVE BYTE=LEN TO MOVE
EE9D D1                     POP DE                  ;TERMINATOR
EE9E 210400                 LD HL,4
EEA1 19                     ADD HL,DE
EEA2 EB                     EX DE,HL                ;DE=TERMINATOR+4, HL=TERMINATOR
EEA3 F3                     DI
EEA4 EDB8                   LDDR
EEA6 23                     INC HL                  ;PT TO 4 BYTE SPACE
EEA7 D1                     POP DE
EEA8 C1                     POP BC
EEA9 72                     LD (HL),D               ;LINE
EEAA 23                     INC HL
EEAB 73                     LD (HL),E               ;I
EEAC
EEAC 23         LD2COL:     INC HL
EEAD 70                     LD (HL),B
EEAE 23                     INC HL
EEAF 71                     LD (HL),C
EEB0 FB                     EI
EEB1 C9                     RET
EEB2
EEB2
EEB2
```

```
EEB2                ;FIND LINE INT TABLE END. EXIT WITH BC=TABLE LEN, HL=END.
EEB2
EEB2 210056         FLITE:      LD HL,LINICOLS
EEB5
EEB5                ;FIND LINE INT TABLE DISP. EXIT WITH BC=DISP TO TERMINATOR FROM ENTRY HL,+1.
EEB5
EEB5 010100         FLITD:      LD BC,1
EEB8
EEB8 7E             FLITL:      LD A,(HL)
EEB9 3C                         INC A
EEBA C8                         RET Z
EEBB
EEBB 23                         INC HL
EEBC 23                         INC HL
EEBD 23                         INC HL
EEBE 23                         INC HL
EEBF 03                         INC BC
EEC0 03                         INC BC
EEC1 03                         INC BC
EEC2 03                         INC BC
EEC3 18F3                       JR FLITL
EEC5
EEC5                ;GET LINE FROM TOP OF FPCS AS 0-191 FROM BASIC'S 175 TO -16
EEC5
EEC5 EF             COLATSR:    DB CALC
EEC6 25                         DB DUP                  ;L,L SO COORDFID CAN BE USED
EEC7 33                         DB EXIT
EEC8
EEC8 CDFB27                     CALL COORDFID
EECB CDB027                     CALL USYCOORD           ;GETS Y AS 0 TO 191 (175 TO -16)
EECE F5                         PUSH AF                 ;LINE
EECF CD121D                     CALL FDELETE            ;JUNK DUMMY X
EED2 F1                         POP AF                  ;A=LINE
EED3 D601                       SUB 1
EED5 D0                         RET NC                  ;RET UNLESS A WAS ZERO
EED6
EED6 CF                         RST 08H
EED7 1E                         DB 30                   ;IOOR
EED8                                                    ;COLOUR CHANGE AT LINE 175 NOT POS.
EED8                                                    ;0-190 FOR ORIG 174 TO -16
EED8                                                    ;EG 1 GIVES INT AT END OF SCAN 0, COLOUR
EED8                                                    ;CHANGE AT END OF SCAN 1.
EED8
EED8                                        ;RECORD, FATPIX, CSIZE, WINDOW,
EED8                        INCLUDE MISC32.SAM  ;BEEP
EED8                ;MISC32.SAM
EED8
EED8 CD5F3A         BEEP:       CALL SYNTAX8
EEDB
EEDB EF                         DB CALC             ;LENGTH,NOTE
EEDC 26                         DB ONELIT
EEDD 1B                         DB 27               ;L,N,27
EEDE 01                         DB ADDN             ;L,N+27
EEDF 26                         DB ONELIT
EEE0 0C                         DB 12
EEE1 05                         DB DIVN             ;L,(N+27)/12
EEE2 32                         DB POWR2            ;L,2^(N+27)/12
EEE3 26                         DB ONELIT
EEE4 37                         DB 55
EEE5 00                         DB MULT             ;L,FREQUENCY
EEE6 25                         DB DUP              ;L,F,F
EEE7 1C                         DB SWOP13           ;F,F,L
EEE8 31                         DB RESTACK          ;F,F,L (FP FORM)
EEE9 33                         DB EXIT
EEEA
EEEA 7E                         LD A,(HL)           ;L EXPONENT
EEEB FE85                       CP 85H              ;**
EEED 3802                       JR C,BEEP2          ;ALLOW DURATION UP TO 16 SECS
EEEF
EEEF CF                         RST 08H
EEF0 32                         DB 50               ;'note too long'
EEF1
EEF1 01FBFF         BEEP2:      LD BC,-5
EEF4 09                         ADD HL,BC
EEF5 7E                         LD A,(HL)           ;F EXPONENT
EEF6 FE84                       CP 84H
EEF8 3804                       JR C,INVNOTE
EEFA
EEFA FE8F                       CP 8FH
EEFC 3802                       JR C,BEEP3          ;ALLOW FREQUENCIES OF 8Hz TO 16KHz.
EEFE
EEFE CF             INVNOTE:    RST 08H
EEFF 31                         DB 49               ;'Invalid note'
EF00
EF00 EF             BEEP3:      DB CALC             ;F,F,L
EF01 00                         DB MULT             ;F,F*L=CYCLES
EF02 06                         DB SWOP             ;CY,F
EF03 27                         DB FIVELIT
EF04 9337                       DB 93H,37H
EF06 1B0000                     DB 1BH,0,0          ;CY,F,375000=8T UNITS IN A 1HZ HALF-CYCLE
EF09 06                         DB SWOP             ;CY,375000,F
EF0A 05                         DB DIVN             ;F,375000/F=8T UNITS PER HALF-CYCLE
EF0B 26                         DB ONELIT
EF0C 0F                         DB 15
EF0D 03                         DB SUBN             ;ALLOW FOR TIME TAKEN BY LOOP EVEN IF NIL 8T UNITS
EF0E                                                ;(15*8=120 TS)
EF0E 33                         DB EXIT
EF0F
EF0F CD2E1D                     CALL GETINT
EF12 C5                         PUSH BC
EF13 CD2E1D                     CALL GETINT
EF16 EB                         EX DE,HL            ;DE=CYCLES (ALSO, BC=DE=BA)
EF17 E1                         POP HL              ;HL=8 T UNITS
```

```
EF18 B0                     OR B
EF19 C8                     RET Z               ;RET IF NONE TO DO
EF1A
EF1A 1B                     DEC DE              ;DE=0 GIVES 1 CYCLE
EF1B
EF1B                ;ENTRY: DE=CYCLES TO DO, MINUS 1
EF1B                ;       HL=8T UNITS PER HALF-CYCLE (ON ENTRY, H=2048 Ts, L=8 Ts)
EF1B                ;TAKES 118 TS EVEN IF HL=0, CONTENTION OF 'OUT' MAKES THAT 120 TS
EF1B
EF1B F3             BEEPP2:     DI
EF1C C5                         PUSH BC
EF1D 0130EF                     LD BC,BEEPLP
EF20 CB3D                       SRL L               ;L=16T UNITS NOW
EF22 3802                       JR C,BEEPER2        ;USE A LOOP 8 TS LONGER IF L WAS ODD
EF24
EF24 03                         INC BC
EF25 03                         INC BC
EF26
EF26 C5             BEEPER2:    PUSH BC
EF27 DDE1                       POP IX              ;IX IS LOOPING ADDR
EF29 3A4B5C                     LD A,(BORDCOL)      ;BITS=STI01GRB
EF2C                                                ;(SOFF/THRO MIDI/INTENSITY/0/MIC/GRB)
EF2C F618                       OR 18H              ;SPKR BIT HI (ON), MIC BIT HI (OFF)
EF2E 180D                       JR BPLENT           ;JUMP INTO LOOP - PULSE SPEAKER OFF
EF30
EF30 40             BEEPLP:     LD B,B
EF31 40                         LD B,B
EF32
EF32                ;ENTER HERE (IX) IF SHORTER LOOP
EF32
EF32 0C                         INC C
EF33 04                         INC B               ;INC IN CASE B OR C=0
EF34
EF34 05             BPTMLP:     DEC B
EF35 20FD                       JR NZ,BPTMLP        ;EACH INNER LOOP=16 TS
EF37
EF37 067F                       LD B,127            ;OUTER LOOP TAKES 32 TS. 127 GIVES 126*16+32=2048
EF39 0D                         DEC C
EF3A C234EF                     JP NZ,BPTMLP
EF3D
EF3D EE10           BPLENT:     XOR 10H             ;REVERSE BIT 4
EF3F D3FE                       OUT (0FEH),A        ;PULSE SPEAKER
EF41 47                         LD B,A              ;SAVE A FOR LATER
EF42 4C                         LD C,H
EF43 CB67                       BIT 4,A
EF45 2009                       JR NZ,BEEPR4        ;JR IF JUST DONE FIRST HALF-CYCLE
EF47
EF47 7A                         LD A,D
EF48 B3                         OR E
EF49 2808                       JR Z,BEEPR5         ;END IF DONE ALL CYCLES
EF4B
EF4B 1B                         DEC DE
EF4C
EF4C 78             BEEPR3:     LD A,B
EF4D 45                         LD B,L
EF4E DDE9                       JP (IX)
EF50
EF50 78             BEEPR4:     LD A,B              ;DELAY 4TS SO BOTH PATHS TAKE THE SAME TIME
EF51 18F9                       JR BEEPR3
EF53
EF53 C1             BEEPR5:     POP BC
EF54 FB                         EI
EF55 C9                         RET
EF56
EF56
EF56 CD153B         ZAP:        CALL CHKEND
EF59
EF59 0606                       LD B,6
EF5B
EF5B C5             ZPL:        PUSH BC
EF5C 010200                     LD BC,2
EF5F 1E01                       LD E,1
EF61 3E1E                       LD A,30
EF63 CD80EF                     CALL PAF
EF66 F3                         DI
EF67 C1                         POP BC
EF68 10F1                       DJNZ ZPL
EF6A
EF6A FB                         EI
EF6B C9                         RET
EF6C
EF6C
EF6C CD153B         BOOM:       CALL CHKEND
EF6F
EF6F 1E01                       LD E,1
EF71 010C00                     LD BC,12
EF74 1808                       JR ZPC
EF76
EF76 CD153B         ZOOM:       CALL CHKEND
EF79
EF79 1E06                       LD E,6
EF7B 01FDFF                     LD BC,65533
EF7E
EF7E 3E4E           ZPC:        LD A,78
EF80
EF80 2EFF           PAF:        LD L,255
EF82 F5                         PUSH AF
EF83
EF83 2600                       LD H,0
EF85 54                         LD D,H
EF86
EF86 D9             SAD:        EXX
EF87 C1                         POP BC
```

```
EF88
EF88 D9          SELP:      EXX
EF89 D5                     PUSH DE
EF8A E5                     PUSH HL
EF8B CD1BEF                 CALL BEEPP2
EF8E F3                     DI
EF8F E1                     POP HL
EF90 D1                     POP DE
EF91 09                     ADD HL,BC
EF92 D9                     EXX
EF93 10F3                   DJNZ SELP
EF95
EF95 D9                     EXX
EF96 FB                     EI
EF97 C9                     RET
EF98
EF98 CD153B      POW:       CALL CHKEND
EF9B EB                     EX DE,HL
EF9C 0600                   LD B,0
EF9E
EF9E 1A          PWLP:      LD A,(DE)
EF9F 6F                     LD L,A
EFA0 2600                   LD H,0
EFA2 13                     INC DE
EFA3 D5                     PUSH DE
EFA4 110100                 LD DE,1
EFA7 CD1BEF                 CALL BEEPP2
EFAA F3                     DI
EFAB D1                     POP DE
EFAC 10F0                   DJNZ PWLP
EFAE
EFAE FB                     EI
EFAF C9                     RET
EFB0
EFB0 CD583A      BGRAPHICS: CALL SYNTAX6
EFB3
EFB3 111E02                 LD DE,0200H+30    ;LIMIT TO <2 OR IOOR
EFB6 CD621F                 CALL LIMBYTE
EFB9
EFB9 3D                     DEC A
EFBA 32345A                 LD (BGFLG),A      ;0 IF USE BLOCKS, FF IF USE UDGS
EFBD C9                     RET               ;(BGRAPHICS 1=USE BLOCKS, 0=USE UDGS)
EFBE
EFBE             ;KEY POSN,VALUE
EFBE
EFBE CD5F3A      KEY:       CALL SYNTAX8
EFC1
EFC1 CD331D                 CALL GETBYTE
EFC4 F5                     PUSH AF
EFC5 CD2E1D                 CALL GETINT
EFC8 F1                     POP AF
EFC9 21E7FE                 LD HL,-281
EFCC 09                     ADD HL,BC
EFCD DA391D                 JP C,IOORERR      ;LIMIT POSN TO 0-280 (0 NOT USED)
EFD0
EFD0 2AD85B                 LD HL,(KBTAB)
EFD3 09                     ADD HL,BC
EFD4 77                     LD (HL),A
EFD5 C9                     RET
EFD6
EFD6
EFD6             ;SELECT SAVE/LOAD DEVICE E.G. DEVICE M: DEVICE T: DEVICE N: DEVICE M2
EFD6             ;DEVICE T45 SETS SPEED=45
EFD6
EFD6 CDEC3A      SLDEVICE:  CALL GETALPH
EFD9 E6DF                   AND 0DFH          ;FORCE UPPER CASE
EFDB F5                     PUSH AF
EFDC E7                     RST 20H
EFDD CDF73A                 CALL FETCHNUM     ;NUMBER OR USE 0
EFE0 D1                     POP DE            ;LETTER
EFE1 CD153B                 CALL CHKEND
EFE4
EFE4 D5                     PUSH DE
EFE5 CD331D                 CALL GETBYTE      ;IN C
EFE8 F1                     POP AF            ;LETTER
EFE9 FE4E                   CP "N"
EFEB 280F                   JR Z,DEVI3        ;IF NET, USE NUMBER OR DEFAULT TO 0
EFED
EFED FE54                   CP "T"
EFEF 2006                   JR NZ,DEVI2
EFF1
EFF1 0C                     INC C
EFF2 0D                     DEC C
EFF3 2007                   JR NZ,DEVI3
EFF5
EFF5 0E70                   LD C,TSPEED       ;DEFAULT TAPE SPEED
EFF7
EFF7 0C          DEVI2:     INC C
EFF8 0D                     DEC C
EFF9 2001                   JR NZ,DEVI3
EFFB
EFFB 0C                     INC C             ;DEFAULT TO 1 IF DISC
EFFC
EFFC 21065A      DEVI3:     LD HL,PSLD
EFFF 77                     LD (HL),A         ;LETTER
F000 23                     INC HL
F001 71                     LD (HL),C         ;DRIVE/STATION/SPEED
F002 C9                     RET
F003
F003
F003 CD503A      PAUSE:     CALL SYNTAX3
F006
F006 CD2E1D                 CALL GETINT
```

