

```
;*****
;*
;*      SmartBASIC 1.x version 20Y      *
;*      (c) 1991, 1997                  *
;*      by Richard F. Drushel          *
;*                                       *
;*      regenerated Z80 assembly source *
;*                                       *
;*****
```

;version history

```
;9711.09      ;after much work, made compatible with
Z80ASM+ assembler (SLR)
;              ;undid all the hacks necessary for the
junky cross-assembler
;              ;aded EOS global equates
;              ;lined up comments
;              ;added a few comments along the way
;              ;still missing many comments that I have
in manuscript
;              ;has one bugfix not in the current SB1.x
distribution
;              ; (see A22538: discussion)
;              ;otherwise exactly duplicates the SB1.x
binary, patches and all
;
;9405.08      ;MASTER.ASM assembly file
;              ;combines all the separate modules into
one file
;              ;did this to try to get Z80ASM+ to
assemble it
;              ;so many syntax errors that it was too
daunting to fix
;
;9111.19      ;final regenerated assembly source
;              ;this was done by hand from UNASMHEX.EXE
listings in
```

```
;           ; WordPerfect 5.0 for MS-DOS on a
286-12 MHz laptop
;           ;multiple module files for MS-DOS cross
assembler obtained
;           ; from wsmr-simtel20.army.mil ftp site
;           ;has hacks to accommodate quirks in
assembler, e.g.
;           ; gives wrong machine code for all
8-bit SUB instructions
;           ; needs weird index register syntax,
e.g. 9(IY) for (IY+9)
;           ; can't have DB "text", must use .ASCII
"text"
;           ; uses . for current program counter
instead of $
;           ; uses = instead of EQU for equates
;           ; programmer must distinguish immediate
data from label
;           ; references using # prefix
;           ;all in all, it works, but it's really
ugly
```

```
;*****
;*
;*          TERMS OF SOFTWARE RELEASE          *
;*
;*****
```

```
;I'm putting the source for SmartBASIC 1.x, such as
it
;is, out for everyone in the ADAM community to look
at.
;While it can be used to generate your own usable
SB1.x
;binary (without buying the software from me), I'm
hoping
;that you won't do that. The SB1.x manual and
sample
```

```
;programs are not included, so unless you can read
the
;code well enough to figure out the commands (or
have
;pirated a manual), it won't be much practical good
to
;you.
;
;The reasons I'm releasing the source are:
;
;(1) I don't want it to get lost in some accident,
;    so multiple copies can guard against that; and
;
;(2) I want the few programmer-inclined among us to
;    be able to see how it works, and maybe be
;    inspired to create their own ADAM software.
;
;Feel free to use any useful code in the listing.
Feel
;free to point out (and maybe even fix) any bugs
that
;you may find.
;
;Note that the majority of the listing is currently
uncommented.
;I have many more comments in manuscript, but have
not had the
;time to type them all in. (Remember that SB1.x was
created by
;hand-assembled patches POKEd into RAM, then
BSAVEd.) I reserve
;the right to amend the listing by adding comments
as I have time
;(if I have time).
;
;Finally, note that some significant SB1.0 bugs
still remain
;uncorrected. The most notable is the screwed-up
```

```
file I/O
;which doesn't allow you to have 2 files open at the
same time
;and successfully read from one while writing to the
other.
;Maybe someday I can try to port in the *working*
version from
;SmartBASIC 2.0...but that code requires a working
_POSITION_FILE
;EOS function call, which doesn't exist in EOS-5
(only in EOS-7).
;If someone wants a place to start bugfixing, I'd
really like to
;see working file I/O.
```

```
*****
;*
;*          ACKNOWLEDGEMENTS          *
;*
;*****
```

```
;(1)  the original Lazer Microsystems programmers:
;      Randy Hyde (president of LMS, author of the
floating-point math code)
;      Joel Lagerquist
;      Johnny Fitzgerald
;      Robert Greenberg
;      Bruce (Winston) Hendrickson
;
;(2)  Ben Hinkle, for his invaluable SmartBASIC 1.0
;      disassembly in "The Hacker's Guide to ADAM
Volume II"
;
;(3)  Herman Mason, George Koczwar, Alan Neeley,
Ron Collins,
;      Ron Mitchell, Steve Major, Guy Cousineau, Tony
Morehen,
;      and Chris Braymen for providing hardware,
```

software,
; testing, and encouragement.

;
;Richard F. Drushel, Ph.D. (drushel@apk.net)
;Sunday, 9 November 1997

;*****

;EOS jump table equates.

_EOS_START	EQU	0FC30H	
_END_RD_KBD	EQU	0FC4BH	
_PR_CH	EQU	0FC66H	
_START_RD_KBD	EQU	0FCA8H	
_INIT_TAPE_DIR	EQU	0FCBDH	
_OPEN_FILE	EQU	0FCC0H	
_CLOSE_FILE	EQU	0FCC3H	
_RESET_FILE	EQU	0FCC6H	
_MAKE_FILE	EQU	0FCC9H	
_QUERY_FILE	EQU	0FCCCH	
_SET_FILE	EQU	0FCCFH	
_READ_FILE	EQU	0FCD2H	
_WRITE_FILE	EQU	0FCD5H	
_SET_DATE	EQU	0FCD8H	
_RENAME_FILE	EQU	0FCDEH	
_DELETE_FILE	EQU	0FCE1H	
_GOTO_WP	EQU	0FCE7H	
_TRIM_FILE	EQU	0FCEDH	
_CHECK_FCB	EQU	0FCF0H	
_READ_BLOCK	EQU	0FCF3H	
_WRITE_BLOCK	EQU	0FCF6H	
_CV_A	EQU	0FD0EH	;a RET in all
EOS versions!			
PUT_ASCII	EQU	0FD17H	
WRITE_VRAM	EQU	0FD1AH	
READ_VRAM	EQU	0FD1DH	
WRITE_REGISTER	EQU	0FD20H	

```
READ_REGISTER      EQU    0FD23H
FILL_VRAM          EQU    0FD26H
INIT_TABLE        EQU    0FD29H
CALC_OFFSET       EQU    0FD32H
PX_TO_PTRN_POS    EQU    0FD35H
LOAD_ASCII        EQU    0FD38H
POLLER            EQU    0FD3EH
```

;EOS global data equates.

```
REV_NUM           EQU    0FD60H      ;EOS revision
number
CURRENT_DEV      EQU    0FD6FH      ;current
(default) device number
EOS_YEAR         EQU    0FDE0H      ;year byte
(2-digit BCD)
EOS_MONTH        EQU    0FDE1H      ;month byte
(2-digit BCD)
EOS_DAY          EQU    0FDE2H      ;day byte
(2-digit BCD)
FCB_DATA_ADDR    EQU    0FDFFH      ;pointer to
start of FCB buffers
SOUNDPORT        EQU    0FC2FH      ;sound port
```

;EOS error message equates.

```
NO_FILE_ERR      EQU    5
NO_FCB_ERR       EQU    7
FULL_DIR_ERR     EQU    12
FULL_TAPE_ERR    EQU    13
DELETE_ERR       EQU    16
RANGE_ERR        EQU    17
```

;SmartBASIC equates

```
SB10_STACK      EQU    54142
SB1X_STACK      EQU    54144      ;I can't
remember why this is different
```

good reason... ;there may be no

;*****

ORG 0

;*****

;BLOAD header for SmartBASIC.

;Take this out if you want a stand-alone binary.