```
F009 1E07                LD E,7             ;BLITZ CODE FOR 'PAUSE'
F00B CD6628              CALL GRAREC        ;USE PAUSE MOD 256 AS PARAM (C)
F00E
F00E CDBD1C     PAU1:    CALL KBFLUSH       ;(HL=FLAGS)
F011 76                  HALT
F012 CD5D0E              CALL BRKCR         ;CHECK BREAK, STOP IF SO
F015 3AD15A              LD A,(LASTSTAT)    ;STATPORT VALUE ON LAST INTERRUPT
F018 E608                AND 8
F01A 20F2                JR NZ,PAU1         ;JR IF IT WASN'T A FRAME INTERRUPT
F01C
F01C 78                  LD A,B
F01D B1                  OR C
F01E 2804                JR Z,PAU2          ;DON'T DEC BC IF PARAM WAS ZERO
F020
F020 0B                  DEC BC
F021 78                  LD A,B
F022 B1                  OR C
F023 C8                  RET Z              ;RET IF COUNTED TO ZERO
F024
F024 CB6E       PAU2:    BIT 5,(HL)
F026 28E6                JR Z,PAU1          ;LOOP IF NO KEY PRESSED
F028
F028 CBAE                RES 5,(HL)         ;'NO KEY'
F02A C9                  RET
F02B
F02B
F02B 42         PRCOITEM2:    LD B,D                ;B ALSO IS PARAM
F02C D610                     SUB 16
F02E CAC6F0                   JP Z,COINK
F031
F031 3D                       DEC A
F032 CACAF0                   JP Z,COPAPER
F035
F035 3D                       DEC A
F036 286A                     JR Z,COFLASH
F038
F038 3D                       DEC A
F039 2836                     JR Z,COBRIGHT
F03B
F03B 3D                       DEC A
F03C 2012                     JR NZ,COOVER
F03E
F03E 0E04       COINVERSE:    LD C,4                ;MASK FOR BIT 2 - INVERSE
F040 7A                       LD A,D
F041 FE02                     CP 2
F043 3072                     JR NC,INVCOLERR       ;ALLOW INVERSE 0 OR 1 ONLY
F045
F045 3D                       DEC A
F046 2F                       CPL                   ;0/1 ->00/FF
F047 32545A                   LD (INVERT),A
F04A 7A                       LD A,D
F04B 07                       RLCA
F04C 07                       RLCA
F04D 47                       LD B,A                ;PARAM BIT TO BIT 2
F04E 181B                     JR COINOVC
F050
F050 0E01       COOVER:       LD C,1                ;MASK FOR BIT 0 - OVER
F052 7A                       LD A,D                ;CHECK ORIG PARAM FOR 0/1
F053 FE04                     CP 4
F055 3060                     JR NC,INVCOLERR       ;OVER 0/1/2/3 ALLOWED
F057
F057 1E04                     LD E,4                ;BLITZ CODE FOR 'OVER'
F059 C5                       PUSH BC
F05A 4F                       LD C,A
F05B F5                       PUSH AF
F05C CD6628                   CALL GRAREC
F05F F1                       POP AF
F060 C1                       POP BC
F061 32555A                   LD (GOVERT),A         ;GRAPHICS OVER IS 0-3
F064 FE02                     CP 2
F066 D0                       RET NC                ;NO FURTHER ACTION IF OVER 2 OR 3
F067
F067 32535A                   LD (OVERT),A
F06A 78                       LD A,B
F06B
F06B 21505A     COINOVC:      LD HL,PFLAGT
F06E C32AF1                   JP COCHNG             ;ADD MODIFIED PARAM TO PFLAG
F071
F071 0E40       COBRIGHT:     LD C,40H              ;MASK FOR BIT 6 - BRIGHT
F073 3A405A                   LD A,(MODE)
F076 FE02                     CP 2
F078 7A                       LD A,D                ;PARAM
F079 2824                     JR Z,COBRI2
F07B
F07B FE02                     CP 2
F07D 3020                     JR NC,COBRI2          ;JR IF BRIGHT 8
F07F
F07F A7                       AND A
F080 3A515A                   LD A,(M23PAPT)
F083 5F                       LD E,A
F084 3A525A                   LD A,(M23INKT)
F087 2808                     JR Z,COBRI1           ;JR IF BRIGHT 0
F089
F089 F688                     OR 88H                ;ADD 8 TO INK
F08B F5                       PUSH AF
F08C 7B                       LD A,E
F08D F688                     OR 88H                ;AND PAPER, IF <8
F08F 1806                     JR COBRI15
F091
F091 E677       COBRI1:       AND 77H
F093 F5                       PUSH AF
F094 7B                       LD A,E
F095 E677                     AND 77H
```

```
F097
F097 32515A     COBRI15:    LD (M23PAPT),A
F09A F1                     POP AF
F09B 32525A                 LD (M23INKT),A
F09E 7A                     LD A,D
F09F
F09F 0F         COBRI2:     RRCA                ;PARAM 0/1 ENDS IN BIT 6...
F0A0 1803                   JR COBRFLC
F0A2
F0A2 0E80       COFLASH:    LD C,80H            ;MASK FOR BIT 7 - FLASH
F0A4 7A                     LD A,D              ;PARAM
F0A5
F0A5 0F         COBRFLC:    RRCA                ;PARAM 0/1 TO BIT 7 (OR 6 IF FLASH)
F0A6 47                     LD B,A              ;MODIFIED PARAM
F0A7 7A                     LD A,D
F0A8 FE10                   CP 16
F0AA 2003                   JR NZ,COBFC2        ;ALLOW ORIG PARAM OF 0/1, OR 8/16=TRANSP
F0AC
F0AC 0F                     RRCA                ;A=8
F0AD CB08                   RRC B               ;ADJ B TOO
F0AF
F0AF FE08       COBFC2:     CP 8
F0B1 2806                   JR Z,COFBOK
F0B3
F0B3 FE02                   CP 2
F0B5 3802                   JR C,COFBOK
F0B7
F0B7 CF         INVCOLERR:  RST 08H
F0B8 17                     DB 23
F0B9
F0B9 214E5A     COFBOK:     LD HL,ATTRT
F0BC 78                     LD A,B
F0BD CD2AF1                 CALL COCHNG         ;ADD FLASH 0/1 TO ATTRT
F0C0 78                     LD A,B
F0C1 0F                     RRCA
F0C2 0F                     RRCA
F0C3 0F                     RRCA                ;FLASH/BRIGHT 8 MOVED TO CORRECT BIT FOR
F0C4 1864                   JR COCHNG           ;ADDING TO MASKT
F0C6
F0C6 0E07       COINK:      LD C,7              ;MASK FOR INK
F0C8 1807                   JR CINKPAPC
F0CA
F0CA 0E38       COPAPER:    LD C,38H            ;MASK FOR PAPER
F0CC 7A                     LD A,D
F0CD 07                     RLCA
F0CE 07                     RLCA
F0CF 07                     RLCA
F0D0 47                     LD B,A              ;GET PARAM BITS IN PLACE FOR PAPER
F0D1
F0D1 214E5A     CINKPAPC:   LD HL,ATTRT
F0D4 7A                     LD A,D              ;PARAM
F0D5 FE10                   CP 16
F0D7 3030                   JR NC,CINKPAP0
F0D9
F0D9 07                     RLCA
F0DA 07                     RLCA
F0DB 07                     RLCA
F0DC 07                     RLCA
F0DD B2                     OR D                ;A=2 COPIES OF PARAM BITS 3-0
F0DE 5F                     LD E,A              ;MODE 3 INK/PAP
F0DF 07                     RLCA
F0E0 3002                   JR NC,CONFBRI       ;JR IF NO NEED TO FORCE BRIGHT 1 (COL<8)
F0E2
F0E2 CBF6                   SET 6,(HL)          ;SET BRIGHT BIT IN ATTRT FOR COLOURS 8-15
F0E4
F0E4 3A405A     CONFBRI:    LD A,(MODE)
F0E7 FE02                   CP 2
F0E9 7B                     LD A,E
F0EA 2007                   JR NZ,COIPM3
F0EC
F0EC 07                     RLCA
F0ED 07                     RLCA                ;BITS 5,4 AND 1,0 TO 7,6 AND 4,3
F0EE AB                     XOR E
F0EF E6CC                   AND 0CCH            ;KEEP BITS 7,6,4,3 OF A
F0F1 AB                     XOR E               ;A=4 COPIES OF PARAM BITS 1,0 IF MODE 2
F0F2                                            ;WHICH ONLY HAS 4 INKS
F0F2 07                     RLCA                ;CORRECT BIT SWOP
F0F3
F0F3 CB41       COIPM3:     BIT 0,C             ;TEST MASK BIT
F0F5 280D                   JR Z,COPAPM3        ;JR IF PAPER
F0F7
F0F7 32525A                 LD (M23INKT),A
F0FA 1E05                   LD E,5              ;BLITZ CODE FOR 'PEN'
F0FC C5                     PUSH BC
F0FD 4F                     LD C,A
F0FE CD6628                 CALL GRAREC
F101 C1                     POP BC
F102 1814                   JR CINKPAP2
F104
F104 32515A     COPAPM3:    LD (M23PAPT),A
F107 180F                   JR CINKPAP2
F109
F109 FE12       CINKPAP0:   CP 18
F10B 30AA                   JR NC,INVCOLERR     ;ONLY ALLOW INK/PAPER 16 OR 17 HERE
F10D
F10D 7E                     LD   A,(HL)         ;GET (ATTRT)
F10E 2807                   JR   Z,CINKPAP1     ;JR IF INK/PAPER 16 - NO CHNG TO ATTRT,
F110                                            ;JUST MASKT
F110 B1                     OR   C              ;MAKE INK OR PAPER WHITE
F111 2F                     CPL                 ;THEN BLACK. FLIP OTHER BITS
F112 E624                   AND  24H            ;00 100 100
F114 2802                   JR Z,CINKPAP2       ;JR IF INK/PAPER WAS LIGHT
F116
```

```
F116 79                         LD   A,C               ;USE WHITE INK OR PAPER TO CONTRAST
F117
F117 47          CINKPAP1:      LD B,A
F118
F118 78          CINKPAP2:      LD A,B
F119 CD2AF1                     CALL COCHNG            ;CHANGE ATTRT
F11C 3E0F                       LD A,15                ;SO GET CARRY FOR INK/PAPER 16/17
F11E CD28F1                     CALL COCHNGP           ;CHANGE MASKT
F121 07                         RLCA
F122 07                         RLCA
F123 E650                       AND 50H
F125 4F                         LD C,A
F126 3E10                       LD A,16                ;SO GET CARRY FOR INK/PAPER 17
F128                                                   ;THEN ALTER PFLAGT
F128 BA          COCHNGP:       CP D
F129 9F                         SBC A,A
F12A
F12A AE          COCHNG:        XOR (HL)
F12B A1                         AND C
F12C AE                         XOR (HL)
F12D 77                         LD (HL),A
F12E 79                         LD A,C
F12F 23                         INC HL
F130 C9                         RET
F131
F131
F131 CD583A      BORDER:    CALL SYNTAX6
F134
F134 111710                 LD DE,1000H+23     ;LIMIT TO <16 OR INVALID COLOUR
F137 CD621F                 CALL LIMBYTE       ;A=BORDER
F13A
F13A 4F          SETBORD:   LD C,A
F13B 69                     LD L,C
F13C 07                     RLCA
F13D 07                     RLCA
F13E 47                     LD B,A
F13F A9                     XOR C
F140 E620                   AND 20H            ;USE BIT 5 FROM A
F142 A9                     XOR C
F143 F608                   OR 08H             ;MIC OFF
F145 EB                     EX DE,HL
F146 214B5C                 LD HL,BORDCOL
F149 AE                     XOR (HL)
F14A E63F                   AND 3FH            ;USE BITS 7 AND 6 (SOFF/THRO) FROM SYS VAR
F14C AE                     XOR (HL)
F14D 77                     LD (HL),A          ;USED BY SAVE/LOAD TO RESTORE BORDER, AND BEEP
F14E                                           ;BIT 7=SOFF STATUS
F14E EB                     EX DE,HL
F14F D3FE                   OUT (KEYPORT),A
F151 78                     LD A,B
F152 07                     RLCA               ;0XXX X000
F153 2600                   LD H,0             ;BLACK INK
F155 CB6F                   BIT 5,A            ;IS BORDER COLOUR LIGHT?
F157 2004                   JR NZ,BORD1        ;JR IF SO - INK IS BLACK
F159
F159 26FF                   LD H,0FFH          ;COLOUR 15/3 (WHITE) FOR LS INK (MODE 3/2)
F15B F607                   OR 07H             ;AND BORDER COLOUR
F15D
F15D 32485C      BORD1:     LD (BORDCR),A
F160 7D                     LD A,L
F161 07                     RLCA
F162 07                     RLCA
F163 07                     RLCA
F164 07                     RLCA
F165 B5                     OR L
F166 6F                     LD L,A             ;DOUBLE PAPER NIBBLE
F167 3A405A                 LD A,(MODE)
F16A FE02                   CP 2
F16C 2009                   JR NZ,BORD3
F16E
F16E CD80F1                 CALL M3TO2
F171 3C                     INC A
F172 2802                   JR Z,BORD2              ;USE INK 0 IF PAPER FF
F174
F174 3EFF                   LD A,0FFH               ;(USUALLY) WHITE INK OTHERWISE
F176
F176 67          BORD2:     LD H,A
F177
F177 22305A      BORD3:     LD (M23LSC),HL
F17A C9                     RET
F17B
F17B             ;CONVERT LS PAPER TO NON-STRIPED MODE 2 COLOUR - CALLED BY MODE
F17B
F17B CD80F1      CVLSP:     CALL M3TO2         ;CONVERT L
F17E 18F7                   JR BORD3
F180
F180 7D          M3TO2:     LD A,L
F181 07                     RLCA
F182 07                     RLCA                    ;BITS 5,4 AND 1,0 TO 7,6 AND 4,3
F183 AD                     XOR L
F184 E6CC                   AND 0CCH                ;KEEP BITS 7,6,4,3 OF A
F186 AD                     XOR L                   ;A=4 COPIES OF PARAM BITS 1,0 IF MODE 2
F187 07                     RLCA                    ;CORRECT BIT SWOP
F188 6F                     LD L,A
F189 C9                     RET
F18A
F18A
F18A CD7A3A      WINDOW:    CALL CRCOLON
F18D 283B                   JR Z,ALLWIND
F18F
F18F CDD43A                 CALL EXPT4NUMS     ;L,R,T,B
F192 CD153B                 CALL CHKEND
F195
```

```
F195 CD331D               CALL GETBYTE      ;B
F198 3A5A5A               LD A,(WINDMAX)    ;LOWEST BOTTOM
F19B 91                   SUB C
F19C 3839                 JR C,WDERR        ;ERROR IF WINDOW BOTTOM WILL BE TO LOW
F19E
F19E 41                   LD B,C
F19F C5                   PUSH BC
F1A0 CD331D               CALL GETBYTE      ;T
F1A3 C1                   POP BC            ;B=BOT
F1A4 B8                   CP B
F1A5 2802                 JR Z,WND2         ;OK IF WINDOW TOP=WINDOW BOTTOM (1 LINE HIGH)
F1A7
F1A7 302E                 JR NC,WDERR       ;ERROR IF WINDOW TOP WILL BE BELOW BOTTOM
F1A9
F1A9 4F          WND2:    LD C,A
F1AA C5                   PUSH BC           ;BOT/TOP
F1AB CD331D               CALL GETBYTE      ;R
F1AE 3A5B5A               LD A,(WINDMAX+1)  ;MAX RHS MARGIN
F1B1 B9                   CP C
F1B2 3823                 JR C,WDERR        ;ERROR IF TOO FAR RIGHT
F1B4
F1B4 C5                   PUSH BC
F1B5 CD331D               CALL GETBYTE      ;LHS
F1B8 E1                   POP HL            ;L=RHS
F1B9 BD                   CP L
F1BA 301B                 JR NC,WDERR       ;ERROR IF WINDOW IS ONE CHAR OR LESS WIDE
F1BC
F1BC 67          SETWIND: LD H,A            ;HL=LHS/RHS
F1BD 22385A               LD (UWRHS),HL
F1C0 E1                   POP HL            ;HL=BOT/TOP
F1C1 223A5A               LD (UWTOP),HL
F1C4 65                   LD H,L
F1C5 6F                   LD L,A            ;LHS
F1C6 226C5A               LD (SPOSNU),HL    ;TOP/LHS
F1C9 C9                   RET
F1CA
F1CA CD153B      ALLWIND: CALL CHKEND
F1CD
F1CD 2A5A5A               LD HL,(WINDMAX)   ;L=BOT, H=RHS
F1D0 AF                   XOR A
F1D1 55                   LD D,L
F1D2 5F                   LD E,A            ;DE=BOT/TOP
F1D3 D5                   PUSH DE
F1D4 6C                   LD L,H            ;L=RHS
F1D5 18E5                 JR SETWIND
F1D7
F1D7 CF          WDERR:   RST 08H
F1D8 36                   DB 54             ;'Invalid WINDOW'
F1D9
F1D9
F1D9
F1D9 CD5F3A      OUT:     CALL SYNTAX8
F1DC
F1DC CD331D               CALL GETBYTE
F1DF F5                   PUSH AF
F1E0 CD2E1D               CALL GETINT
F1E3 F1                   POP AF
F1E4 ED79                 OUT (C),A
F1E6 C9                   RET
F1E7
F1E7 CD153B      STOP:    CALL CHKEND
F1EA
F1EA CF                   RST 08H
F1EB 10                   DB 16             ;'STOP statement'
F1EC
F1EC
F1EC CD503A      RANDOMIZE: CALL SYNTAX3     ;NUMBER OR ZERO
F1EF
F1EF CD2E1D               CALL GETINT
F1F2 7C                   LD A,H
F1F3 B5                   OR L
F1F4 2003                 JR NZ,RANDOM1
F1F6
F1F6 2A785C               LD HL,(FRAMES)    ;RANDOMIZE OR RANDOMIZE 0 USES FRAMES
F1F9
F1F9 22765C      RANDOM1: LD (SEED),HL
F1FC C9                   RET
F1FD
F1FD                                            ;ON ERROR, FINDER, MNINIT, PALETTE, BLOCKS,
F1FD                                            ;KEY, DEVICE, PAUSE, COITEM2
F1FD                      INCLUDE SCRFN.SAM     ;COPY, SCREEN$, LIST, SPACES, EDPTR2
F1FD             ;SCRFN.SAM - TEXT/GRAPHICS COPY, SCREEN$ FUNCTION, ATTR FN, POINT, BORDER,
F1FD             ;LOOK SAM, WINDOW, ZAP ETC.
F1FD
F1FD             ;MAIN COPY ROUTINE:
F1FD
F1FD FEFF        COPY:    CP 0FFH
F1FF 2007                 JR NZ,GRCOPY
F201
F201             ;TEXT COPY
F201
F201 E7                   RST 20H           ;SKIP 0FFH
F202 FE70                 CP CHRSTOK
F204 C2290D               JP NZ,NONSENSE
F207
F207 E7                   RST 20H
F208
F208 CD153B      GRCOPY:  CALL CHKEND
F20B
F20B 3E          JGCOPY:  DB 3EH            ;'LD A,0AFH' **
F20C
F20C AF          JTCOPY:  XOR A
F20D
```

```
F20D 2ADA5A               LD HL,(DMPV)
F210
F210 24          JPOPT:   INC H
F211 25                   DEC H
F212 C8                   RET Z
F213
F213 E9                   JP (HL)           ;A=0 FOR TEXT COPY, 0AF FOR GRAPHICS
F214
F214             ;SCREEN$(L,C)
F214
F214 CDAE3A      IMSCREENS: CALL EXB2NUMB  ;CHECK (X,Y). CY IF RUNNING
F217 D0                   RET NC
F218
F218 3A385A               LD A,(UWRHS)
F21B 3C                   INC A
F21C 6F                   LD L,A            ;L=RHS LIMIT FOR COL, PLUS 1
F21D 3A3B5A               LD A,(UWBOT)
F220 C603                 ADD A,3           ;ALLOWS FOR LW OF 2 LINES
F222 67                   LD H,A            ;H=BOT LIMIT FOR LINE, PLUS 1
F223 CDA4CF               CALL GETCP        ;GET POSN IN DE
F226 CD30F2               CALL IMSCSR
F229 D2673C               JP NC,CWKSTK      ;IF FOUND, STACK 1 BYTE STRING, COPY TO WORKSP
F22C
F22C AF                   XOR A
F22D C3DA1C               JP STACKA         ;ELSE STACK NULL STRING
F230
F230             ;CALLED BY TEXT COPY AND SCREEN$ FN
F230             ;ENTRY: DE=LINE/COL
F230             ;EXIT: CY IF FOUND, (HL)=CHAR, BC=1, ELSE NC, A=0
F230
F230 21735A      IMSCSR:  LD HL,DEVICE
F233 7E                   LD A,(HL)
F234 3600                 LD (HL),0         ;ENSURE DEVICE=UPPER SCREEN, OR ANYDEADDR FAILS
F236 F5                   PUSH AF
F237 E5                   PUSH HL
F238 CDB73D               CALL ANYDEADDR    ;GET DE=SCREEN ADDR, CY IF 6-PIX CHARS IN USE,
F23B                                        ;(IN WHICH CASE NZ/Z=ODD/EVEN COL)
F23B E1                   POP HL
F23C C1                   POP BC
F23D 70                   LD (HL),B
F23E EB                   EX DE,HL
F23F 118851               LD DE,SCRNBUF     ;BUFFER WILL HOLD TARGET CHAR IN A STANDARD FORM
F242 F5                   PUSH AF
F243 3A405A               LD A,(MODE)
F246 FE02                 CP 2
F248 3822                 JR C,SCM01
F24A
F24A F1                   POP AF            ;CY IF 6-PIX, NZ/Z=ODD/EVEN COL
F24B F5                   PUSH AF
F24C CD4727               CALL CHARCOMP     ;COMPRESS SCREEN AREA TO BUFFER USING TOP LHS
F24F                                        ;PIXEL OF CHAR AS BACKGROUND
F24F F1                   POP AF
F250 3030                 JR NC,SCBMCH      ;JR IF NOT 6-PIX - BUFFER OK
F252
F252 017E08               LD BC,087EH       ;8 BYTES TO ROTATE BY 1 PIX LEFT OR RIGHT FOR
F255                                        ;CENTRAL POSITION, C=MASK TO TRIM LHS AND RHS BITS
F255 218851               LD HL,SCRNBUF
F258 2809                 JR Z,SC6EAL       ;JR IF EVEN COLUMN
F25A
F25A 7E          SC6OAL:  LD A,(HL)
F25B 07                   RLCA
F25C A1                   AND C             ;MASK OFF BITS 7 AND 0, WHICH MAY BE JUNK
F25D 77                   LD (HL),A
F25E 23                   INC HL
F25F 10F9                 DJNZ SC6OAL
F261
F261 181F                 JR SCBMCH
F263
F263 7E          SC6EAL:  LD A,(HL)
F264 0F                   RRCA
F265 A1                   AND C
F266 77                   LD (HL),A
F267 23                   INC HL
F268 10F9                 DJNZ SC6EAL
F26A
F26A 1816                 JR SCBMCH
F26C
F26C             ;MODES 0 AND 1 DO NOT NEED COMPRESSION OF SCRN CHAR DATA - BUT TRANSFER IT
F26C
F26C F1          SCM01:   POP AF            ;JUNK FLAGS FOR 6-PIX
F26D 010008               LD BC,0800H       ;BYTES TO DO/NO INVERSE
F270 CDBB3F               CALL SREAD
F273 17                   RLA
F274 3001                 JR NC,SCM01L
F276
F276 0D                   DEC C             ;MASK=FF FOR INVERTING CHAR IF TOP LH PIX=FORGROUND
F277
F277 CDBB3F      SCM01L:  CALL SREAD
F27A A9                   XOR C
F27B 12                   LD (DE),A
F27C CDF70C               CALL NXTDOWN      ;DROP HL TO NEXT SCAN
F27F 13                   INC DE            ;NEXT BUFF POSN
F280 10F5                 DJNZ SCM01L
F282
F282             ;SCREEN$ BUFFER MATCH - MATCH BUFFER WITH CHAR PATTERNS
F282
F282 2A365C      SCBMCH:  LD HL,(CHARS)
F285 24                   INC H
F286 3E60                 LD A,96           ;CHARS TO CHECK (32-127)
F288 CDA6F2               CALL SCREENSR
F28B 300F                 JR NC,SCRNFND
F28D
F28D 2A7B5C               LD HL,(UDG)       ;PTS TO CHR$ 144
```

```
F290 0180FF               LD BC,-128
F293 09                   ADD HL,BC          ;PT TO CHR$ 128
F294 3E29                 LD A,41            ;128-168
F296 CDA6F2               CALL SCREENSR
F299 D8                   RET C              ;RET IF NOT FOUND
F29A
F29A C660       UDGFND:   ADD A,96           ;1-41 -> 97-137
F29C
F29C C620       SCRNFND:  ADD A,32           ;1-96 -> 32-127, 97-137 -> 128-168
F29E 010100               LD BC,1            ;LEN=1
F2A1 21C85A               LD HL,TEMPW1
F2A4 77                   LD (HL),A          ;NC='FOUND'
F2A5 C9                   RET
F2A6
F2A6
F2A6            ;SR TO CHECK B CHARS IN CHAR. GENERATOR AT (HL) VS. CHAR IN SCRNBUF
F2A6            ;ENTRY: A=NO. OF CHARS, HL PTS TO CHARSET
F2A6            ;EXIT WITH NC IF MATCHED, A=CHAR MATCH POSN, 0 TO B-1, OR CY SET=FAIL
F2A6
F2A6 47         SCREENSR: LD B,A
F2A7 08                   EX AF,AF'          ;A'=CHARS TO CHECK (KEPT TILL END)
F2A8 118C51               LD DE,SCRNBUF+4
F2AB 1A                   LD A,(DE)
F2AC 4F                   LD C,A             ;C=BYTE 4 OF TARGET CHAR
F2AD 1B                   DEC DE
F2AE 1A                   LD A,(DE)          ;A=BYTE 3 OF TARGET CHAR
F2AF 23                   INC HL
F2B0 23                   INC HL
F2B1 23                   INC HL             ;PT HL TO BYTE 3 OF CHAR GEN. CHARACTER
F2B2 110800               LD DE,8            ;FOR STEPPING TO NEXT CHAR
F2B5 FE                   DB 0FEH            ;'JR+1'
F2B6
F2B6 19         SCREENLP: ADD HL,DE
F2B7 BE                   CP (HL)            ;CHECK BYTE 3 FOR MATCH
F2B8 2804                 JR Z,SCREEN2       ;JR TO CHECK BYTE 4 IF NEEDED
F2BA
F2BA 10FA       SCREEN1:  DJNZ SCREENLP
F2BC
F2BC 37                   SCF                ;'FAILED'
F2BD C9                   RET
F2BE
F2BE 79         SCREEN2:  LD A,C             ;BYTE 4 OF TARGET CHARACTER
F2BF 23                   INC HL
F2C0 BE                   CP (HL)            ;VS. BYTE 4 IN CHAR. GEN. CHARACTER
F2C1 2B                   DEC HL
F2C2 7E                   LD A,(HL)
F2C3 20F5                 JR NZ,SCREEN1
F2C5
F2C5 E5                   PUSH HL
F2C6 C5                   PUSH BC
F2C7 2B                   DEC HL
F2C8 2B                   DEC HL
F2C9 2B                   DEC HL             ;PT TO BYTE 0 IN CHAR. GEN CHARACTER
F2CA 118851               LD DE,SCRNBUF      ; AND BYTE 0 OF TARGET CHAR IN BUFFER
F2CD 0608                 LD B,8             ;DOING A FULL CHECK IS NOW WORTHWHILE
F2CF
F2CF 1A         FULCKLP:  LD A,(DE)
F2D0 BE                   CP (HL)
F2D1 2808                 JR Z,FULCK1        ;JR IF FULL MATCH STILL OK
F2D3
F2D3 C1                   POP BC
F2D4 E1                   POP HL
F2D5 7E                   LD A,(HL)
F2D6 110800               LD DE,8
F2D9 18DF                 JR SCREEN1
F2DB
F2DB 23         FULCK1:   INC HL
F2DC 13                   INC DE
F2DD 10F0                 DJNZ FULCKLP       ;LOOP TILL ALL 8 MATCHED OK
F2DF
F2DF C1                   POP BC
F2E0 E1                   POP HL
F2E1 08                   EX AF,AF'          ;CHARS TO CHECK
F2E2 90                   SUB B              ;FIRST CHAR MATCH GIVES 0, LAST GIVES ORIG B-1
F2E3 C9                   RET                ;WITH NC ='MATCHED'.
F2E4
F2E4            ;PT 2 OF LIST
F2E4
F2E4 CD9006     LSTR1:    CALL SPACAN
F2E7
F2E7 CD14F3     LSTLNL:   CALL OUTLINE
F2EA D8                   RET C              ;RET IF HIT END OF PROG
F2EB
F2EB D7                   RST 10H            ;PRINT 0DH
F2EC CD5602               CALL R1OFRD
F2EF 47                   LD B,A
F2F0 CDEF3F               CALL CHKHL         ;**
F2F3 23                   INC HL
F2F4 CD5602               CALL R1OFRD
F2F7 4F                   LD C,A
F2F8 2B                   DEC HL             ;BC=NUMBER OF NEXT LINE
F2F9 EB                   EX DE,HL
F2FA 2A7F5A               LD HL,(LAST)    ;LAST LINE
F2FD A7                   AND A
F2FE ED42                 SBC HL,BC
F300 EB                   EX DE,HL
F301 30E4                 JR NC,LSTLNL       ;LOOP UNTIL PAST LAST LINE ASKED FOR ('LAST LINE'=
F303                                         ;EPPC IF AUTO-LIST)
F303 3A3C5C               LD A,(TVFLAG)
F306 E610                 AND 10H
F308 C8                   RET Z              ;RET IF NOT AN AUTO-LIST, ELSE FILL REST OF SCREEN
F309
F309 3A595A               LD A,(WINDBOT)
```