DB 1,0,2 ;header

DW A0 ;load address

;*****

.PHASE 0

A0:

JP A32609 ;Cold entry point from boot
;later changed to RET

A2 EQU A0+2 ;used by clock wakeups

;*****

NOP

NOP

NOP

NOP

NOP

JP A6345 ;RST 08H

;*****

NOP

NOP

NOP

NOP

NOP

JP A6345 ;RST 10H

;*****

```

NOP
NOP
NOP
NOP
NOP
JP      A6345      ;RST 18H
;*****
NOP
NOP
NOP
NOP
NOP
JP      A6345      ;RST 20H
;*****
NOP
NOP
NOP
NOP
NOP
JP      A6345      ;RST 28H
;*****
NOP
NOP
NOP
NOP
NOP
JP      A6345      ;RST 30H
;*****
NOP
NOP
NOP
NOP
NOP
JP      A73        ;RST 38H
;*****
NOP
NOP
NOP
```



```

        NOP
;*****
A63:
        DB      1          ;weekday (1=Sunday)
A64:
        DB      0          ;current TERM
;*****
;WINDOW data.
;
A65:
        DB      0          ;WINDOW left margin
A66:
        DB      0          ;WINDOW right margin
A67:
        DB      0          ;WINDOW top margin
A68:
        DB      0          ;WINDOW bottom margin
A69:
        DB      1          ;min left margin
A70:
        DB      31         ;max right margin
A71:
        DB      0          ;min top margin
A72:
        DB      23         ;max bottom margin
;*****
A73:
        RETI             ;default exit for RST 38H
;*****
;
;
A75:
        CP      34
        JP      NZ,A7998
        LD      HL,A30623
        JP      A28890
;*****
;Table of month lengths for DATE.

```

```

;
A86:
    DB      32,29,32,31,32,31,32,32,31,32,31,32
;*****
;TIME data.
;
A98:
    DB      0                ;hour
A99:
    DB      0                ;minute
A100:
    DB      0                ;second
A101:
    DB      0                ;sixtieth of second
;*****
;NMI routine.
;
A102:
    RETN                    ;later changed
;*****
    LD      A,D              ;some garbage which gets
overwritten
    NOP                    ;NOP from patch
A106:
    OR      A
    JR      NZ,A120
    DEC    A
    LD      (A17010),A
    LD      A,(A17009)
    OR      A
    CALL   NZ,A123
A120:
    POP    AF
    RETN
;*****
A123:
    PUSH   BC
    LD     A,(A17011)

```

```
DEC    A
JR     NZ,A159
PUSH  DE
PUSH  HL
PUSH  IX
PUSH  IY
LD     HL,(A17012)
LD     A,16
XOR   H
LD     H,A
LD     (A17012),HL
LD     A,2
CALL  INIT_TABLE
POP   IY
POP   IX
POP   HL
POP   DE
LD     A,12
```

A159:

```
LD     (A17011),A
XOR   A
LD     (A17010),A
CALL  READ_REGISTER
POP   BC
RET
```

;*****

A171:

```
PUSH  HL
LD     HL,A101
LD     A,(HL)
INC   A
CP    60
JR     NZ,A208
XOR   A
LD     (HL),A
DEC   HL
LD     A,(HL)
INC   A
```

```

        CP      60
        JR      NZ ,A208
        XOR     A
        LD      (HL) ,A
        DEC     HL
        LD      A , (HL)
        INC     A
        CP      60
        JR      NZ ,A208
        XOR     A
        LD      (HL) ,A
        DEC     HL
        LD      A , (HL)
        INC     A
        CP      24
        JR      Z ,A213
A208:
        LD      (HL) ,A
A209:
        POP     HL
        JP      A31314
;*****
A213:
        CALL   A32281
        NOP
        NOP
        PUSH   BC
        LD     A , (EOS_MONTH)
        LD     C , A
        LD     B , 0
        ADD   HL , BC
        LD     B , (HL)
        LD     HL , EOS_DAY
        LD     A , (HL)
        INC   A
        CP    B
        POP   BC
        JR    C , A208

```

```

        LD      A,1
        LD      (HL),A
        DEC    HL
        LD      A,(HL)
        INC    A
        CP     13
        JR     C,A208
        LD      A,1
        LD      (HL),A
        DEC    HL
        INC    (HL)
        JR     A209
;*****
A253:
        DB     0                ;current CLK
A254:
        DW     A6345;          ;current ADDR(5) user NMI
routine
                                ;A6345 is a RET
;*****
;Cold start entry.
;
A256:
        JP     A16035
;*****
        DB     195, 79, 1      ;could be version 1.79 ??
(or 01.4F hex)
                                ;or more likely, JP 0179H
;*****
;Powers of 10.
;
A262:
        DW     10000
        DW     1000
        DW     100
        DW     10
        DW     1
;*****

```

;Primary word table. Format: token, parse pointer,
length, word.

;

A272:

DB 1
DW A938
DB 0 ;implicit LET

;

DB 2
DW A941
DB 5
DB "GOSUB"

;

DB 3
DW A941
DB 4
DB "GOTO"

;

DB 4
DW A983
DB 5
DB "INPUT"

;

DB 5
DW A938
DB 3
DB "LET"

;

DB 6
DW A977
DB 4
DB "NEXT"

;

DB 7
DW A996
DB 5
DB "PRINT"

;

```
DB      8
DW      A980
DB      4
DB      "READ"
;
DB      9
DW      A953
DB      3
DB      "REM"
;
DB      10
DW      A959
DB      3
DB      "FOR"
;
DB      11
DW      A968
DB      2
DB      "IF"
;
DB      12
DW      A956
DB      4
DB      "DATA"
;
DB      13
DW      A980
DB      3
DB      "DIM"
;
DB      14
DW      A986
DB      2
DB      "ON"
;
DB      15
DW      A991
DB      5
```

```
DB      "ONERR "  
;  
DB      16  
DW      A976  
DB      4  
DB      "STOP "  
;  
DB      17  
DW      A944  
DB      6  
DB      "RETURN "  
;  
DB      18  
DW      A976  
DB      3  
DB      "END "  
;  
DB      19  
DW      A973  
DB      3  
DB      "DEF "  
;  
DB      20  
DW      A976  
DB      5  
DB      "CLEAR "  
;  
DB      21  
DW      A944  
DB      6  
DB      "RESUME "  
;  
DB      22  
DW      A976  
DB      3  
DB      "NEW "  
;  
DB      23
```



```
DW      A976
DB      3
DB      "POP"
;

DB      24
DW      A944
DB      3
DB      "RUN"
;

DB      25
DW      A950
DB      4
DB      "LIST"
;

DB      26
DW      A976
DB      5
DB      "TRACE"
;

DB      27
DW      A976
DB      7
DB      "NOTRACE"
;

DB      28
DW      A947
DB      3
DB      "DEL"
;

DB      29
DW      A30266
DB      4
DB      "CALL"
;

DB      30
DW      A976
DB      4
DB      "CONT"
```

```
;
    DB      31
    DW      A976
    DB      6
    DB      "CLRERR"
;
    DB      32
    DW      A1002
    DB      3
    DB      "GET"
;
    DB      33
    DW      A10104
    DB      4
    DB      "POKE"
;
    DB      34
    DW      A944
    DB      7
    DB      "RESTORE"
;
    DB      35
    DW      A976
    DB      4
    DB      "HOME"
;
    DB      36
    DW      A1024
    DB      4
    DB      "DRAW"
;
    DB      37
    DW      A1024
    DB      5
    DB      "XDRAW"
;
    DB      38
    DW      A976
```

```
DB 5
DB "FLASH"
;
DB 39
DW A976
DB 7
DB "INVERSE"
;
DB 40
DW A976
DB 6
DB "NORMAL"
;
DB 41
DW A976
DB 4
DB "TEXT"
;
DB 42
DW A976
DB 2
DB "GR"
;
DB 43
DW A976
DB 3
DB "HGR"
;
DB 44
DW A976
DB 4
DB "HGR2"
;
DB 45
DW A1029
DB 4
DB "HLIN"
;
```

```
DB 46
DW A1029
DB 4
DB "VLIN"
;
DB 47
DW A1014
DB 5
DB "HPLOT"
;
DB 48
DW A1017
DB 4
DB "PLOT"
;
DB 49
DW A999
DB 4
DB "HTAB"
;
DB 50
DW A999
DB 4
DB "VTAB"
;
DB 51
DW A976
DB 6
DB "SHLOAD"
;
DB 52
DW A1002
DB 6
DB "RECALL"
;
DB 53
DW A1002
DB 5
```

```
DB      "STORE"  
;  
DB      54  
DW      A1005  
DB      4  
DB      "WAIT"  
;  
DB      55  
DW      A1040  
DB      5  
DB      "SPEED"  
;  
DB      56  
DW      A1040  
DB      3  
DB      "ROT"  
;  
DB      57  
DW      A1040  
DB      5  
DB      "SCALE"  
;  
DB      58  
DW      A1083  
DB      5  
DB      "COLOR"  
;  
DB      59  
DW      A1040  
DB      6  
DB      "HCOLOR"  
;  
DB      60  
DW      A1045  
DB      2  
DB      "IN"  
;  
DB      61
```

```

        DW      A1045
        DB      2
        DB      "PR"
;
        DB      62
        DW      A1050
        DB      5
        DB      "HIMEM"
;
        DB      63
        DW      A1050
        DB      5
        DB      "LOMEM"
;
        DB      64
        DW      A976
        DB      5
        DB      "BREAK"
;
        DB      65
        DW      A976
        DB      7
        DB      "NOBREAK"
;
        DB      7
        DW      A996
        DB      1
        DB      "?"
;
        DB      66
        DW      A953
        DB      1
        DB      "&"
;
        DB      0 ;end of table

```

```

;*****

```

```

;Secondary word table.  Format:  token, length,
word.

```

```
;
A818:
    DB    160, 1
    DB    "+"
;
    DB    161, 1
    DB    "-"
;
    DB    162, 1
    DB    "*"
;
    DB    163, 1
    DB    "/"
;
    DB    164, 1
    DB    "^"
;
    DB    165, 1
    DB    "<"
;
    DB    166, 1
    DB    ">"
;
    DB    167, 2
    DB    "<="
;
    DB    168, 2
    DB    ">="
;
    DB    169, 2
    DB    "<>"
;
    DB    170, 1
    DB    "="
;
    DB    171, 3
    DB    "AND"
;
```

```
DB 172, 2
DB "OR"
;
DB 173, 3
DB "NOT"
;
DB 174, 4
DB "GOTO"
;
DB 175, 5
DB "GOSUB"
;
DB 176, 4
DB "STEP"
;
DB 177, 2
DB "AT"
;
DB 178, 4
DB "THEN"
;
DB 179, 4
DB "THEN"
;
DB 180, 2
DB "TO"
;
DB 181, 1
DB ":"
;
DB 182, 1
DB "#"
;
DB 183, 1
DB "("
;
DB 184, 1
DB ")"
```



```

;
    DB      185, 1
    DB      ", "
;
    DB      186, 1
    DB      "; "
;
    DB      167, 2
    DB      "=<"
;
    DB      168, 2
    DB      "=>"
;
    DB      169, 2
    DB      "><"
;
    DB      0 ;end of table
;*****
;Parse vector table.  Format:  number of vectors,
vectors.
;
A938:
    DB      1
    DW      A15020
A941:
    DB      1
    DW      A15756
A944:
    DB      1
    DW      A15232
A947:
    DB      1
    DW      A15247
A950:
    DB      1
    DW      A15243
A953:
    DB      1

```

A956: DW A15817
DB 1
DW A15814
A959: DB 4
DW A14991, A15963, A14875, A15093
A968: DB 2
DW A14947, A15035
A973: DB 1
DW A15125
A976: DB 0
A977: DB 1
DW A15567
A980: DB 1
DW A15574
A983: DB 1
DW A15543
A986: DB 2
DW A14875, A15209
A991: DB 2
DW A15991, A15756
A996: DB 1
DW A15580
A999: DB 1
DW A14875
A1002: DB 1

```

        DW      A15364
A1005:
        DB      4
        DW      A14875, A15939, A14875, A14969
A1014:
        DB      1
        DW      A15102
A1017:
        DB      3
        DW      A14875, A15939, A14875
A1024:
        DB      2
        DW      A14875, A14976
A1029:
        DB      5
        DW      A14875, A15939, A14875, A15977, A14875
A1040:
        DB      2
        DW      A15926, A14875
A1045:
        DB      1
        DW      A30552
```

```
;*****
```

```
    NOP
```

```
    NOP
```

```
;*****
```

```
;Parse vector table.
```

```
A1050:
```

```
    DB      2
```

```
    DW      A15911, A14875
```

```
;*****
```

```
;Lazer Microsystems programmer Joel Lagerquist says  
that this
```

```
;refers to fellow programmer Johnny Fitzgerald's  
girlfriend.
```

```
;(JKL to RFD, spring 1997)
```

```

        DB      4
        DB      "Hi Cathy"
;
A1064:
        DB      18
        DB      "FATAL SYSTEM ERROR"
;*****
;Parse vector table.
;
A1083:
        DB      2
        DW      A1111, A14875
;*****
        DB      32      ;      garbage here
        DB      32      ;
        DB      67      ;C
        DB      111     ;o
;*****
A1092:
        PUSH    AF
        LD      A,B
        CP      178
A1096:
        JR      Z,A1103
        POP     AF
        CALL    C,A11994
        RET
;*****
A1103:
        POP     AF
        LD      A,(A16148)
        OR      A
        JP      A13355
;*****
A1111:
        LD      A,170
        CALL    A14581

```

```

        RET      C
A1117:
        CALL    A14575
        JR      C,A1129
A1122:
        CALL    A11921
;
        DB      3
        DB      "' (' "
;*****
A1129:
        CALL    A14875
        CALL    A15532
        JP      A15926
;*****
;NOMERGE command.
;Restores LOAD command to its original form,
disabling MERGE.
;
A1138:
        LD      HL,6356      ;NEW command
        JP      A29902      ;patch the code
;*****
        NOP
;*****
;System characters. Format: length, word.
;
A1145:
        DB      1
A1146:
        DB      "]"          ;prompt
;
        DB      0            ;unknown, patched out
;
A1148:
        DB      1
        DB      ":"          ;statement separator
;

```

```
A1150:
    DB      1, 13          ;carriage return
;*****
;Error messages.  Format:  length, message.
;
A1152:
    DB      16
    DB      "NEXT without FOR"
;
    DB      6
    DB      "Syntax"
;
    DB      20
    DB      "RETURN without GOSUB"
;
    DB      11
    DB      "Out of DATA"
;
    DB      16
    DB      "Illegal Quantity"
;
    DB      8
    DB      "Overflow"
;
A1235:
    DB      13
    DB      "Out of Memory"
;
    DB      14
    DB      "Stack Overflow"
;
    DB      19
    DB      "Undefined Statement"
;
    DB      13
    DB      "Bad Subscript"
;
    DB      19
```

```

        DB      "Redimensioned Array"
;
        DB      14
        DB      "Divide by Zero"
;
A1333:
        DB      13
        DB      "Type Mismatch"
;
        DB      15
        DB      "String Too Long"
;
        DB      19
        DB      "Formula Too Complex"
;
        DB      18
        DB      "Undefined Function"
;
        DB      24
        DB      "Incorrect Function Usage"
;
        DB      12
        DB      "Illegal Mode"
;
        DB      0
;
        DB      0
;
        DB      5
        DB      "Break"
;
        DB      14
        DB      "Can't Continue"
;*****
;Old ERRNUM numbers, for corresponding messages.
;
        DB      0
        DB      16