```
F30C 5F                 LD E,A
F30D 3A6D5A             LD A,(SPOSNU+1)   ;PRINT POSN ROW
F310 93                 SUB E
F311 38D4               JR C,LSTLNL
F313
F313 C9                 RET
F314
F314            ;EXIT: CY=HIT END OF PROG, ELSE NC AND 0D JUST PRINTED.
F314
F314 CD5602     OUTLINE:  CALL R1OFRD       ;LN MSB
F317 47                 LD B,A
F318 C601               ADD A,1
F31A D8                 RET C             ;RET IF FF END-OF-PROG STOPPER
F31B
F31B 23                 INC HL
F31C CD5602             CALL R1OFRD
F31F 4F                 LD C,A            ;BC=CURRENT LN NO.
F320 3ABE5A             LD A,(NXTSPCS)
F323 32BC5A             LD (OLDSPCS),A    ;IN CASE LINE IS LAST ON SCREEN AND NEEDS
F326                                      ;TO BE PRINTED TWICE - ALLOWS SPACES TO BE
F326                                      ;RESET BEFORE 2ND TIME **
F326 AF                 XOR A             ;'BEFORE CURRENT LINE'
F327 EB                 EX DE,HL
F328 2A495C             LD HL,(EPPC)
F32B ED42               SBC HL,BC         ;NC IF CUR LN=EPPC OR IS LESS THAN EPPC
F32D EB                 EX DE,HL          ;NC,Z IF '>' TO BE PRINTED
F32E F5                 PUSH AF
F32F 3001               JR NC,OUTLN2      ;JR IF 'BEFORE CURRENT LINE'
F331
F331 3C                 INC A
F332
F332 32865B     OUTLN2:   LD (BCREG),A      ;1=CURRENT LINE HAS BEEN PRINTED
F335 CDB4F5             CALL PRNUMB2      ;O/P NUMBER BC WITH LEADING SPACES
F338 3A735A             LD A,(DEVICE)
F33B A7                 AND A
F33C 200F               JR NZ,OUTLN22     ;JR IF NOT UPPER SCREEN
F33E
F33E 3A585A             LD A,(WINDTOP)
F341 5F                 LD E,A
F342 3A6D5A             LD A,(SPOSNU+1)   ;LINE
F345 93                 SUB E             ;GET LINE REL. TO WINDOW
F346 C652               ADD A,>LPT
F348 5F                 LD E,A
F349 165B               LD D,LPT/256
F34B 7A                 LD A,D
F34C 12                 LD (DE),A         ;ENTRY IN LPT SHOWS LINE HAS A LINE NUMBER
F34D
F34D 23         OUTLN22:  INC HL
F34E 23                 INC HL
F34F 23                 INC HL            ;PT TO FIRST CHAR IN LINE
F350 F1                 POP AF
F351 200C               JR NZ,OUTLN3
F353
F353 CD64F5             CALL PRLCU        ;PRINT '>'
F356
F356            ;USED BY EDPRNT
F356
F356 EB         OUTLN25:  EX DE,HL
F357 213B5C             LD HL,FLAGS
F35A CBC6               SET 0,(HL)        ;'NO LEADING SPACE NOW'
F35C EB                 EX DE,HL
F35D 1803               JR OUTLN4
F35F
F35F 3E20       OUTLN3:   LD A," "
F361 D7                 RST 10H           ;PRINT SPACE (ALSO SETS 'NO LEADING SPACE')
F362
F362 AF         OUTLN4:   XOR A
F363 32C05A             LD (NXTHSPCS),A
F366 3AB85A             LD A,(LISTFLG)
F369 A7                 AND A
F36A CD5602             CALL R1OFRD       ;FIRST CHAR IN LINE
F36D E5                 PUSH HL           ;LINE PTR
F36E C411F4             CALL NZ,SPACES    ;INDENT IF FLAG SAYS 'PRETTY LISTING'
F371 E1                 POP HL
F372 EB                 EX DE,HL          ;SAVE LINE PTR IN DE
F373 AF                 XOR A
F374 32BA5A             LD (INQUFG),A     ;'NOT IN QUOTES'
F377
F377 2AA35A     OUTLNLP:  LD HL,(XPTR)
F37A A7                 AND A
F37B ED52               SBC HL,DE
F37D CC6EF5             CALL Z,PRFLQUERY  ;PRINT A FLASHING '?' IF SYNTAX ERROR PTR
F380 CD39F5             CALL OPCURSOR
F383 EB                 EX DE,HL
F384 CD0102             CALL R1OFFCL
F387 A100               DW RDCN           ;=LD A,(HL): CALL NUMBER, WITH ROM1 OFF
F389 228B5B             LD (LSPTR),HL     ;LINE SCAN PTR - USED BY CUOP O/P ROUTINE
F38C 23                 INC HL
F38D FE0D               CP 0DH
F38F C8                 RET Z
F390
F390 EB                 EX DE,HL          ;DE=LINE PTR
F391
F391 FE3A               CP ":"
F393 2041               JR NZ,OUTCH2
F395
F395 67                 LD H,A
F396 3AB85A             LD A,(LISTFLG)
F399 A7                 AND A
F39A 7C                 LD A,H
F39B 2841               JR Z,OUTCH3       ;JR IF NO PRETTY LISTING
F39D
F39D 3ABA5A             LD A,(INQUFG)
```

```
F3A0 0F                     RRCA              ;BIT 0=1 IF INSIDE
F3A1 7C                     LD A,H            ;':'
F3A2 383A                   JR C,OUTCH3       ;JR IF ':' IS INSIDE QUOTES
F3A4
F3A4 3A715C                 LD A,(FLAGX)
F3A7 E620                   AND 20H
F3A9 7C                     LD A,H
F3AA 2032                   JR NZ,OUTCH3      ;JR IF INPUT MODE
F3AC
F3AC            ;DO TAB 0 IF USING LOWER SCREEN (TO OVER-WRITE JUNK WITH SPACES) OR DO CR FOR
F3AC            ;SPEED IF NOT; FOLLOW EITHER WITH 6 SPACES TO INDENT TEXT
F3AC
F3AC 3ABD5A                 LD A,(INDOPFG)
F3AF F5                     PUSH AF
F3B0 AF                     XOR A
F3B1 32BD5A                 LD (INDOPFG),A    ;NO INDENTED O/P SO TAB FILLING LINE AVOIDS INDENT
F3B4 3A735A                 LD A,(DEVICE)
F3B7 3D                     DEC A
F3B8 3E0D                   LD A,0DH
F3BA 2006                   JR NZ,TABS2      ;JR IF NOT LOWER SCREEN
F3BC
F3BC 3E17                   LD A,17H          ;TAB
F3BE D7                     RST 10H
F3BF AF                     XOR A
F3C0 D7                     RST 10H
F3C1 AF                     XOR A
F3C2
F3C2 D7         TABS2:      RST 10H
F3C3 0606                   LD B,6
F3C5 CD76DE                 CALL OPBSP
F3C8 F1                     POP AF
F3C9 32BD5A                 LD (INDOPFG),A
F3CC
F3CC EB                     EX DE,HL
F3CD CD5602                 CALL R1OFRD
F3D0 EB                     EX DE,HL          ;A=(DE) (CHAR TO O/P NEXT)
F3D1 CD11F4                 CALL SPACES       ;SET SPACES, O/P CURRENT SPACES
F3D4 18A1                   JR OUTLNLP
F3D6
F3D6 FE22       OUTCH2:     CP 22H
F3D8 2004                   JR NZ,OUTCH3
F3DA
F3DA 21BA5A                 LD HL,INQUFG
F3DD 34                     INC (HL)          ;FLIP BIT 0 (INSIDE/OUTSIDE QUOTES FLAG)
F3DE
F3DE D7         OUTCH3:     RST 10H
F3DF 1896                   JR OUTLNLP
F3E1
F3E1
F3E1            ;SCROLL TABLE ENTRIES
F3E1            ;ENTRY: A=LINES TO SCROLL BY, D=1 IF UP, NOT 1 IF DOWN
F3E1
F3E1 6F         STENTS:     LD L,A
F3E2 3E1E                   LD A,30
F3E4 95                     SUB L
F3E5 4F                     LD C,A
F3E6 AF                     XOR A
F3E7 47                     LD B,A
F3E8 67                     LD H,A
F3E9 7D                     LD A,L
F3EA
F3EA            ;HL=LINES TO SCROLL BY, BC=30-LINES TO SCROLL BY, A=LINES TO SCROLL BY, NC
F3EA
F3EA 15                     DEC D
F3EB 2811                   JR Z,PTU          ;JR IF SCROLL UP
F3ED
F3ED 116F5B                 LD DE,LPT+29
F3F0 D5                     PUSH DE
F3F1 EB                     EX DE,HL          ;HL=LPT+29, DE=LINES
F3F2 ED52                   SBC HL,DE         ;HL=LPT+29-LINES
F3F4 D1                     POP DE            ;LPT+29
F3F5 EDB8                   LDDR
F3F7 47                     LD B,A
F3F8                                          ;(NEW LINE FOR CURSOR SET BY PRINTING OF LINE)
F3F8 AF                     XOR A
F3F9
F3F9 12         PDCL:       LD (DE),A         ;NEW ENTRIES IN TABLE ARE ZEROS
F3FA 1B                     DEC DE
F3FB 10FC                   DJNZ PDCL
F3FD
F3FD C9                     RET
F3FE
F3FE 11525B     PTU:        LD DE,LPT
F401 19                     ADD HL,DE
F402 EDB0                   LDIR
F404 47                     LD B,A
F405 218D5B                 LD HL,LNPTR
F408 7E                     LD A,(HL)
F409 90                     SUB B
F40A 77                     LD (HL),A         ;NEW LINE FOR CURSOR
F40B AF                     XOR A
F40C
F40C 12         PTCL:       LD (DE),A
F40D 13                     INC DE
F40E 10FC                   DJNZ PTCL
F410
F410 C9                     RET
F411
F411 21BE5A     SPACES:     LD HL,NXTSPCS     ;POINT TO 'NEXT-SPACES' VAR.
F414 E5                     PUSH HL
F415 46                     LD B,(HL)
F416 23                     INC HL
F417 70                     LD (HL),B         ;CURRENT SPACES=NEXT SPACES
```

```
F418 23                     INC HL
F419 46                     LD B,(HL)
F41A 23                     INC HL
F41B 70                     LD (HL),B           ;CURRENT THEN SPACES=NEXT THEN SPACES
F41C E1                     POP HL              ;PTR TO NEXT-SPACES
F41D
F41D FED4                   CP 0D4H             ;DOTOK
F41F 2843                   JR Z,INCSPCS
F421
F421 FED7                   CP 0D7H             ;LIFTOK
F423 283F                   JR Z,INCSPCS
F425
F425 FECA                   CP 0CAH             ;DEFPROCTOK
F427 283B                   JR Z,INCSPCS
F429
F429 FEC0                   CP 0C0H             ;FORTOK
F42B 2837                   JR Z,INCSPCS        ;THESE CMDS NEED NEXT SP=NEXT SP+INDENT
F42D                                            ;(INDENT FROM NEXT STATEMENT ONWARDS)
F42D FED5                   CP 0D5H             ;LOOPTOK
F42F 280C                   JR Z,DECSPCS
F431
F431 FECB                   CP 0CBH             ;ENDPROCTOK
F433 2808                   JR Z,DECSPCS
F435
F435 FEDB                   CP 0DBH             ;ENDIFTOK
F437 2804                   JR Z,DECSPCS        ;THESE CMDS (AND NEXT) NEED TO CANCEL INDENT FOR
F439                                            ;CURRENT AND LATER STATEMENTS.
F439
F439 FEC1                   CP 0C1H             ;NEXTTOK
F43B 2008                   JR NZ,SPACES2
F43D
F43D CD88F4     DECSPCS:    CALL SPACESR        ;CURRENT=CURRENT-INDENT
F440 46                     LD B,(HL)
F441 2B                     DEC HL
F442 70                     LD (HL),B           ;NEXT SPACES=CURRENT
F443 1828                   JR SPACES4
F445
F445 FED6       SPACES2:    CP 0D6H             ;EXITIFTOK
F447 280E                   JR Z,SPACES3
F449
F449 FED9                   CP 0D9H             ;LELSETOK
F44B 280A                   JR Z,SPACES3
F44D
F44D FED3                   CP 0D3H             ;LOOPIFTOK
F44F 2806                   JR Z,SPACES3        ;LOOP IF, LELSE AND EXIT IF CANCEL INDENT FOR
F451                                            ;CURRENT STAT ONLY.
F451 23                     INC HL
F452 23                     INC HL              ;PT TO SPACES THAT WILL CANCEL AT LINE END (SIF)
F453 FEDA                   CP 0DAH             ;ELSETOK
F455 2005                   JR NZ,SPACES35      ;SHORT ELSE CANCELS SOME OF THEM, AS DOES 'ON'
F457
F457 CD88F4     SPACES3:    CALL SPACESR
F45A 1811                   JR SPACES4
F45C
F45C FED8       SPACES35:   CP 0D8H             ;SIFTOK
F45E 2804                   JR Z,INCSPCS        ;SHORT IF INDENTS FOR LATER STATEMENTS, THIS LINE
F460
F460 FEDE                   CP 0DEH             ;ONTOK. AS DOES 'ON'
F462 2009                   JR NZ,SPACES4
F464
F464 3AB85A     INCSPCS:    LD A,(LISTFLG)
F467 86                     ADD A,(HL)
F468 77                     LD (HL),A
F469 FE                     DB 0FEH             ;"JR+3" (CP 3EH: LD B,FEH)
F46A
F46A            ;CALLED FROM MAIN PRINT ROUTINE WHEN LINE FULL
F46A
F46A 3E06       INDOPEN:    LD A,6
F46C FE                     DB 0FEH             ;'JR+1'
F46D
F46D AF         SPACES4:    XOR A
F46E 21BF5A                 LD HL,CURSPCS
F471 86                     ADD A,(HL)
F472 23                     INC HL
F473 23                     INC HL
F474 86                     ADD A,(HL)
F475 C8                     RET Z
F476
F476 47                     LD B,A
F477 2A565A                 LD HL,(WINDRHS)
F47A 7D                     LD A,L
F47B 94                     SUB H               ;SUB RHS,LHS=WIDTH-1
F47C FE0A                   CP 10
F47E D8                     RET C               ;NO INDENT IF 10 COLS OR LESS
F47F
F47F D605                   SUB 5
F481 B8                     CP B
F482 3001                   JR NC,SPCS5         ;FULL INDENT IF ROOM
F484
F484 47                     LD B,A              ;ELSE INDENT BY WIDTH-6
F485
F485 C376DE     SPCS5:      JP OPBSP
F488
F488 23         SPACESR:    INC HL
F489 3AB85A                 LD A,(LISTFLG)
F48C 47                     LD B,A
F48D 7E                     LD A,(HL)
F48E 90                     SUB B
F48F 77                     LD (HL),A
F490 D0                     RET NC
F491
F491 3600                   LD (HL),00H
F493 C9                     RET
```