```

```
DB 22
DB 42
DB 53
DB 69
DB 77
DB 77
DB 90
DB 107
DB 120
DB 133
DB 163
DB 176
DB 191
DB 224
DB 16
DB 149
DB 254
DB 255
DB 255
DB 210
```

```
;*****
```

```
A1485:
```

```
LD A, (A16148)
CALL A11994
JP A12959
```

```
;*****
```

```
;Parse vector table.
```

```
A1494:
```

```
DB 1
DW A14947
```

```
;*****
```

```
A1497:
```

```
DB 0 ;PUT toggle (255=active,
0=inactive)
```

```
A1498:
```

```
DB 1 ;current ROT
;@@@ by mixup @@@ also
```

```
INPUT/LINPUT toggle @@@
```



```
;255=LINPUT, 0=INPUT  
;will need to fix this in
```

```
next SB1.x revision
```

```
A1499:
```

```
    DB      20          ;SB1.x revision byte
```

```
;*****
```

```
;Load HL with number from crunch code.
```

```
;
```

```
A1500:
```

```
    CALL    A5939  
    OR      A  
    JP      NZ,A7939  
    CALL    A2354  
    JP      C,A7936  
    LD      A,H  
    OR      H  
    RET     Z  
    JP      A7936
```

```
;*****
```

```
;Load BC with integer 0-65535 from crunch code.
```

```
;
```

```
A1519:
```

```
    CP      138  
    JP      C,A1538  
    JP      Z,A1544  
    EXX  
    DEC     C  
    DEC     C  
    EXX  
    INC     DE  
    LD      A,(DE)  
    LD      C,A  
    INC     DE  
    LD      A,(DE)  
    LD      B,A  
    RET
```

```
;*****
```

```
A1538:
```

```

        AND    15
        LD     C,A
        LD     B,0
        RET
;*****
A1544:
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        LD    C,A
        LD    B,0
        RET
;*****
;Print FPA1 in decimal.
;
A1553:
        PUSH  DE
        PUSH  HL
        PUSH  BC
        CALL  A3260
        LD    HL,A16246
        CALL  A12110
        POP   BC
        POP   HL
        POP   DE
        RET
;*****
;Add FPA1 with (HL).  Result in FPA1.
;
A1569:
        CALL  A4375
        JP    A1583
;*****
;Subtract FPA2 from FPA1.  Result in FPA1.
;
A1575:

```

```

        LD      A, (A16174)
        XOR     128
        LD      (A16174), A
;*****
;Add FPA1 and FPA2.  Result in FPA1.
;
A1583:
        LD      A, (A16166)
        LD      HL, (A16174)
        SUB     H
        JR      C, A1606
        LD      BC, (A16171)
        LD      DE, (A16173)
        LD      HL, (A16165)
        JP      A1619
;*****
A1606:
        LD      BC, (A16162)
        LD      DE, (A16164)
        CALL   A1832
        NEG
A1619:
        CP      H
        RET     Z
        EX     AF, AF'
        LD      A, (A16165)
        LD      (A16151), A
        XOR     D
        EX     AF, AF'
        LD      HL, A16165
        SET    7, (HL)
        SET    7, D
        LD      L, 0
A1639:
        LD      H, A
        SRL    H
        SRL    H
        SRL    H

```

```

    JR      Z,A1657
A1648:
    LD      L,C
    LD      C,B
    LD      B,E
    LD      E,D
    LD      D,0
    DEC     H
    JR      NZ,A1648
A1657:
    AND     7
    JR      Z,A1674
A1661:
    SRL     D
    RR      E
    RR      B
    RR      C
    RR      L
    DEC     A
    JR      NZ,A1661
A1674:
    EX      AF,AF'
    LD      A,L
    JP      M,A1739
    LD      HL,(A16162)
    ADD     HL,BC
    LD      B,H
    LD      C,L
    LD      HL,(A16164)
    ADC     HL,DE
    JR      NC,A1704
    RR      H
    RR      L
    RR      B
    RR      C
    RRA
A1701:
    CALL   A1875

```

A1704:

RLA

A1705:

JR NC,A1723

INC C

JP NZ,A1723

INC B

JP NZ,A1723

INC L

JP NZ,A1723

INC H

CALL Z,A1875

A1723:

LD A,(A16151)

RL H

RLA

RR H

LD (A16164),HL

LD (A16162),BC

RET

;*****

A1739:

NEG

LD HL,(A16162)

SBC HL,BC

LD B,H

LD C,L

LD HL,(A16164)

SBC HL,DE

LD E,A

JR NC,A1780

LD A,(A16151)

XOR 128

LD (A16151),A

XOR A

SUB E

EX DE,HL

LD HL,0

```

        SBC     HL,BC
        LD      B,H
        LD      C,L
        LD      HL,0
        SBC     HL,DE
        LD      E,A
A1780:
        LD      A,(A16166)
        LD      D,A
A1784:
        LD      A,H
        OR      A
        JR      NZ,A1820
        LD      H,L
        LD      L,B
        LD      B,C
        LD      C,E
        LD      E,0
        LD      A,D
        SUB     8
        LD      D,A
        JR      Z,A1802
        JR      NC,A1784
A1802:
        XOR     A
A1803:
        LD      (A16166),A
        RET
;*****
A1807:
        DEC     D
        JR      Z,A1802
        SLA     E
        RL      C
        RL      B
        RL      L
        RL      H
A1820:

```

```

        JP      P,A1807
        LD      A,D
        LD      (A16166),A
        RL      E
        JP      A1705
;*****
;Load FPA2 to FPA1.
;
A1832:
        LD      HL,(A16171)
        LD      (A16162),HL
        LD      HL,(A16173)
        LD      (A16164),HL
        LD      HL,(A16174)
        LD      (A16165),HL
        RET
;*****
;Find sign.
;
A1851:
        LD      DE,(A16165)
        LD      HL,(A16174)
        LD      A,L
        XOR     E
        LD      (A16151),A
        SET     7,E
        SET     7,L
        LD      (A16174),HL
        LD      (A16165),DE
        RET
;*****
;Multiply FPA1*2.
;
A1875:
        EX     DE,HL
        LD      HL,A16166
        INC     (HL)
        JP     Z,A7921

```

```

        EX      DE,HL
        RET
;*****
;Multiply (HL)*FPA1.
;
A1885:
        CALL   A4375
;*****
;Multiply FPA1*FPA2.
;
A1888:
        CALL   A1851
        LD     A,D
        OR     A
        RET    Z
        LD     A,H
        OR     A
        JR     Z,A1803
        ADD   A,D
        JR     NC,A1913
        SUB   129
        JP    M,A1917
        CP   127
        JR     Z,A1917
        JP    A7921
;*****
A1913:
        SUB   129
        JR     C,A1802
A1917:
        LD    (A16166),A
        LD    BC,0
        LD    D,B
        LD    E,C
        LD    A,(A16162)
        CALL  A1968
        LD    A,(A16163)
        CALL  A1968

```



```
LD      A, (A16164)
CALL   A1968
LD      A, (A16165)
CALL   A1968
EX      AF, AF '
EX      DE, HL
BIT     7, H
JP      NZ, A1701
RLA
RL      C
RL      B
RL      L
RL      H
JP      A1704
```

```
;*****
```

```
A1968:
```

```
OR      A
JR      Z, A2009
SCF
RRA
```

```
A1973:
```

```
JR      NC, A1993
EX      AF, AF '
LD      HL, (A16171)
ADD     HL, BC
LD      B, H
LD      C, L
LD      HL, (A16173)
ADC     HL, DE
EX      DE, HL
RR      D
JP      A1996
```

```
;*****
```

```
A1993:
```

```
EX      AF, AF '
SRL     D
```

```
A1996:
```

```
RR      E
```

```

RR      B
RR      C
RRA
EX      AF,AF'
SRL     A
JR      NZ,A1973
RET
;*****
A2009:
EX      AF,AF'
LD      A,C
EX      AF,AF'
LD      C,B
LD      B,E
LD      E,D
LD      D,0
RET
;*****
;Divide FPA1 by FPA2.
;
A2018:
CALL    A1851
LD      A,H
OR      A
JP      Z,A7918
LD      A,D
OR      A
JP      Z,A1803
SUB     H
JP      NC,A2047
ADD     A,128
JP      P,A2052
CP      128
JR      Z,A2052
JP      A1802
;*****
A2047:
ADD     A,128

```

```

        JP      P,A7921
A2052:
        LD      (A16166),A
        LD      HL,0
        LD      (A16152),HL
        LD      (A16154),HL
        LD      (A16167),HL
        LD      (A16169),HL
        LD      (A16158),HL
        LD      (A16160),HL
        LD      D,32
A2078:
        LD      HL,A16167
        LD      BC,A16158
        LD      A,(BC)
        SUB     (HL)
        LD      (BC),A
        INC     BC
        INC     HL
        LD      A,(BC)
        SBC     A,(HL)
        LD      (BC),A
        INC     BC
        INC     HL
        LD      A,(BC)
        SBC     A,(HL)
        LD      (BC),A
        INC     BC
        INC     HL
        LD      A,(BC)
        SBC     A,(HL)
        LD      (BC),A
        INC     BC
        INC     HL
        LD      A,(BC)
        SBC     A,(HL)
        LD      (BC),A
        INC     BC

```



```

RR      (HL)
DEC     HL
RR      (HL)
DEC     D
JP      NZ,A2078
LD      A,(A16155)
OR      A
JP      M,A2206
LD      HL,(A16152)
ADD     HL,HL
LD      (A16162),HL
LD      HL,(A16154)
ADC     HL,HL
LD      (A16164),HL
A2192:
LD      A,(A16151)
OR      A
RET     M
LD      A,(A16165)
AND     127
LD      (A16165),A
RET
;*****
A2206:
LD      A,(A16166)
INC     A
LD      (A16166),A
LD      HL,(A16152)
LD      (A16162),HL
LD      HL,(A16154)
LD      (A16164),HL
JP      A2192
;*****
A2228:
LD      HL,A16167
LD      BC,A16158
LD      A,(BC)
ADD     A,(HL)

```



```

;*****
;ABS(n) function.
;
A2276:
    JP    NZ,A7939
    LD    HL,A16165
    RES   7,(HL)
    RET
;*****
;SGN(n) function.
;
A2285:
    JP    NZ,A7939
    LD    A,(A16166)
    OR    A
    RET   Z
    LD    A,(A16165)
    OR    A
    LD    HL,0
    LD    (A16162),HL
    LD    (A16164),HL
    LD    A,129
    LD    (A16166),A
    RET   P
A2312:
    LD    HL,A16174
    JP    C,A2321
A2318:
    LD    HL,A16165
A2321:
    LD    A,128
    XOR   (HL)
    LD    (HL),A
    RET
;*****
;Toggle FPA1 (CF=0) or FPA2 (CF=1).
;
A2326:

```

```

        LD     HL,A16175
        JP     C,A2335
        LD     HL,A16166
A2335:
        LD     A,(HL)
        OR     A
        JP     NZ,A2351
        LD     (HL),129
        DEC    HL
        LD     (HL),A
        DEC    HL
        LD     (HL),A
        DEC    HL
        LD     (HL),A
        DEC    HL
        LD     (HL),A
        RET
;*****
A2351:
        XOR    A
        LD     (HL),A
        RET
;*****
;Load FPA1 to HL in integer format.
;
A2354:
        LD     A,(A16166)
        CP     129
        JP     C,A2401
        SUB    144
        JP     NC,A2405
        NEG
        EX     AF,AF'
        LD     HL,(A16164)
        BIT    7,H
        SET    7,H
        EX     AF,AF'
A2378:

```



```

        SRL     H
        RR      L
        DEC     A
        JP      NZ,A2378
        EX      AF,AF'
        JP      NZ,A2392
        OR      A
        RET
;*****
A2392:
        XOR     A
        SUB     L
        LD      L,A
        LD      A,0
        SBC     A,H
        LD      H,A
        OR      A
        RET
;*****
A2401:
        LD      HL,0
        DB      230
A2405:
        SCF
        RET
;*****
;Load HL to FPA1 (CF=0) or FPA2 (CF=1).
;
A2407:
        PUSH    DE
        EX      DE,HL
        LD      HL,A16175
        JP      C,A2418
        LD      HL,A16166
A2418:
        LD      A,D
        OR      E
        JP      Z,A2471

```

```

AND      D
JP       Z,A2474
JP       P,A2445
EX       AF,AF'
LD       A,E
NEG
LD       E,A
LD       A,D
CPL
CCF
ADC      A,0
LD       D,A
JP       Z,A2475
DB       62
A2445:
EX       AF,AF'
LD       A,144
A2448:
DEC      A
SLA     E
RL      D
JP       P,A2448
A2456:
LD       (HL),A
DEC     HL
EX      AF,AF'
JP      M,A2464
RES     7,D
A2464:
LD      (HL),D
DEC     HL
LD      (HL),E
DEC     HL
XOR     A
LD      (HL),A
DEC     HL
A2471:
LD      (HL),A

```

```

        POP     DE
        RET
;*****
A2474:
        EX     AF,AF'
A2475:
        LD     D,E
        LD     E,0
        LD     A,D
        OR     A
        LD     A,136
        JP     P,A2448
        JP     A2456
;*****
;Compare FPA1 with FPA2.  CF=1 if FPA1>FPA2.
;
A2488:
        PUSH   DE
        PUSH   HL
        LD     DE,(A16165)
        LD     HL,(A16174)
        LD     A,D
        OR     A
        JP     Z,A2559
        LD     A,H
        OR     A
        JP     Z,A2570
        LD     A,L
        XOR    E
        JP     M,A2570
        LD     A,D
        CP     H
        JP     NZ,A2547
        LD     A,E
        CP     L
        JP     NZ,A2547
        LD     DE,A16164
        LD     HL,A16173

```

```
LD    A, (DE)
CP    (HL)
JP    NZ, A2547
DEC   DE
DEC   HL
LD    A, (DE)
CP    (HL)
JP    NZ, A2547
DEC   DE
DEC   HL
LD    A, (DE)
CP    (HL)
JP    Z, A2573
```

A2547:

```
POP   HL
POP   DE
LD    A, (A16165)
RLA
ADC   A, 0
RRA
RET   NZ
INC   A
RET
```

;*****

A2559:

```
OR    H
JP    Z, A2573
LD    A, L
OR    A
JP    M, A2573
SET   7, E
```

A2570:

```
SCF
RL    E
```

A2573:

```
POP   HL
POP   DE
RET
```

```
;*****  
;Convert number from ASCII to FPA1.
```

```
;
```

```
A2576:
```

```
    EXX  
    PUSH    BC  
    PUSH    DE  
    PUSH    HL  
    XOR     A  
    LD      (A16151),A  
    LD      B,A  
    LD      C,A  
    LD      D,A  
    LD      E,A  
    EXX  
    LD      L,A  
    LD      C,A  
    LD      B,1  
    EX      AF,AF'  
    LD      A,(DE)  
    CP      43  
    JR      Z,A2607  
    SUB     45  
    JR      NZ,A2608  
    DEC     A  
    LD      (A16151),A
```

```
A2607:
```

```
    INC     DE
```

```
A2608:
```

```
    LD      A,(DE)  
    CP      69  
    JR      Z,A2719  
    CP      101  
    JR      Z,A2719  
    CP      46
```

```
A2619:
```

```
    JR      NZ,A2628  
    DEC     C
```

```
DEC B
JR Z,A2607
JP A2770
```

```
;*****
```

```
A2628:
```

```
SUB 48
CP 10
JP NC,A2770
EX AF,AF'
ADD A,C
EX AF,AF'
LD H,A
LD A,L
AND 240
JR NZ,A2607
EXX
CALL A2708
PUSH DE
PUSH BC
EXX
LD A,L
ADC A,A
LD L,A
EXX
CALL A2708
EXX
ADC A,A
EXX
CALL A2708
EXX
ADC A,A
EXX
POP HL
ADD HL,BC
LD B,H
```

```
A2670:
```

```
LD C,L
POP HL
```

```
ADC    HL,DE
EX     DE,HL
EXX
ADC    A,L
LD     L,A
LD     A,H
EXX
ADD    A,C
LD     C,A
JR     NC,A2695
XOR    A
INC    B
JR     NZ,A2695
INC    E
JR     NZ,A2695
INC    D
JR     NZ,A2695
SCF
```