```
F494
F494
F494
F494 CD8E11     EDPTR2:     CALL TEMPS
F497 CD893D                 CALL POFETCH
F49A D5                     PUSH DE             ;SAVE SCRN POSN SO THAT SEVERAL EDPRTS CAN ALL
F49B                                            ;START AT THE SAME POSN, NOT CONCATENATED!
F49B 3A585A                 LD A,(WINDTOP)
F49E 32CF5A                 LD (TEMPB2),A
F4A1 213C5C                 LD HL,TVFLAG
F4A4 CBAE                   RES 5,(HL)          ;'NO NEED TO CLEAR LS ON KEYSTROKE'
F4A6 CB9E                   RES 3,(HL)          ;'NO NEED TO COPY LINE TO LOWER SCRN'
F4A8 2A3D5C                 LD HL,(ERRSP)
F4AB E5                     PUSH HL
F4AC 21F7F4                 LD HL,EDPE          ;NEW ERROR HANDLER
F4AF E5                     PUSH HL
F4B0 ED733D5C               LD (ERRSP),SP
F4B4 CDC204                 CALL SETDE          ;DE=START OF ELINE OR INPUT LINE
F4B7 2A825C                 LD HL,(OLDPOS)
F4BA E5                     PUSH HL             ;POSN OF END OF LINE IN LOWER SCREEN
F4BB EB                     EX DE,HL            ;HL=LINE START
F4BC 0156F3                 LD BC,OUTLN25
F4BF 3A715C                 LD A,(FLAGX)
F4C2 E620                   AND 20H
F4C4 200C                   JR NZ,EDIM          ;JR IF INPUT MODE - NO INDENT
F4C6
F4C6 7E                     LD A,(HL)
F4C7 FE0D                   CP 0DH
F4C9 280A                   JR Z,EDCOP          ;JR IF LINE EMPTY ** BUG FIX
F4CB
F4CB CD9006                 CALL SPACAN
F4CE CDC508                 CALL IOPCL          ;PRINT THE LINE, INDENTED
F4D1 BF                     CP A                ;Z
F4D2
F4D2 C41600     EDIM:       CALL NZ,BCJUMP
F4D5
F4D5 EB         EDCOP:      EX DE,HL            ;DE=LINE PTR
F4D6 CD39F5                 CALL OPCURSOR       ;PRINT THE CURSOR IF IT IS AT THE END OF THE LINE
F4D9                                            ;(OR IT HAS BEEN PRINTED ALREADY)
F4D9 CD8E11                 CALL TEMPS
F4DC CD893D                 CALL POFETCH        ;USE VALUE AS NEW 'OLDPOS'
F4DF E1                     POP HL              ;OLDPOS
F4E0 CD14F5                 CALL LSASR          ;ADJUST IF LS SCROLLED
F4E3 D5                     PUSH DE             ;NEW OLDPOS
F4E4 E5                     PUSH HL
F4E5
F4E5 C1         EDBL:       POP BC              ;OLDPOS
F4E6 CD893D                 CALL POFETCH        ;NEW POSN OF END OF LINE IN LOWER SCREEN
F4E9 EB                     EX DE,HL
F4EA A7                     AND A
F4EB ED42                   SBC HL,BC           ;NORMALLY NEW POSN IS FURTHER RIGHT, OR DOWN, AS
F4ED 300F                   JR NC,EDBE          ;TEXT IS ADDED, SO NC. BUT IF DELETE USED, CY
F4EF
F4EF C5                     PUSH BC
F4F0 3E20                   LD A," "
F4F2 CD7905                 CALL FONOP          ;NORMAL OUTPUT
F4F5 18EE                   JR EDBL             ;LOOP TILL LINE BLANKED TO OLD END-OF-LINE.
F4F7
F4F7            ;EDPRNT ERRORS COME HERE
F4F7
F4F7 CD8804     EDPE:       CALL WARNBZ
F4FA CD893D                 CALL POFETCH        ;FOR OLDPOS
F4FD 21                     DB 21H              ;'JR+2'
F4FE
F4FE D1         EDBE:       POP DE              ;LAST END-OF-LINE POSN
F4FF E1                     POP HL              ;ERROR HANDLER ADDR
F500
F500 E1                     POP HL
F501 223D5C                 LD (ERRSP),HL
F504 ED53825C               LD (OLDPOS),DE
F508 E1                     POP HL              ;ORIG SCRN POSN
F509 CD14F5                 CALL LSASR
F50C CD62DC                 CALL POSTORE        ;ORIG SCRN POSN, MODIFIED IF SCOLLED UP
F50F AF                     XOR A
F510 32A45A                 LD (XPTR+1),A       ;CANCEL ANY '?' ERROR MARKER
F513 C9                     RET
F514
F514 3ACF5A     LSASR:      LD A,(TEMPB2)
F517 47                     LD B,A
F518 3A585A                 LD A,(WINDTOP)
F51B 90                     SUB B               ;A IS EG -1 IF LW HAS EXPANDED UP BY ONE
F51C 84                     ADD A,H             ;EG ADJUST PRINT LINE UP BY ONE IF LS SCROLLED
F51D 67                     LD H,A
F51E C9                     RET
F51F
F51F            ;EDITOR'S SPECIAL CURSOR OUTPUT TO SET SCREEN POSN
F51F
F51F            CUOPP:
F51F ED4B6E5A               LD BC,(SPOSNL)
F523 2AC25A                 LD HL,(KPOS)
F526 A7                     AND A
F527 ED42                   SBC HL,BC
F529 200B                   JR NZ,CUOP2
F52B
F52B 2A8B5B                 LD HL,(LSPTR)
F52E 229A5A                 LD (KCUR),HL
F531 F5                     PUSH AF
F532 CD50F5                 CALL OPCUR2         ;PREVENT LINE BEING BRIEFLY PRINTED 1 CHAR SHORTER
F535 F1                     POP AF              ;AS OLD CURSOR VANISHES AND NEW CURS. POSN FOUND
F536
F536 C37905     CUOP2:      JP FONOP
F539
F539
```

```
F539                ;OP CURSOR IF DE=KCUR.
F539
F539 2A9A5A     OPCURSOR:  LD HL,(KCUR)
F53C A7                    AND A
F53D ED52                  SBC HL,DE
F53F C0                    RET NZ              ;RET IF THIS ISN'T CURSOR'S ADDR IN LINE
F540
F540 DBFB                  IN A,(251)
F542 67                    LD H,A
F543 3A995A                LD A,(KCURP)
F546 AC                    XOR H
F547 E61F                  AND 1FH
F549 C0                    RET NZ              ;**
F54A
F54A 2A6E5A                LD HL,(SPOSNL)
F54D 22C25A                LD (KPOS),HL        ;SAVE SCREEN POSN OF CURSOR
F550
F550 2AFE5A     OPCUR2:    LD HL,(KURV)
F553 CD10F2                CALL JPOPT
F556
F556 2A015A                LD HL,(KURCHAR)
F559 3A6A5C                LD A,(FLAGS2)
F55C E608                  AND 08H
F55E 7D                    LD A,L
F55F 280F                  JR Z,PRINVERT       ;JR IF CAPS LOCK OFF
F561
F561 7C                    LD A,H
F562 180C                  JR PRINVERT
F564
F564                ;PRINT LINE CURSOR - USED BY MAIN LIST ROUTINE AND FUPDN
F564
F564 3A6D5A     PRLCU:     LD A,(SPOSNU+1)
F567 328D5B                LD (LNPTR),A        ;STORE LINE NO. WITH CURSOR
F56A 3A005A                LD A,(LNCUR)
F56D FE                    DB 0FEH             ;"JR+2" (CP 3EH: CCF). PRINT INVERSE '>'
F56E
F56E 3E3F       PRFLQUERY: LD A,"?"
F570
F570                ;PRINT CHAR IN 'A' REG IN INVERSE VIDEO
F570                ;USE CALL TO MNOP (VIA FONOP), NOT RST 10H, IN CASE INVERSE CHAR IS PRINTED
F570                ;BETWEEN CC AND ITS PARAMETER (AND CHANNEL ADDR HAS BEEN CHANGED).
F570
F570 47         PRINVERT:  LD B,A
F571 E5                    PUSH HL
F572 21505A                LD HL,PFLAGT
F575 7E                    LD A,(HL)
F576 F5                    PUSH AF
F577 CBD6                  SET 2,(HL)          ;INVERSE 1
F579 3A545A                LD A,(INVERT)
F57C F5                    PUSH AF
F57D 3ABA5A                LD A,(INQUFG)
F580 F5                    PUSH AF
F581 3A345A                LD A,(BGFLG)
F584 F5                    PUSH AF
F585 3EFF                  LD A,0FFH
F587 32BA5A                LD (INQUFG),A       ;'IN QUOTES' SO CURSORS CAN BE UDGS
F58A 32545A                LD (INVERT),A
F58D 32345A                LD (BGFLG),A
F590 78                    LD A,B
F591 D5                    PUSH DE
F592 CD7905                CALL FONOP
F595 D1                    POP DE
F596 F1                    POP AF
F597 32345A                LD (BGFLG),A
F59A F1                    POP AF
F59B 32BA5A                LD (INQUFG),A
F59E F1                    POP AF
F59F 32545A                LD (INVERT),A
F5A2 F1                    POP AF
F5A3 32505A                LD (PFLAGT),A
F5A6 E1                    POP HL
F5A7 C9                    RET
F5A8
F5A8
F5A8                ;PRINT INTEGER IN A, AS 1-3 CHARS. HL IS SAVED. USED FOR STAT AND ERROR NUMS.
F5A8
F5A8 4F         PRAREG:    LD C,A
F5A9 0600                  LD B,0
F5AB
F5AB                ;USED FOR LINE NUMS IN REPORTS
F5AB
F5AB 1E00       PRNUMB1:   LD E,0              ;NO LEADING SPACES
F5AD 78                    LD A,B
F5AE 3C                    INC A
F5AF 2005                  JR NZ,PRNUMBC
F5B1
F5B1 4F                    LD C,A
F5B2 47                    LD B,A
F5B3 22                    DB 22H              ;"JR+2". NUMBERS >=FF00 PRINT AS 0 (EG ELINE)
F5B4
F5B4                ;PRINT INTEGER IN BC, PADDED TO 5 CHARS WITH LEADING SPACES. HL IS SAVED.
F5B4                ;USED FOR LINE NUMS IN LISTINGS. USES BC, DE, AF.
F5B4
F5B4 1E20       PRNUMB2:   LD E,20H            ;LEADING SPACES
F5B6
F5B6 E5         PRNUMBC:   PUSH HL
F5B7 C5                    PUSH BC             ;NUMBER
F5B8 2146FE                LD HL,SUBTAB
F5BB
F5BB 4E         PRNUOLP:   LD C,(HL)
F5BC 23                    INC HL
F5BD 46                    LD B,(HL)
F5BE 23                    INC HL
```

```
F5BF E3                   EX (SP),HL        ;HL=NUMBER
F5C0 7D                   LD A,L
F5C1 0C                   INC C
F5C2 0D                   DEC C
F5C3 280A                 JR Z,PRNTNO1      ;JR IF WE ARE AT 'UNITS' STAGE (C=TERMINATOR)
F5C5
F5C5 AF                   XOR A
F5C6
F5C6 3C         PRNUILP:  INC A
F5C7 09                   ADD HL,BC
F5C8 38FC                 JR C,PRNUILP      ;SUCCESSIVE SUBTRACTIONS OF EG. 10, 100
F5CA
F5CA ED42                 SBC HL,BC
F5CC 3D                   DEC A             ;A=DIGIT FOR 1000'S, 100'S ETC.
F5CD 2802                 JR Z,PRNTNO2      ;LEAVE E ALONE IF IT'S A ZERO DIGIT
F5CF
F5CF 1E30       PRNTNO1:  LD E,30H          ;AFTER A NON-ZERO DIGIT USE '0' FOR ZERO, RATHER
F5D1                                        ;THAN ' ' OR ''.
F5D1 83         PRNTNO2:  ADD A,E
F5D2 C41000               CALL NZ,0010H     ;CALL UNLESS LEADING SPACES SUPPRESSED.
F5D5 E3                   EX (SP),HL        ;HL=TABLE PTR
F5D6 0C                   INC C
F5D7 0D                   DEC C
F5D8 20E1                 JR NZ,PRNUOLP     ;LOOP IF NOT AT 'UNITS' YET
F5DA
F5DA E1                   POP HL            ;NUMBER
F5DB E1                   POP HL            ;ORIG
F5DC C9                   RET
F5DD
F5DD                                            ;OPCURSOR, PRLCU, PRINVERT, STENTS, PRAREG,
F5DD                                            ;PRNUMB
F5DD
F5DD                      INCLUDE TEXT.SAM       ;TABLES, MSGS, KEYWORDS, CMDADDRT, CHARSET
F5DD            ;TEXT.SAM - ERROR MESSAGES, KEYWORDS, MSGS, TABLES, CHARSET ETC
F5DD
F5DD            ;REPORTS GENERATED BY E.G. RST 08H: DB 0 FOR OK
F5DD            ;UTILITY MESSAGES (START AT 0)
F5DD
F5DD            ;UMVAL IS LOADED INTO UMSGS SYS VAR AT INIT.
F5DD            UMVAL:
F5DD
F5DD 2020204D49   DB "   MILES GORDON TECHNOLOGY plc  "
F5FD 2020202020   DB "     " 127 " 1990  SAM Cou","p"+80H        ;0
F612 7363726F6C   DB "scroll","?"+80H                            ;1
F619 A0           DC "                                ;2 NOT USED
F61A 53741F7420   DB "St",CAR,"t tape and then press a ke","y"+80H    ;3
F638 4261736963   DB "Basic:"," "+80H                           ;4
F63F 4E751E7269   DB "Nu",CME,"ric",ARRAY+80H                     ;5
F646 138E         DB STRING,ARRAY+80H                  ;6
F648 436F1D3AA0   DB "Co",CDE,":"," "+80H                        ;7
F64D 530D3AA0     DB "S",CREEN,":"," "+80H                       ;8
F651
F651            ;ERRMVAL IS LOADED INTO ERRMSGS SYS VAR AT INIT.
F651            ERRMVAL:
F651
F651 4FCB         DB "O","K"+80H                                 ;0
F653 4F7574061E   DB "Out",SOFS,CME,"mor","y"+80H                ;1
F65C 11666F756E   DB SNOTS,"foun","d"+80H                       ;2
F662 4441544120   DB "DATA has all been rea","d"+80H            ;3
F678 5375627363   DB "Subscript wron","g"+80H                   ;4
F687 0A01464FD2   DB CNXT,WITHOUT,"FO","R"+80H                  ;5
F68C 464F52018A   DB "FOR",WITHOUT,CNXT+80H                ;6
F691 464E014445   DB "FN",WITHOUT,"DEF F","N"+80H               ;7
F69A 5245545552   DB "RETURN",WITHOUT,"GOSU","B"+80H           ;8
F6A6 029B         DB MISSING,CLOOP+80H                  ;9
F6A8 1B0144CF     DB CLOOP,WITHOUT,"D","O"+80H                 ;10
F6AC 0B504F5020   DB NO,"POP dat","a"+80H                      ;11
F6B5 024445469A   DB MISSING,"DEF",CPROC+80H                  ;12
F6BA 0B454E449A   DB NO,"END",CPROC+80H                       ;13
F6BF 0C2D20434F   DB BREAK,"-CONTINUE",CTO," repea","t"+80H       ;14
F6D3 0C191C2070   DB BREAK,IN,CTO," progra","m"+80H            ;15
F6DE 167395       DB CSTOP,"s",TATEMENT+80H                 ;16
F6E1 161920494E   DB CSTOP,IN," INPU","T"+80H                   ;17
F6E9 001792       DB INVALID,FILE,SNAME+80H              ;18
F6EC 4C6F616419   DB "Load",IN,"g",ERROR+80H                  ;19
F6F3 001D766963   DB INVALID,CDE,"vic","e"+80H                 ;20
F6F9 007308206E   DB INVALID,"s",TREAM," n",UMBER+80H       ;21
F6FF 456E640697   DB "End",SOFS,FILE+80H                  ;22
F704 0098         DB INVALID,CLOUR+80H                   ;23
F706 000598       DB INVALID,PALET,CLOUR+80H             ;24
F709 1405636861   DB TOOMANY,PALET,"change","s"+80H        ;25
F712 501F611E74   DB "P",CAR,"a",CME,"ter",ERROR+80H              ;26
F71A 001F67751E   DB INVALID,CAR,"gu",CME,"n","t"+80H             ;27
F721 4E09036C1F   DB "N",UMBER,TOO,"l",CAR,"g","e"+80H          ;28
F728 4E6F742075   DB "Not un",CDE,"rs",CTO,"o","d"+80H                ;29
F734 496E746567   DB "Integer out",SOFS,"rang","e"+80H        ;30
F745 531520646F   DB "S",TATEMENT," doesn't exis","t"+80H        ;31
F755 4F66662073   DB "Off s",CREEN+80H                  ;32
F75B 0B726F6F6D   DB NO,"room for l",IN,"e"+80H                 ;33
F768 00730D206D   DB INVALID,"s",CREEN," mod","e"+80H           ;34
F770 00424C4954   DB INVALID,"BLITZ cod","e"+80H               ;35
F77B 531C726564   DB "S",CTO,"red ",CAR,"ea",TOO,"bi","g"+80H          ;36
F788 0050555420   DB INVALID,"PUT bloc","k"+80H                ;37
F792 505554206D   DB "PUT mask mismatc","h"+80H                ;38
F7A3 02454E4420   DB MISSING,"END I","F"+80H                   ;39
F7AA 00761F6961   DB INVALID,"v",CAR,"iable",SNAME+80H         ;40
F7B3 4241534943   DB "BASIC stack ful","l"+80H                 ;41
F7C3 130390       DB STRING,TOO,LONG+80H                 ;42
F7C6 00730D206E   DB INVALID,"s",CREEN," n",UMBER+80H        ;43
F7CC 530D84       DB "S",CREEN,ISALREDOP+80H                 ;44
F7CF 530884       DB "S",TREAM,ISALREDOP+80H                 ;45
F7D2 4375727265   DB "Current s",CREEN+80H                   ;46  WAS INVALID CHANNEL
F7DC 5308206973   DB "S",TREAM," is",SNOTS,"ope","n"+80H      ;47
```