A2695:

```
EXX
LD     A,L
ADC    A,0
LD     L,A
AND    240
JR     Z,A2607
INC    C
JP     A2607
```

;*****

A2708:

```
LD     H,B
LD     L,C
ADD    HL,HL
LD     B,H
LD     C,L
LD     H,D
LD     L,E
ADC    HL,HL
EX     DE,HL
```

```

        RET
;*****
A2719:
        XOR    A
        LD     B,A
        LD     C,A
        INC   DE
        LD     A,(DE)
        CP    43
        JR    Z,A2733
        CP    45
        JR    NZ,A2735
        DEC   C
A2733:
        INC   DE
        LD     A,(DE)
A2735:
        SUB   48
        CP    10
        JR    NC,A2757
        LD     H,A
        LD     A,B
        CP    12
        JR    NC,A2833
        RLCA
        LD     B,A
        RLCA
        RLCA
        ADD   A,B
        ADD   A,H
        LD     B,A
        JP    A2733
;*****
A2757:
        INC   C
        JR    NZ,A2764
        LD     A,B
        NEG

```



```

        LD      B,A
A2764:  EX      AF,AF'
        ADD    A,B
        JP    PE,A2842
        EX    AF,AF'
A2770:  LD      H,L
        EXX
        PUSH  BC
        PUSH  DE
        EXX
        POP   BC
        LD    L,B
        LD    B,C
        LD    (A16162),DE
        POP  DE
        LD    C,D
        LD    D,168
        LD    (A16164),HL
        CALL A2826
        LD    HL,(A16162)
        PUSH HL
        LD    HL,(A16164)
        CALL A1784
        EX    AF,AF'
        ADD  A,38
        CP   77
        JR   NC,A2850
A2809:  LD      B,0
        LD    C,A
        LD    HL,A2875
        ADD  HL,BC
        ADD  HL,BC
        ADD  HL,BC
        ADD  HL,BC
        ADD  HL,BC

```

```

        CALL    A1885
        POP     DE
        OR      A
        RET
;*****
A2826:
        POP     HL
        EXX
        POP     HL
        POP     DE
        POP     BC
        EXX
        JP      (HL)
;*****
A2833:
        CALL    A2826
        INC     C
        JP      Z,A1802
A2840:
        SCF
        RET
;*****
A2842:
        CALL    A2826
        JP      P,A1802
        JR      A2840
;*****
A2850:
        CP      245
        JR      C,A2867
        ADD     A,11
        PUSH    AF
        LD      HL,A3010
        CALL    A1885
        POP     AF
        JP      A2809
;*****
A2867:

```

```
POP    DE
SUB    38
JP     M,A1802
JR     A2840
```

```
;*****
```

```
;Powers of 10 in floating point format.
```

```
;
```

```
A2875:
```

```
DB     237,220,199,089,002;1E-38
```

```
A2879 EQU    A2875+4
```

```
DB     020,234,028,008,006      ;1E-37
DB     153,036,036,042,009      ;1E-36
DB     192,045,173,084,012      ;1E-35
DB     152,060,236,004,016      ;1E-34
DB     190,075,039,038,019      ;1E-33
DB     173,030,177,079,022      ;1E-32
DB     044,179,206,001,026      ;1E-31
DB     247,095,066,034,029      ;1E-30
DB     245,247,210,074,032      ;1E-29
DB     243,181,135,125,035      ;1E-28
DB     184,209,116,030,039      ;1E-27
DB     037,006,018,070,042      ;1E-26
DB     175,135,150,119,045      ;1E-25
DB     205,020,190,026,049      ;1E-24
DB     001,154,109,065,052      ;1E-23
DB     129,000,201,113,055      ;1E-22
DB     080,160,029,023,059      ;1E-21
DB     101,008,229,060,062      ;1E-20
DB     126,074,030,108,065      ;1E-19
DB     143,238,146,019,069      ;1E-18
DB     050,170,119,056,072      ;1E-17
DB     191,148,149,102,075      ;1E-16
DB     247,124,029,016,079      ;1E-15
DB     053,220,036,052,082      ;1E-14
DB     066,019,046,097,085      ;1E-13
DB     009,204,188,012,089      ;1E-12
```

A3010:

DB	012,255,235,047,092	;1E-11
DB	207,254,230,091,095	;1E-10
DB	065,095,112,009,099	;1E-09
DB	018,119,204,043,102	;1E-08
DB	214,148,191,086,105	;1E-07
DB	006,189,055,006,109	;1E-06
DB	071,172,197,039,112	;1E-05
DB	089,023,183,081,115	;1E-04
DB	152,110,018,003,119	;1E-03
DB	061,010,215,035,122	;1E-02

A3060:

DB	205,204,204,076,125	;1E-01
----	---------------------	--------

A3065:

DB	000,000,000,000,129	;1E+00
----	---------------------	--------

A3070:

DB	000,000,000,032,132	;1E+01
DB	000,000,000,072,135	;1E+02
DB	000,000,000,122,138	;1E+03
DB	000,000,064,028,142	;1E+04
DB	000,000,080,067,145	;1E+05
DB	000,000,036,116,148	;1E+06
DB	000,128,150,024,152	;1E+07
DB	000,032,188,062,155	;1E+08
DB	000,040,107,110,158	;1E+09
DB	000,249,002,021,162	;1E+10
DB	064,183,067,058,165	;1E+11
DB	016,165,212,104,168	;1E+12
DB	042,231,132,017,172	;1E+13
DB	245,032,230,053,175	;1E+14
DB	050,169,095,099,178	;1E+15
DB	191,201,027,014,182	;1E+16
DB	047,188,162,049,185	;1E+17
DB	058,107,011,094,188	;1E+18
DB	005,035,199,010,192	;1E+19
DB	198,235,120,045,195	;1E+20
DB	183,038,215,088,198	;1E+21
DB	050,120,134,007,202	;1E+22

```

DB      063,022,104,041,205      ;1E+23
DB      207,027,194,083,208      ;1E+24
DB      097,081,089,004,212      ;1E+25
DB      186,165,111,037,215      ;1E+26
DB      040,143,203,078,218      ;1E+27
DB      121,057,063,001,222      ;1E+28
DB      215,007,143,033,225      ;1E+29
DB      205,201,242,073,228      ;1E+30
DB      064,124,111,124,231      ;1E+31
DB      168,173,197,029,235      ;1E+32
DB      018,025,055,069,238      ;1E+33
DB      087,223,132,118,241      ;1E+34
DB      150,011,019,026,245      ;1E+35
DB      124,206,151,064,248      ;1E+36
DB      027,194,189,112,251      ;1E+37
DB      081,153,118,022,255      ;1E+38

```

```
;*****
```

```
;Convert FPA1 to decimal ASCII.
```

```
;
```

```
A3260:
```

```

LD      HL,A16247
LD      A,(A16166)
OR      A
JR      NZ,A3275
LD      (HL),48
DEC     HL
LD      (HL),1
RET

```

```
;*****
```

```
A3275:
```

```

LD      A,(A16165)
OR      A
JP      P,A3290
LD      (HL),45
INC     HL
AND     127
LD      (A16165),A

```

```
A3290:
```

```

        PUSH   HL
        LD     E, 38
A3293:
        LD     A, (A16166)
        SUB    3
        INC    A
        JR     NC, A3311
        LD     HL, A3070
        CALL   A1885
        LD     E, 37
        JR     A3293
;*****
A3311:
        LD     D, A
        LD     BC, 5
        LD     HL, A2879
A3318:
        ADD    HL, BC
        DEC    E
        LD     A, (HL)
        ADD    A, D
        JR     NC, A3318
        LD     A, E
        LD     (A16156), A
        LD     DE, A16175
        LDDR
        CALL   A1888
        CALL   A3580
        LD     A, (A16178)
        LD     E, A
        SLA   A
        SLA   A
        ADD    A, E
        LD     E, A
        LD     D, 0
        LD     HL, A3070
        SBC   HL, DE
        CALL   A4375

```

```
DEC    DE
EX     DE,HL
DEC    (HL)
CALL   A1583
CALL   A3580
LD     A,(A16166)
LD     BC,(A16162)
LD     DE,(A16164)
SET    7,D
SUB    132
JR     Z,A3396
```

A3385:

```
SRL    D
RR     E
RR     B
RR     C
INC    A
JR     NZ,A3385
```

A3396:

```
LD     A,(A16178)
```

A3399:

```
POP    HL
PUSH   AF
LD     A,D
RRA
RRA
RRA
RRA
AND    15
ADD    A,48
LD     (HL),A
INC    HL
LD     A,D
AND    15
LD     D,A
POP    AF
DEC    A
JR     Z,A3451
```

```
PUSH HL
LD H,B
LD L,C
ADD HL,HL
LD B,H
LD C,L
EX DE,HL
ADC HL,HL
PUSH HL
EX DE,HL
ADD HL,HL
EX DE,HL
ADC HL,HL
EX DE,HL
ADD HL,HL
EX DE,HL
ADC HL,HL
EX DE,HL
ADD HL,BC
LD B,H
LD C,L
POP HL
ADC HL,DE
EX DE,HL
JP A3399
```

```
;*****
```

```
A3451:
```

```
PUSH AF
LD A,(A16178)
LD B,A
```

```
A3456:
```

```
DEC HL
LD A,(HL)
CP 48
JR NZ,A3466
DJNZ A3456
```

```
A3464:
```

```
DEC HL
```



```

        LD      A, (HL)
A3466:  PUSH     AF
        DJNZ   A3464
        LD      A, (A16156)
        LD      C, 0
        ADD    A, 2
        CP     11
        JR     C, A3483
        INC    C
        LD      A, 2
A3483:  SUB     2
        JP     P, A3508
        LD      (HL), 46
        NEG
        LD      B, A
        JR     A3498
;*****
A3495:  INC     HL
        LD      (HL), 48
A3498:  DJNZ   A3495
A3500:  POP     AF
A3501:  INC     HL
        LD      (HL), A
        JR     NZ, A3500
        JP     A3530
;*****
A3508:  LD      B, A
        INC    B
A3510:  POP     AF
        JR     Z, A3525

```

```

LD      (HL),A
INC     HL
DJNZ   A3510
LD      (HL),46
POP     AF
JR      NZ,A3501
JP      A3530
;*****
A3525:
LD      (HL),48
INC     HL
DJNZ   A3525
A3530:
DEC     C
JR      NZ,A3570
LD      (HL),69
INC     HL
LD      A,(A16156)
OR      A
LD      B,43
JP      P,A3549
LD      B,45
NEG
A3549:
LD      (HL),B
INC     HL
LD      B,47
A3553:
INC     B
SUB     10
JP      P,A3553
LD      (HL),B
INC     HL
LD      B,58
A3563:
DEC     B
INC     A
JP      M,A3563

```

```

        LD      (HL),B
        INC    HL
A3570:  DEC     HL
        LD     DE,A16246
        OR     A
        SBC   HL,DE
        LD     A,L
        LD     (DE),A
        RET
;*****
;Scale FPA1.  If FPA1>110, then multiply it by 0.1.
;
A3580:  LD     A,(A16166)
        CP     132
        RET    C
        JR    NZ,A3594
        LD     A,(A16165)
        CP     32
        RET    C
A3594:  LD     HL,A16156
        INC   (HL)
        LD     HL,A3060
        JP    A1885
;*****
;LOG(n) function.
;
A3604:  JP     NZ,A7939
A3607:  LD     HL,(A16165)
        LD     A,H
        OR     A
        JP    Z,A7936
        BIT   7,L
        JP    NZ,A7936

```

```
PUSH AF
LD A,128
LD (A16166),A
LD HL,A4543
CALL A1569
CALL A4396
LD HL,A4538
CALL A1569
CALL A4441
CALL A2018
LD HL,A4675
CALL A4156
LD HL,A4533
CALL A1569
POP AF
SUB 128
LD L,A
SBC A,A
LD H,A
SCF
CALL A2407
CALL A1583
LD HL,A4498
JP A1885
```

```
;*****
```

```
;SQR(n) function.
```

```
;
```

```
A3678:
```

```
JP NZ,A7939
LD A,(A16166)
OR A
RET Z
CALL A3607
LD HL,A16166
DEC (HL)
JP A3819
```

```
;*****
```

```
;Raise to power (^).
```

```
;
A3696:
    LD     A, (A16175)
    OR     A
    JP     Z, A4384
    LD     A, (A16166)
    OR     A
    RET    Z
    LD     DE, A3804
    CALL  A4482
    LD     HL, A16174
    BIT   7, (HL)
    JR     Z, A3727
    RES   7, (HL)
    LD     HL, A3900
    PUSH  HL
```

```
A3727:
    CALL  A4396
    CALL  A1832
    CALL  A10675
    AND   1
    LD    (A16194), A
    CALL  A2488
    JR    Z, A3754
    LD    A, 255
    LD    (A16194), A
    JP    A3786
```

```
;*****
```

```
A3754:
    CALL  A4456
    JR    C, A3786
    CP    26
    JR    NC, A3786
    CALL  A4411
    CALL  A4356
    PUSH  AF
    JP    A3782
```

```
;*****
```

```

A3773:
    PUSH    BC
    LD      HL,A16174
    RES     7,(HL)
    CALL    A1888
A3782:
    POP     BC
    DJNZ    A3773
    RET
;*****
A3786:
    CALL    A4411
    CALL    A4426
    CALL    A3607
    CALL    A4441
    CALL    A1888
    JP      A3819
;*****
A3804:
    LD      A,(A16194)
    SRA    A
    JP      M,A7936
    RET    NC
    JP      A2318
;*****
;EXP(n) function.
;
A3816:
    JP      NZ,A7939
A3819:
    LD      HL,A4503
    CALL    A1885
    CALL    A4356
    CALL    A4456
    JR      C,A3890
    OR     A
    JP      P,A3848
    NEG

```

```
LD HL,A3900
PUSH HL
LD HL,A16174
RES 7,(HL)
```

A3848:

```
PUSH AF
NEG
LD L,A
SBC A,A
LD H,A
XOR A
CALL A2407
CALL A1583
LD HL,A4587
CALL A4278
LD A,(A16166)
POP BC
ADD A,B
JP NC,A1803
POP HL
LD DE,A3900
PUSH HL
OR A
SBC HL,DE
POP HL
JP NZ,A7921
JP A1802
```

;*****

A3890:

```
LD A,(A16174)
OR A
JP M,A1802
JP A7936
```

;*****

A3900:

```
CALL A4356
LD HL,A3065
CALL A4387
```

```

        JP      A2018
;*****
;TAN(n) function.
;
A3912:
        JP      NZ,A7939
        LD      A,(A16165)
        RLA
        LD      A,128
        RRA
        CALL    A3964
        CALL    A4396
        LD      BC,(A16179)
        LD      DE,(A16181)
        XOR     A
        CALL    A4051
        CALL    A4441
        JP      A2018
;*****
;COS(n) function.
;
A3946:
        JP      NZ,A7939
        LD      A,64
        JP      A3964
;*****
;SIN(n) function.
;
A3954:
        JP      NZ,A7939
        LD      A,(A16165)
        RLA
        LD      A,0
        RRA
A3964:
        LD      HL,A16165
        RES    7,(HL)
        LD      (A16194),A

```



```

LD HL,A4513
CALL A1885
LD BC,(A16162)
LD DE,(A16164)
SET 7,D
LD A,(A16166)
SUB 130
JR C,A4010
JR Z,A4040
CP 32
JR C,A4029
A4001:
LD BC,0
LD DE,0
JP A4040
;*****
A4010:
CP 225
JR C,A4001
A4014:
SRL D
RR E
RR B
RR C
INC A
JP NZ,A4014
JP A4040
;*****
A4029:
SLA C
RL B
RL E
RL D
DEC A
JR NZ,A4029
A4040:
LD (A16179),BC
LD (A16181),DE