```
F7E6 00434C4541   DB INVALID,"CLEAR addres","s"+80H          ;48
F7F4 008F         DB INVALID,NOTE+80H                      ;49
F7F6 0F0390       DB NOTE,TOO,LONG+80H                     ;50
F7F9 46504387     DB "FPC",ERROR+80H                       ;51
F7FD 141D661969   DB TOOMANY,CDE,"f",IN,"ition","s"+80H          ;52
F807 4E6F20444F   DB "No DO","S"+80H                       ;53
F80D 0057494E44   DB INVALID,"WINDO","W"+80H               ;54
F814 02646973EB   DB MISSING,"dis","k"+80H                 ;55
F819
F819              ;SAVES OVER 200 BYTES.
F819
F819 496E76616C COMPLIST:  DB "Invalid"," "+80H           ;14 uses
F821
F821 0000=      INVALID    EQU 0
F821
F821 2077697468            DB " without"," "+80H          ;5 uses
F82A
F82A 0001=      WITHOUT    EQU 1
F82A
F82A 4D69737369            DB "Missing"," "+80H           ;5 uses
F832
F832 0002=      MISSING    EQU 2
F832
F832 20746F6FA0            DB " too"," "+80H              ;3 uses
F837
F837 0003=      TOO        EQU 3
F837
F837 2069732061            DB " is already ope","n"+80H   ;2 uses
F847
F847 0004=      ISALREDOP  EQU 4
F847
F847 70616C6574            DB "palette"," "+80H           ;2 uses
F84F
F84F 0005=      PALET      EQU 5
F84F
F84F 206F66A0              DB " of"," "+80H               ;3 uses
F853
F853 0006=      SOFS       EQU 6
F853
F853 206572726F            DB " erro","r"+80H             ;4 uses
F859
F859 0007=      ERROR      EQU 7
F859
F859 74726561ED            DB "trea","m"+80H              ;3 uses
F85E
F85E 0008=      TREAM      EQU 8
F85E
F85E 756D6265F2            DB "umbe","r"+80H              ;3 uses
F863
F863 0009=      UMBER      EQU 9
F863
F863 4E4558D4              DB "NEX","T"+80H               ;2 uses
F867
F867 000A=      CNXT       EQU 10
F867
F867 4E6FA0                DB "No"," "+80H                ;3 uses
F86A
F86A 000B=      NO         EQU 11
F86A
F86A 425245414B            DB "BREAK"," "+80H             ;2 uses
F870
F870 000C=      BREAK      EQU 12
F870
F870 63726565EE            DB "cree","n"+80H              ;6 USES
F875
F875 000D=      CREEN      EQU 13
F875
F875 2061727261            DB " array:"," "+80H           ;2 USES
F87D
F87D 000E=      ARRAY      EQU 14
F87D
F87D 4E6F74E5              DB "Not","e"+80H               ;2 USES
F881
F881 000F=      NOTE       EQU 15
F881
F881 6C6F6EE7              DB "lon","g"+80H               ;2 USES
F885
F885 0010=      LONG       EQU 16
F885
F885 206E6F74A0            DB " not"," "+80H              ;2 USES
F88A
F88A 0011=      SNOTS      EQU 17
F88A
F88A 206E616DE5            DB " nam","e"+80H              ;2 USES
F88F
F88F 0012=      SNAME      EQU 18
F88F
F88F 537472696E            DB "Strin","g"+80H          ;2 USES
F895
F895 0013=      STRING     EQU 19
F895
F895 546F6F206D            DB "Too many"," "+80H       ;2 USES
F89E
F89E 0014=      TOOMANY    EQU 20
F89E
F89E 746174656D            DB "tatemen","t"+80H         ;2 USES
F8A6
F8A6 0015=      TATEMENT   EQU 21
F8A6
F8A6 53544F50A0            DB "STOP"," "+80H           ;2 USES
F8AB
F8AB 0016=      CSTOP      EQU 22
F8AB
```

```
F8AB 66696CE5              DB "fil","e"+80H            ;2 USES
F8AF
F8AF 0017=      FILE       EQU 23
F8AF
F8AF 636F6C6F75            DB "colou","r"+80H          ;2 USES
F8B5
F8B5 0018=      CLOUR      EQU 24
F8B5
F8B5 69EE                  DB "i","n"+80H              ;5 USES
F8B7
F8B7 0019=      IN         EQU 25
F8B7
F8B7 2050524FC3            DB " PRO","C"+80H           ;2 USES
F8BC
F8BC 001A=      CPROC      EQU 26
F8BC
F8BC 4C4F4FD0              DB "LOO","P"+80H            ;2 USES
F8C0
F8C0 001B=      CLOOP      EQU 27
F8C0
F8C0 74EF                  DB "t","o"+80H              ;4 USES
F8C2
F8C2 001C=      CTO        EQU 28
F8C2
F8C2 64E5                  DB "d","e"+80H              ;4 USES
F8C4
F8C4 001D=      CDE        EQU 29
F8C4
F8C4 6DE5                  DB "m","e"+80H              ;2 USES
F8C6
F8C6 001E=      CME        EQU 30
F8C6
F8C6 61F2                  DB "a","r"+80H              ;6 USES
F8C8
F8C8 001F=      CAR        EQU 31
F8C8
F8C8            ;TOKENS.SAM - SAM TOKEN TABLE.
F8C8            ;FUNCTIONS RANGE 21-69H, PRECEDED BY FF
F8C8            ;ASCII RANGE 20-7F
F8C8            ;BLOCK GRAPHICS/EXTENDED CHARS  80-8F
F8C8            ;EXTENDED CHARS 90-A8
F8C8            ;(80H-A8H ACT AS UDGS - PROGRAMMED AS BLOCK GRAPHICS, FOLLOWED BY EXTENDED CHARS
F8C8            ;ON POWER-UP. UDGS SYS VAR PTS TO USR "A". TAKES 328 BYTES OF RAM. OVERLAP WITH
F8C8            ;COMMAND TOKENS USES CONTEXT TO AVOID PROBLEMS - PRINT CHR$ 160 GIVES UDG, LIST
F8C8            ;COMMAND GIVES KEYWORD. QUOTES FORCES UDGS.
F8C8            ;ALLOW 80-95 IN PROC NAMES SO A FACE SYMBOL CAN BE A COMMAND?
F8C8            ;USE "." FOR EXTERNAL COMMAND
F8C8
F8C8            ;QUALIFIERS  96-9F
F8C8            ;COMMANDS RANGE A0-FE
F8C8
F8C8            ;FUNCTIONS HAVE 0FFH PREFIX
F8C8
F8C8            ;"IMMEDIATE" FUNCTIONS
F8C8
F8C8 A0         KEYWTAB:   DB 0A0H
F8C9 50C9       IMFNTL:    DC "PI"             ;3BH
F8CB 524EC4                DC "RND"            ;
F8CE 504F494ED4            DC "POINT"          ;
F8D3 465245C5              DC "FREE"           ;
F8D7 4C454E4754            DC "LENGTH"         ;
F8DD 495445CD              DC "ITEM"           ;
F8E1 415454D2              DC "ATTR"           ;
F8E5 46CE                  DC "FN"             ;
F8E7 4249CE                DC "BIN"            ;
F8EA 584D4F5553            DC "XMOUSE"         ;
F8F0 594D4F5553            DC "YMOUSE"         ;
F8F6 585045CE              DC "XPEN"           ;
F8FA 595045CE              DC "YPEN"           ;
F8FE 52414D544F            DC "RAMTOP"         ;
F904 AD                    DC "-"              ; UNUSED INARRAY
F905 494E5354D2            DC "INSTR"          ;4AH
F90A
F90A 494E4B4559            DC "INKEY$"         ;4BH
F910 5343524545            DC "SCREEN$"        ;
F917 4D454DA4              DC "MEM$"           ;
F91B AD                    DC "-"              ; UNUSED CHAR$
F91C 50415448A4            DC "PATH$"          ;
F921 535452494E            DC "STRING$"        ;
F928 AD                    DC "-"              ; UNUSED USING$
F929 AD                    DC "-"              ;52H UNUSED SHIFT$
F92A
F92A            ;FPC FUNCTIONS
F92A
F92A 5349CE     FPCFNTL:   DC "SIN"            ;53H
F92D 434FD3                DC "COS"            ;
F930 5441CE                DC "TAN"            ;
F933 4153CE                DC "ASN"            ;
F936 4143D3                DC "ACS"            ;
F939 4154CE                DC "ATN"            ;
F93C 4CCE                  DC "LN"             ;
F93E 4558D0                DC "EXP"            ;
F941 4142D3                DC "ABS"            ;
F944 5347CE                DC "SGN"            ;
F947 5351D2                DC "SQR"            ;
F94A 494ED4                DC "INT"            ;
F94D 5553D2                DC "USR"            ;
F950 49CE                  DC "IN"             ;
F952 504545CB              DC "PEEK"           ;
F956 44504545CB            DC "DPEEK"          ;
F95B 445641D2              DC "DVAR"           ;
F95F 535641D2              DC "SVAR"           ;
F963 425554544F            DC "BUTTON"         ;
```

```
F969 454FC6                  DC "EOF"           ;
F96C 5054D2                  DC "PTR"           ;   DISC USE?
F96F AD                      DC "-"             ; RESERVED
F970 5544C7                  DC "UDG"           ;
F973 AD                      DC "-"             ; NUMBER
F974 4C45CE                  DC "LEN"           ;
F977 434F44C5                DC "CODE"          ;
F97B 56414CA4                DC "VAL$"          ;
F97F 5641CC                  DC "VAL"           ;
F982 5452554E43              DC "TRUNC$"        ;
F988 434852A4                DC "CHR$"          ;
F98C 535452A4                DC "STR$"          ;
F990 42494EA4                DC "BIN$"          ;
F994 484558A4                DC "HEX$"          ;
F998 555352A4                DC "USR$"          ;
F99C AD                      DC "-"             ; CORRESPONDS TO INKEY$ FPC CODE
F99D 4E4FD4                  DC "NOT"           ;
F9A0 AD                      DC "-"             ;
F9A1 AD                      DC "-"             ;
F9A2 AD                      DC "-"             ;79H
F9A3
F9A3                ;BINARY OPERATORS
F9A3
F9A3 4D4FC4         BINFNTL: DC "MOD"           ;7AH
F9A6 4449D6                  DC "DIV"           ;
F9A9 424FD2                  DC "BOR"           ;
F9AC AD                      DC "-"             ;BXOR
F9AD 42414EC4                DC "BAND"          ;
F9B1 4FD2                    DC "OR"            ;
F9B3 414EC4                  DC "AND"           ;
F9B6 3CBE                    DC "<>"            ;
F9B8 3CBD                    DC "<="            ;
F9BA 3EBD                    DC ">="            ;83H
F9BC
F9BC
F9BC                ;QUALIFIERS - RANGE 85H-8FH
F9BC
F9BC 5553494EC7 KWDS85:      DC "USING"             ;85H
F9C1 57524954C5              DC "WRITE"             ;86
F9C6 41D4                    DC "AT"                ;87
F9C8 5441C2                  DC "TAB"               ;88
F9CB 4F46C6                  DC "OFF"               ;89
F9CE 5748494CC5              DC "WHILE"             ;8A
F9D3 554E5449CC              DC "UNTIL"             ;8B
F9D8 4C494EC5                DC "LINE"              ;8C
F9DC 544845CE                DC "THEN"              ;8D
F9E0 54CF                    DC "TO"                ;8E
F9E2 535445D0                DC "STEP"              ;8F
F9E6
F9E6                ;COMMANDS.  RANGE 90H-FEH
F9E6
F9E6 4449D2                  DC "DIR"               ;90
F9E9 464F524D41              DC "FORMAT"            ;91
F9EF 45524153C5              DC "ERASE"             ;92
F9F4 4D4F56C5                DC "MOVE"              ;93
F9F8 534156C5                DC "SAVE"              ;94
F9FC 4C4F41C4                DC "LOAD"              ;95
FA00 4D455247C5              DC "MERGE"             ;96
FA05 5645524946              DC "VERIFY"            ;97
FA0B 4F5045CE                DC "OPEN"              ;98
FA0F 434C4F53C5              DC "CLOSE"             ;99
FA14
FA14 434952434C              DC "CIRCLE"            ;9A
FA1A 504C4FD4                DC "PLOT"              ;9B
FA1E 4C45D4                  DC "LET"               ;9C
FA21 424C4954DA              DC "BLITZ"             ;9D
FA26 424F524445              DC "BORDER"            ;9E
FA2C 434CD3                  DC "CLS"               ;9F
FA2F
FA2F 50414C4554 KWDSA0:      DC "PALETTE"           ;A0
FA36 5045CE                  DC "PEN"               ;A1
FA39 50415045D2              DC "PAPER"             ;A2
FA3E 464C4153C8              DC "FLASH"             ;A3
FA43 4252494748              DC "BRIGHT"            ;A4
FA49 494E564552              DC "INVERSE"           ;A5
FA50 4F5645D2                DC "OVER"              ;A6
FA54
FA54 4641545049              DC "FATPIX"            ;A7  0=THIN, 1=FAT
FA5A 4353495AC5              DC "CSIZE"             ;A8
FA5F 424C4F434B              DC "BLOCKS"            ;A9
FA65 4D4F44C5                DC "MODE"              ;AA
FA69 475241C2                DC "GRAB"              ;AB
FA6D 5055D4                  DC "PUT"               ;AC
FA70
FA70 424545D0                DC "BEEP"              ;AD
FA74 534F554EC4              DC "SOUND"             ;AE
FA79
FA79 4E45D7                  DC "NEW"               ;AF
FA7C 5255CE                  DC "RUN"               ;B0
FA7F 53544FD0                DC "STOP"              ;B1
FA83 434F4E5449              DC "CONTINUE"          ;B2
FA8B 434C4541D2              DC "CLEAR"             ;B3
FA90 474F2054CF              DC "GO TO"             ;B4
FA95 474F205355              DC "GO SUB"            ;B5
FA9B 5245545552              DC "RETURN"            ;B6
FAA1 5245CD                  DC "REM"               ;B7
FAA4
FAA4 524541C4                DC "READ"              ;B8
FAA8 444154C1                DC "DATA"              ;B9
FAAC 524553544F              DC "RESTORE"           ;BA
FAB3
FAB3 5052494ED4              DC "PRINT"             ;BB
FAB8 4C5052494E              DC "LPRINT"            ;BC
```

```
FABE 4C4953D4                DC "LIST"              ;BD
FAC2 4C4C4953D4              DC "LLIST"             ;BE
FAC7 44554DD0                DC "DUMP"              ;BF
FACB
FACB 464FD2     KWDSC0:      DC "FOR"               ;C0
FACE 4E4558D4                DC "NEXT"              ;C1
FAD2 50415553C5              DC "PAUSE"             ;C2
FAD7 445241D7                DC "DRAW"              ;C3
FADB 4445464155              DC "DEFAULT"           ;C4
FAE2 4449CD                  DC "DIM"               ;C5
FAE5 494E5055D4              DC "INPUT"             ;C6
FAEA 52414E444F              DC "RANDOMIZE"         ;C7
FAF3 4445462046              DC "DEF FN"            ;C8
FAF9 444546204B              DC "DEF KEYCODE"       ;C9
FB04 4445462050              DC "DEF PROC"          ;CA
FB0C 454E442050              DC "END PROC"          ;CB
FB14 52454E55CD              DC "RENUM"             ;CC
FB19 44454C4554              DC "DELETE"            ;CD
FB1F 5245C6                  DC "REF"               ;CE
FB22 434F50D9                DC "COPY"              ;CF
FB26 AD                      DC "-"                 ;D0 EDIT
FB27 4B455949CE              DC "KEYIN"             ;D1
FB2C 4C4F4341CC              DC "LOCAL"             ;D2
FB31 4C4F4F5020              DC "LOOP IF"           ;D3
FB38 44CF                    DC "DO"                ;D4
FB3A 4C4F4FD0                DC "LOOP"              ;D5
FB3E 4558495420              DC "EXIT IF"           ;D6
FB45 49C6                    DC "IF"                ;D7
FB47 49C6                    DC "IF"                ;D8
FB49 454C53C5                DC "ELSE"              ;D9
FB4D 454C53C5                DC "ELSE"              ;DA
FB51 454E442049              DC "END IF"            ;DB
FB57 4B45D9                  DC "KEY"               ;DC
FB5A 4F4E204552              DC "ON ERROR"          ;DD
FB62 4FCE                    DC "ON"                ;DE
FB64 4745D4                  DC "GET"               ;DF
FB67
FB67 4F55D4     KWDSE0:      DC "OUT"               ;E0
FB6A 504F4BC5                DC "POKE"              ;E1
FB6E 44504F4BC5              DC "DPOKE"             ;E2
FB73 52454E414D              DC "RENAME"            ;E3
FB79 43414CCC                DC "CALL"              ;E4
FB7D 524F4CCC                DC "ROLL"              ;E5
FB81 5343524F4C              DC "SCROLL"            ;E6
FB87 5343524545              DC "SCREEN"            ;E7
FB8D 4449535043              DC "DISPLAY"           ;E8
FB94 424F4FD4   BTWD:        DC "BOOT"              ;E9 USED BY BOOT TO CHECK FILE NAME
FB98 4C414245CC              DC "LABEL"             ;EA
FB9D 46494CCC                DC "FILL"              ;EB
FBA1 57494E444F              DC "WINDOW"            ;EC
FBA7 415554CF                DC "AUTO"              ;ED
FBAB 504FD0                  DC "POP"               ;EE
FBAE 5245434F52              DC "RECORD"            ;EF
FBB4 4445564943              DC "DEVICE"            ;F0
FBBA 50524F5445              DC "PROTECT"           ;F1
FBC1 484944C5                DC "HIDE"              ;F2
FBC5 5A41D0                  DC "ZAP"               ;F3
FBC8 504FD7                  DC "POW"               ;F4
FBCB 424F4FCD                DC "BOOM"              ;F5
FBCF 5A4F4FCD                DC "ZOOM"              ;F6
FBD3 AD                      DC "-"                 ;F7
FBD4 AD                      DC "-"                 ;F8
FBD5 AD                      DC "-"                 ;F9
FBD6 AD                      DC "-"                 ;FA
FBD7 AD                      DC "-"                 ;FB
FBD8 AD                      DC "-"                 ;FC
FBD9 AD                      DC "-"                 ;FD
FBDA AD                      DC "-"                 ;FE
FBDB 494ECB                  DC "INK"               ;FF (FUNCTION PREFIX) AND TEMP "INK" TOKEN
FBDE
FBDE
FBDE
FBDE             ;DEF KEY INIT TABLE
FBDE
FBDE C00100BD   DKSRC:     DB 192,1,0,0BDH                          ;F0 LIST
FBE2 C10200CC3A            DB 193,2,0,0CCH,":"                      ;F1 RENUM :
FBE7 C20200BB3A            DB 194,2,0,0BBH,":"                      ;F2 PRINT :
FBEC C30200AA3A            DB 195,2,0,0AAH,":"                      ;F3 MODE :
FBF1 C40100B0              DB 196,1,0,0B0H                          ;F4 RUN
FBF5 C50100B2              DB 197,1,0,0B2H                          ;F5 CONTINUE
FBF9 C602009F23            DB 198,2,0,09FH,"#"                      ;F6 CLS #
FBFE C703009522            DB 199,3,0,LOADTOK,22H,22H               ;F7 LOAD ""
FC04 C805009522            DB 200,5,0,LOADTOK,22H,22H,0FFH,CODETOK  ;F8 LOAD "" CODE
FC0C C90100E9              DB 201,1,0,0E9H                          ;F9 BOOT
FC10
FC10 FC0200063A            DB 252,2,0,6,":"                         ;TAB
FC15 FD03001401            DB 253,3,0,20,1,":"                      ;INVERSE 1 CC
FC1B FE03001400            DB 254,3,0,20,0,":"                      ;INVERSE 0 CC
FC21 FF          DKEN:     DB 0FFH
FC22
FC22             ;INIT TABLE - 18 BYTES
FC22
FC22 3E          CHIT:     DB ">"          ;CURRENT LINE CURSOR
FC23 8081                  DB 128,129      ;CURSOR CHARACTERS - LOWER CASE/UPPER CASE
FC25 3130                  DB "1","0"      ;DIGIT CHARS FOR "BIN$"
FC27 23                    DB "#"          ;INSTR WILD CARD CHAR
FC28 5470                  DB "T",TSPEED   ;DEVICE T, TAPE SPEED
FC2A 11                    DB 17           ;COLOUR FLASH SPEED
FC2B 0056                  DW LINICOLS     ;LINIPTR
FC2D FF0000                DB 0FFH,0,0     ;XCMDP SHOWS "NO EXTERNAL CMDS"
FC30
FC30 4F                    DB 79           ;PRINTER RHS
FC31 0A                    DB 10           ;AFTER CR CHARACTER
```

```
FC32 E9                    DB 0E9H       ;PRINTER CONTROL PORT (E8=DATA)
FC33 01                    DB 1          ;PRINTER STROBE BYTE
FC34
FC34 FF4A       MAIT:      DW BSTACK     ;BASSTK
FC36 0040                  DW HPEND      ;HEAPEND
FC38 0040                  DW HPEND      ;HPST  START WITH EMPTY HEAP
FC3A 004D                  DW FPSB       ;FPSBOT
FC3C 0058                  DW DKBU       ;DEF KEY BUFFER START
FC3E 8058                  DW DKBU+128   ;LIMIT OF DEF KEY BUFFER GROWTH
FC40 77DC                  DW ENDOUTP    ;"PRINTABLE CHARS" O/P ROUTINE
FC42 51F6                  DW ERRMVAL    ;ERROR MESSAGES
FC44 DDF5                  DW UMVAL      ;"UTILITY" MESSAGES
FC46 E058                  DW KTAB       ;KEYBOARD TABLE
FC48 65FD                  DW CMDADT     ;START OF CMD ADDR TABLE
FC4A                                     ;FIRST 4 BYTES IN CHANTAB ALSO USED!
FC4A
FC4A CC01       CHANTAB:   DW PRMAIN
FC4C 0205                  DW KYIP
FC4E 4B                    DB "K"
FC4F
FC4F CC01                  DW PRMAIN
FC51 0205                  DW KYIP
FC53 53                    DB "S"
FC54
FC54 F202                  DW ADDCHAR
FC56 123F                  DW IDERR
FC58 52                    DB "R"
FC59
FC59 CC01                  DW PRMAIN
FC5B 123F                  DW IDERR
FC5D 50                    DB "P"
FC5E
FC5E AF01                  DW S16OP
FC60 123F                  DW IDERR
FC62 24                    DB "$"
FC63
FC63 8101                  DW SENDA
FC65 123F                  DW IDERR
FC67 42                    DB "B"
FC68
FC68 0D                    DB 0DH             ;KEEPS READ CMD HAPPY ON FIRST LINE
FC69
FC69            ;KEY MAP FOR 70 KEYS (3 NOT READ THIS WAY) IN UNSHIFTED, CAPS SHIFTED AND SYMB.
FC69            ;SHIFTED STATES. EACH ROW CORRESPONDS TO A BIT ON THE PORT (BIT 7 AT TOP) AND
FC69            ;EACH COLUMN TO A PORT MSB. CAPS SHIFT (BIT 0,PORT FE) SCANNED SEPARATELY, AS
FC69            ;IS SYM SHIFT (BIT 1,PORT 7F) AND CONTROL (BIT 0,PORT FF)
FC69
FC69            ;PORT MSB:     FF  7F  BF  DF  EF  F7  FB  FD  FE
FC69            ;BIT MAP BYTE  8   7   6   5   4   3   2   1   0
FC69
FC69            ;254 DEFINED AS INVERSE CC,0
FC69            ;253 DEFINED AS INVERSE CC,1
FC69            ;252 DEFINED AS CHR$ 6
FC69
FC69 FD07C00C06 KSRC:      DB     253,07H,0C0H,12,6H,0C9H,198,195  ;BIT 7
FC71 582E3A222B            DB "X","."":"""+",252,200,197,194  ;    6
FC7A 582C3B3D2D            DB "X",",",";","=","-"," ",199,196,193  ;    5
FC83
FC83 0962687936            DB 09H,"b","h","y","6","5","t","g","v"  ;    4
FC8C 086E6A7537            DB 08H,"n","j","u","7","4","r","f","c"  ;    3
FC95 0A6D6B6938            DB 0AH,"m","k","i","8","3","e","d","x"  ;    2
FC9E 0B586C6F39            DB 0BH,"X","l","o","9","2","w","s","z"  ;    1
FCA7 58200D7030            DB "X"," ",0DH,"p","0","1","q","a"      ;    0
FCAF
FCAF            ;CAPS SHIFTED
FCAF
FCAF 5C07CA0E06            DB     "\",07H,202,0EH,06H,211,208,205
FCB7 582E3A7F2A            DB "X","."":",127,"*",252,210,207,204
FCC0 582C3B5F2F            DB "X",",",";","_","/"," ",209,206,203
FCC9
FCC9 0942485926            DB 09H,"B","H","Y","&","%","T","G","V"
FCD2 084E4A5527            DB 08H,"N","J","U","'","$","R","F","C"
FCDB 0A4D4B4928            DB 0AH,"M","K","I","(","#","E","D","X"
FCE4 0B584C4F29            DB 0BH,"X","L","O",")","@","W","S","Z"
FCED 58200D507E            DB "X"," ",0DH,"P",126,"!","Q","A"
FCF5
FCF5            ;SYMBOL SHIFTED
FCF5
FCF5 FE0F300E06            DB     254,0FH,"0",0EH,06H,"9","6","3"
FCFD 583E2ACF2A            DB "X",">","*",0CFH,"*",252,"8","5","2"
FD06 583C2B5F2F            DB "X","<","+","_","/"," ","7","4","1"
FD0F
FD0F 099E5E9D86            DB 09H,9EH,5EH,157,134,133,"]","\",VERIFYTOK
FD18 08A42D8187            DB 08H,164,"-",129,135,132,"[","{",168
FD21 0AA52BA580             DB 0AH,165,"+",0A5H,128,131,130,9CH,"?"
FD2A 0B5860947C            DB 0BH,"X",60H,148,124,130,">",SAVETOK,"?"
FD33 58200DBB7E            DB "X"," ",0DH,187,126,129,"<",0B1H
FD3B
FD3B
FD3B            ;CONTROL KEY TABLE - DISP/VALUE
FD3B
FD3B 1303       CKTAB:     DB 19,3            ;BRIGHT CC
FD3D 8901                  DB 137,1           ;GRAPHICS
FD3F 8A08                  DB 138,8
FD41 8801                  DB 136,1
FD43 8B07                  DB 139,7
FD45 1001                  DB 16,1            ;PEN CC
FD47 8F01                  DB 143,1
FD49 8C09                  DB 140,9
FD4B 8D07                  DB 141,7
FD4D 1102                  DB 17,2            ;PAPER CC
FD4F 8E00                  DB 142,0
FD51
```

```
FD51
FD51            ;INITIAL PALETTE COLOURS - MS GRB BITS, BRIGHT BIT, LS GRB BITS
FD51            ;                  GRBbGRB
FD51
FD51 00         INITCOLS:    DB 0000000B     ;BLACK
FD52 10                      DB 0010000B     ;BLUE
FD53 20                      DB 0100000B     ;RED
FD54 30                      DB 0110000B     ;MAGENTA
FD55 40                      DB 1000000B     ;GREEN
FD56 50                      DB 1010000B     ;CYAN
FD57 60                      DB 1100000B     ;YELLOW
FD58 78                      DB 1111000B     ;WHITE
FD59
FD59 00                      DB 0000000B     ;BRIGHT BLACK=BLACK
FD5A 11                      DB 0010001B     ;BLUE
FD5B 22                      DB 0100010B     ;RED
FD5C 33                      DB 0110011B     ;MAGENTA
FD5D 44                      DB 1000100B     ;GREEN
FD5E 55                      DB 1010101B     ;CYAN
FD5F 66                      DB 1100110B     ;YELLOW
FD60 7F                      DB 1111111B     ;WHITE
FD61
FD61                        ;COLOURS FOR 4-COLOUR MODE
FD61
FD61 00                      DB 0000000B     ;BLACK
FD62 11                      DB 0010001B     ;BLUE
FD63 22                      DB 0100010B     ;RED
FD64 7F                      DB 1111111B     ;WHITE
FD65
FD65            ;COMMAND ADDRESS TABLE.  RANGE 90H-FEH
FD65
FD65 290D       CMDADT:      DW NONSENSE     ;DIR      90 ** ALTERED
FD67 290D                    DW NONSENSE     ;FORMAT   91
FD69 290D                    DW NONSENSE     ;ERASE    92
FD6B 290D                    DW NONSENSE     ;MOVE     93
FD6D 19E0                    DW SLMVC        ;SAVE     94
FD6F 19E0                    DW SLMVC        ;LOAD     95
FD71 19E0                    DW SLMVC        ;MERGE    96
FD73 19E0                    DW SLMVC        ;VERIFY   97
FD75 03D3                    DW OPSCRN       ;OPEN     98
FD77 93D2                    DW CLSCRN       ;CLOSE    99
FD79 7120                    DW CIRCLE       ;CIRCLE   9A
FD7B AC22                    DW PLOT         ;PLOT     9B
FD7D AB38                    DW LET          ;LET      9C
FD7F B224                    DW BLITZ        ;BLITZ    9D
FD81 31F1                    DW BORDER       ;BORDER   9E
FD83 9B06                    DW CLS          ;CLS      9F
FD85
FD85 9FED                    DW COLOUR       ;PALETTE  A0
FD87 8913                    DW PERMS        ;INK      A1
FD89 8913                    DW PERMS        ;PAPER    A2
FD8B 8913                    DW PERMS        ;FLASH    A3
FD8D 8913                    DW PERMS        ;BRIGHT   A4
FD8F 8913                    DW PERMS        ;INVERSE  A5
FD91 8913                    DW PERMS        ;OVER     A6
FD93 B5D7                    DW FATPIX       ;FATPIX   A7  PIXEL WIDTH
FD95 EAD7                    DW WIDTH        ;CSIZE    A8
FD97 B0EF                    DW BGRAPHICS    ;BLOCKS   A9  BLOCK GRAPHICS/UDGS
FD99 9001                    DW MODECMD      ;MODE     AA
FD9B D028                    DW GRAB         ;GRAB     AB
FD9D 5B29                    DW PUT          ;PUT      AC
FD9F D8EE                    DW BEEP         ;BEEP     AD
FDA1 8AD8                    DW SOUND        ;SOUND    AE
FDA3 8BEC                    DW NEW          ;NEW      AF
FDA5
FDA5 F638                    DW RUN          ;RUN      B0
FDA7 E7F1                    DW STOP         ;STOP     B1
FDA9 A219                    DW CONTINUE     ;CONTINUE B2
FDAB 0139                    DW CLEAR        ;CLEAR    B3
FDAD 61D6                    DW GOTO         ;GO TO    B4
FDAF 5DD6                    DW GOSUB        ;GO SUB   B5
FDB1 CD19                    DW RETURN       ;RETURN   B6
FDB3 110E                    DW REMARK       ;REM      B7
FDB5 9D12                    DW READ         ;READ     B8
FDB7 2411                    DW DATA         ;DATA     B9
FDB9 8F34                    DW RESTORE      ;RESTORE  BA
FDBB A607                    DW PRINT        ;PRINT    BB
FDBD A307                    DW LPRINT       ;LPRINT   BC
FDBF 3A06                    DW LIST         ;LIST     BD
FDC1 3706                    DW LLIST        ;LLIST    BE
FDC3 FDF1                    DW COPY         ;DUMP     BF
FDC5
FDC5 331B                    DW FOR          ;FOR         C0
FDC7 D11B                    DW NEXT         ;NEXT        C1
FDC9 03F0                    DW PAUSE        ;PAUSE       C2
FDCB 1C21                    DW DRAW         ;DRAW        C3
FDCD A838                    DW DEFAULT      ;DEFAULT     C4
FDCF DB2E                    DW DIM          ;DIM         C5
FDD1 5B38                    DW INPUT        ;INPUT       C6
FDD3 ECF1                    DW RANDOMIZE    ;RAND        C7
FDD5 9138                    DW DEFFN        ;DEF FN      C8
FDD7 8D38                    DW DEFKEY       ;DEF KEYCODE C9
FDD9 6C32                    DW DEFPROC      ;DEF PROC    CA
FDDB DE19                    DW ENDPROC      ;END PROC    CB
FDDD 5F38                    DW RENUM        ;RENUM       CC
FDDF 8538                    DW DELETE       ;DELETE      CD
FDE1 290D                    DW NONSENSE     ;REF         CE
FDE3 290D                    DW NONSENSE     ;COPY        CF
FDE5
FDE5 290D                    DW NONSENSE     ;EDIT     D0
FDE7 6738                    DW KEYIN        ;KEYIN    D1
FDE9 3832                    DW LOCAL        ;LOCAL    D2
FDEB 1C19                    DW LOOPIF       ;LOOP IF  D3
```

```
FDED E218                   DW DO             ;DO       D4
FDEF 3219                   DW LOOP           ;LOOP     D5
FDF1 2619                   DW EXITIF         ;EXIT IF  D6
FDF3 7F1A                   DW LIF            ;LONG IF  D7
FDF5 7F1A                   DW SIF            ;SHORT IF D8
FDF7 E31A                   DW LELSE          ;LELSE    D9
FDF9 191B                   DW ELSE           ;ELSE     DA
FDFB 321B                   DW ENDIF          ;END IF   DB
FDFD BEEF                   DW KEY            ;KEY      DC
FDFF 731C                   DW ONERROR        ;ON ERROR DD
FE01 5219                   DW ON             ;ON       DE
FE03 8138                   DW GET            ;GET      DF
FE05
FE05 D9F1                   DW OUT            ;OUT      E0
FE07 0512                   DW POKE           ;POKE     E1
FE09 6212                   DW DPOKE          ;DPOKE    E2
FE0B 290D                   DW NONSENSE       ;RENAME   E3
FE0D 44EB                   DW CALLER         ;CALL     E4
FE0F 6109                   DW ROLL           ;ROLL     E5
FE11 6509                   DW SCROLL         ;SCROLL   E6
FE13 5BD2                   DW SCREEN         ;SCREEN   E7
FE15 9B3B                   DW DISPLAY        ;DISPLAY  E8
FE17 CDD8                   DW BOOT           ;BOOT     E9
FE19 E338                   DW LABEL          ;LABEL    EA
FE1B DA24                   DW FILL           ;FILL     EB
FE1D 8AF1                   DW WINDOW         ;WINDOW   EC
FE1F 51D8                   DW AUTO           ;AUTO     ED
FE21 8938                   DW POP            ;POP      EE
FE23 1A2C                   DW RECORD         ;RECORD   EF
FE25
FE25 D6EF                   DW SLDEVICE       ;DEVICE   F0
FE27 290D                   DW NONSENSE       ;PROTECT  F1
FE29 290D                   DW NONSENSE       ;HIDE     F2
FE2B 56EF                   DW ZAP            ;ZAP      F3
FE2D 98EF                   DW POW            ;POW      F4
FE2F 6CEF                   DW BOOM           ;BOOM     F5
FE31 76EF                   DW ZOOM           ;ZOOM     F6
FE33                      ;  DW NONSENSE       ;         F7
FE33                       ; DW NONSENSE       ;         F8
FE33                       ; DW NONSENSE       ;         F9
FE33                       ; DW NONSENSE       ;         FA
FE33                       ; DW NONSENSE       ;         FB
FE33                       ; DW NONSENSE       ;         FC
FE33                       ; DW NONSENSE       ;         FD
FE33                       ; DW NONSENSE       ;         FE
FE33                       ; DW NONSENSE       ;         FF (FUNCTION PREFIX)
FE33
FE33
FE33            ;TABLE FOR CHARS THAT USE ALL 8 SCANS; VALUE TO USE FOR SCAN 8, DISP TO NEXT
FE33            ;SUCH CHAR, ZERO ENDS TABLE
FE33
FE33 0878       U8TAB:      DB 08H,78H        ;"_"
FE35 0880                   DB 08H,80H        ;"¿"
FE37 00A0                   DB 00H,0A0H       ;DÚMMY
FE39 FF40                   DB 0FFH,40H       ;" "
FE3B 1C18                   DB 1CH,18H        ;"g"
FE3D 1030                   DB 10H,30H        ;"j"
FE3F 2008                   DB 20H,08H        ;"p"
FE41 0240                   DB 02H,40H        ;"q"
FE43 1C                     DB 1CH            ;"y"
FE44
FE44 005A       VVAR2:      DB 00H,5AH        ;U8TAB TERMINATOR/SVAR BASE ADDR
FE46
FE46            ;USED BY PRINT NUMBER
FE46
FE46 F0D8       SUBTAB:     DW -10000
FE48 18FC                   DW -1000
FE4A 9CFF                   DW -100
FE4C F6FF                   DW -10
FE4E                ;        DB  0  CHARSRC 1ST BYTE IS TERMINATOR!
FE4E
FE4E            ;CHARSET BY SIMON N. GOODWIN
FE4E            ;CHARACTERS USE CENTRE 6 PIXELS, CAN BE UP TO 8 PIXELS HIGH
FE4E            ;COMPRESSED FORM USES 7 5-BIT SLICES PER CHAR, TAKES 600 BYTES VS 1096
FE4E
FE4E            CHARSRC:
FE4E
FE4E 0000000004     DB 000H,000H,000H,000H,004H
FE53 210840114A     DB 021H,008H,040H,011H,04AH
FE58 000000295F     DB 000H,000H,000H,029H,05FH
FE5D 57D4A7528E     DB 057H,0D4H,0A7H,052H,08EH
FE62 295D9C8888     DB 029H,05DH,09CH,088H,088H
FE67 9CD14A22B2     DB 09CH,0D1H,04AH,022H,0B2H
FE6C 6984400000     DB 069H,084H,040H,000H,000H
FE71 0888420830     DB 008H,088H,042H,008H,030H
FE76 4108444095     DB 041H,008H,044H,040H,095H
FE7B 77DD520084     DB 077H,0DDH,052H,000H,084H
FE80 F908000000     DB 0F9H,008H,000H,000H,000H
FE85 31840003E0     DB 031H,084H,000H,003H,0E0H
FE8A 0000000006     DB 000H,000H,000H,000H,006H
FE8F 3042222210     DB 030H,042H,022H,022H,010H
FE94 74675CC5C4     DB 074H,067H,05CH,0C5H,0C4H
FE99 61084239D1     DB 061H,008H,042H,039H,0D1H
FE9E 08888FBA21     DB 008H,088H,08FH,0BAH,021H
FEA3 3062E11952     DB 030H,062H,0E1H,019H,052H
FEA8 F885F87821     DB 0F8H,085H,0F8H,078H,021H
FEAD 8B8C887A31     DB 08BH,08CH,088H,07AH,031H
FEB2 77C2222210     DB 077H,0C2H,022H,022H,010H
FEB7 7462E8C5CE     DB 074H,062H,0E8H,0C5H,0CEH
FEBC 8C5E113000     DB 08CH,05EH,011H,030H,000H
FEC1 3180630006     DB 031H,080H,063H,000H,006H
FEC6 300C208888     DB 030H,00CH,020H,088H,088H
FECB 2082007C1F     DB 020H,082H,000H,07CH,01FH
```

```
FED0 0020820888     DB 000H,020H,082H,008H,088H
FED5 83A2111004     DB 083H,0A2H,011H,010H,004H
FEDA 74675BC1CE     DB 074H,067H,05BH,0C1H,0CEH
FEDF 8C7F18C7D1     DB 08CH,07FH,018H,0C7H,0D1H
FEE4 8FA31F3A30     DB 08FH,0A3H,01FH,03AH,030H
FEE9 8422EF4631     DB 084H,022H,0EFH,046H,031H
FEEE 8C7DF843D0     DB 08CH,07DH,0F8H,043H,0D0H
FEF3 87FF087A10     DB 087H,0FFH,008H,07AH,010H
FEF8 83A3084E2E     DB 083H,0A3H,008H,04EH,02EH
FEFD 8C63F8C62E     DB 08CH,063H,0F8H,0C6H,02EH
FF02 2108423821     DB 021H,008H,042H,038H,021H
FF07 0843174654     DB 008H,043H,017H,046H,054H
FF0C C525184210     DB 0C5H,025H,018H,042H,010H
FF11 843F1DD631     DB 084H,03FH,01DH,0D6H,031H
FF16 8C631CD671     DB 08CH,063H,01CH,0D6H,071H
FF1B 8BA318C62E     DB 08BH,0A3H,018H,0C6H,02EH
FF20 F463E8420E     DB 0F4H,063H,0E8H,042H,00EH
FF25 8C635937D1     DB 08CH,063H,059H,037H,0D1H
FF2A 8FA928BA30     DB 08FH,0A9H,028H,0BAH,030H
FF2F 7062EF9084     DB 070H,062H,0EFH,090H,084H
FF34 210918C631     DB 021H,009H,018H,0C6H,031H
FF39 8BA318C62A     DB 08BH,0A3H,018H,0C6H,02AH
FF3E 246318D6AA     DB 024H,063H,018H,0D6H,0AAH
FF43 8C54454631     DB 08CH,054H,045H,046H,031H
FF48 8A884213E1     DB 08AH,088H,042H,013H,0E1H
FF4D 11110F9C84     DB 011H,011H,00FH,09CH,084H
FF52 2108784104     DB 021H,008H,078H,041H,004H
FF57 1043C21084     DB 010H,043H,0C2H,010H,084H
FF5C 2708EA9084     DB 027H,008H,0EAH,090H,084H
FF61 2000000000     DB 020H,000H,000H,000H,000H
FF66 3251C423E0     DB 032H,051H,0C4H,023H,0E0H
FF6B 0382F8BE10     DB 003H,082H,0F8H,0BEH,010H
FF70 F4631F000F     DB 0F4H,063H,01FH,000H,00FH
FF75 8420F085F1     DB 084H,020H,0F0H,085H,0F1H
FF7A 8C5E003A3F     DB 08CH,05EH,000H,03AH,03FH
FF7F 83CC947908     DB 083H,0CCH,094H,079H,008H
FF84 4000F8C5E1     DB 040H,000H,0F8H,0C5H,0E1H
FF89 843D18C624     DB 084H,03DH,018H,0C6H,024H
FF8E 0108420880     DB 001H,008H,042H,008H,080H
FF93 2108424211     DB 021H,008H,042H,042H,011H
FF98 9725121084     DB 097H,025H,012H,010H,084H
FF9D 210C006AB5     DB 021H,00CH,000H,06AH,0B5H
FFA2 AD400F4631     DB 0ADH,040H,00FH,046H,031H
FFA7 8800E8C62E     DB 088H,000H,0E8H,0C6H,02EH
FFAC 003D18FA00     DB 000H,03DH,018H,0FAH,000H
FFB1 03E3178400     DB 003H,0E3H,017H,084H,000H
FFB6 B66108000F     DB 0B6H,061H,008H,000H,00FH
FFBB 8383E211E4     DB 083H,083H,0E2H,011H,0E4H
FFC0 2106004631     DB 021H,006H,000H,046H,031H
FFC5 8BC008C62A     DB 08BH,0C0H,008H,0C6H,02AH
FFCA 200118C6AA     DB 020H,001H,018H,0C6H,0AAH
FFCF 0022A22A20     DB 000H,022H,0A2H,02AH,020H
FFD4 0463178400     DB 004H,063H,017H,084H,000H
FFD9 F8888F8884     DB 0F8H,088H,08FH,088H,084H
FFDE 4108221084     DB 041H,008H,022H,010H,084H
FFE3 2108821044     DB 021H,008H,082H,010H,044H
FFE8 220AA00000     DB 022H,00AH,0A0H,000H,000H
FFED 0782D4B43E     DB 007H,082H,0D4H,0B4H,03EH
FFF2 00000F7BC0     DB 000H,000H,00FH,07BH,0C0H
FFF7 F7BC0000C8     DB 0F7H,0BCH,000H,000H,0C8H
FFFC 747F0780       DB 074H,07FH,007H,080H
0000
0000            ;    DB "AW,BG"
0000
0000            ;          DS 0FFFFH-$,0
0000
0000
0000                       SEND COMPUTER1
0000 8000=                 END 8000H     ;dummy value
```