```

```
LD      A, (A16194)
A4051:  ADD     A, D
        LD     D, A
        JP     P, A4062
        LD     HL, A2318
        PUSH  HL
        RES   7, D
```

```
A4062:  BIT     6, D
        JR     Z, A4080
        LD     HL, 0
        XOR   A
        SBC   HL, BC
        LD     B, H
        LD     C, L
        LD     HL, 32768
        SBC   HL, DE
        EX    DE, HL
```

```
A4080:  LD     HL, (A4550)
        OR    A
        SBC   HL, DE
        JR    C, A4102
        JR    NZ, A4095
        LD     HL, (A4548)
        SBC   HL, BC
```

```
A4095:  LD     HL, A4628
        PUSH  AF
        JR    NC, A4120
        DB    38 ;LD H,245 for awful
overlapping code
```

```
;avoids the PUSH AF
```

```
A4102:  PUSH  AF
        LD     HL, 0
        OR    A
```

```

        SBC    HL,BC
        LD     B,H
        LD     C,L
        LD     HL,16384      ;not an address; not sure
what it is
        SBC    HL,DE
        EX     DE,HL
        LD     HL,A4649
A4120:
        LD     A,B
        OR     C
        OR     D
        OR     E
        JR     Z,A4142
        LD     A,130
A4128:
        DEC    A
        SLA   C
        RL    B
        RL    E
        RL    D
        JP    P,A4128
        RES   7,D
A4142:
        LD    (A16166),A
        LD    (A16162),BC
        LD    (A16164),DE
        POP   AF
        JR    C,A4270
;*****
;Power series calculator #1.
;
A4156:
        LD    (A16183),HL
        CALL  A4356
        CALL  A4396
        CALL  A1888
        LD    HL,(A16183)

```

```
CALL A4278
CALL A4441
JP A1888
```

```
;*****
```

```
;ATN(n) function.
```

```
;
```

```
A4180:
```

```
JP NZ,A7939
CALL A4479
LD HL,A4523
CALL A4375
CALL A2488
JR C,A4249
LD HL,A4528
CALL A4375
CALL A2488
JR C,A4218
CALL A3900
LD HL,A4255
PUSH HL
JP A4249
```

```
;*****
```

```
A4218:
```

```
LD HL,A3065
CALL A1569
CALL A3900
LD HL,(A16165)
SET 7,L
INC H
LD (A16165),HL
LD HL,A3065
CALL A1569
LD HL,A4264
PUSH HL
CALL A4479
```

```
A4249:
```

```
LD HL,A4556
JP A4156
```

```

;*****
A4255:
        CALL    A2318
        LD      HL,A4508
        JP      A1569
;*****
A4264:
        LD      HL,A4518
        JP      A1569
;*****
;Power series calculator #2.
;
A4270:
        PUSH    HL
        CALL    A4356
        CALL    A1888
        POP     HL
;*****
;Power series calculator #3.
;
A4278:
        LD      A,(HL)
        PUSH    AF
        INC     HL
        LD      BC,5
        LD      DE,A16171
        LDIR
        LD      (A16183),HL
        LD      HL,(A16162)
        LD      (A16185),HL
        LD      HL,(A16164)
        LD      (A16187),HL
        LD      A,(A16166)
        LD      (A16189),A
        JP      A4332
;*****
A4313:
        PUSH    BC

```

```
LD HL,(A16185)
LD (A16171),HL
LD HL,(A16187)
LD (A16173),HL
LD A,(A16189)
LD (A16175),A
```

A4332:

```
CALL A1888
LD HL,(A16183)
LD BC,5
LD DE,A16171
LDIR
LD (A16183),HL
CALL A1583
POP BC
DJNZ A4313
RET
```

;*****

;Load FPA1 to FPA2.

;

A4356:

```
LD HL,(A16162)
LD (A16171),HL
LD HL,(A16164)
LD (A16173),HL
LD HL,(A16165)
LD (A16174),HL
RET
```

;*****

;Load (HL) to FPA2.

;

A4375:

```
LD DE,A16171
LD BC,5
LDIR
RET
```

;*****

;Load FPA1 with 1.

```

;
A4384:
    LD    HL,A3065
A4387:
    LD    DE,A16162
    LD    BC,5
    LDIR
    RET
;*****
;Push FPA1 to stack.
;
A4396:
    POP   DE
    LD    HL,(A16165)
    PUSH HL
    LD    HL,(A16163)
    PUSH HL
    LD    HL,(A16161)
    PUSH HL
    EX   DE,HL
    JP   (HL)
;*****
;Pop FPA1 from stack.
;
A4411:
    POP   DE
    POP   HL
    LD    (A16161),HL
    POP   HL
    LD    (A16163),HL
    POP   HL
    LD    (A16165),HL
    EX   DE,HL
    JP   (HL)
;*****
;Push FPA2 to stack.
;
A4426:

```

```
POP    DE
LD     HL,(A16174)
PUSH   HL
LD     HL,(A16172)
PUSH   HL
LD     HL,(A16170)
PUSH   HL
EX     DE,HL
JP     (HL)
```

```
;*****
```

```
;Pop FPA2 from stack.
```

```
;
```

```
A4441:
```

```
POP    DE
POP    HL
LD     (A16170),HL
POP    HL
LD     (A16172),HL
POP    HL
LD     (A16174),HL
EX     DE,HL
JP     (HL)
```

```
;*****
```

```
;Determine if  $-127 < \text{FPA1} < 127$ . CF=0 if yes, CF=1 if  
no.
```

```
;
```

```
A4456:
```

```
CALL  A2354
RET   C
LD    A,H
OR    A
JP    P,A4472
CP    255
RET   C
LD    A,L
CP    129
RET
```

```
;*****
```



```

A4472:
    JR    NZ,A4477
    LD    A,L
    CP    128
A4477:
    CCF
    RET
;*****
;Temporary ABS.  Used internally.
;
A4479:
    LD    DE,A2318
A4482:
    LD    HL,A16165
    BIT   7,(HL)
    RET   Z
    LD    A,(A16166)
    OR    A
    RET   Z
    RES   7,(HL)
    EX    DE,HL
    EX    (SP),HL
    JP    (HL)
;*****
;Floating point constants.
;
A4498:
    DB    248,023,114,049,128        ;ln 2
A4503:
    DB    041,059,170,056,129        ;1/(ln 2)
A4508:
    DB    162,218,015,073,129        ;pi/2
A4513:
    DB    110,131,249,034,128        ;2/pi
A4518:
    DB    162,218,015,073,128        ;pi/4
A4523:
    DB    208,204,019,084,127        ;SQR(2)-1

```

```

A4528:
    DB      154,121,130,026,130      ;SQR(2)+1
A4533:
    DB      000,000,000,128,128      ;-1/2
A4538:
    DB      052,243,004,181,129      ;-SQR(2)
A4543:
    DB      052,243,004,053,128      ;1/SQR(2)
;*****
;Data table.
;
A4548:
    DB      026,180                    ;
A4550:
    DB      236,024                    ;
A4552:
    DB      045,230,064,187           ;used by RND
;*****
;ATN(n) power series coefficients.
;
A4556:
    DB      005
    DB      014,075,046,247,124
    DB      079,071,139,088,125
    DB      102,130,209,145,126
    DB      006,032,200,076,126
    DB      010,162,170,170,127
    DB      253,255,255,127,128
;*****
;EXP(n) power series coefficients.
;
A4587:
    DB      007
    DB      013,108,093,045,113
    DB      124,111,101,025,116
    DB      246,248,118,047,119
    DB      118,143,141,029,122
    DB      178,152,088,099,124

```

```

        DB      188,238,253,117,126
        DB      249,023,114,049,128
        DB      000,000,000,000,129
;*****
;SIN(n) (routine #1) power series coefficients.
;
A4628:
        DB      003
        DB      132,136,209,151,121
        DB      074,118,051,035,125
        DB      210,228,093,165,128
        DB      161,218,015,073,129
;*****
;SIN(n) (routine #2) power series coefficients.
;
A4649:
        DB      004
        DB      255,251,185,106,118
        DB      054,018,217,170,123
        DB      074,218,224,001,127
        DB      215,229,233,157,129
        DB      254,255,255,127,128
;*****
;LOG(n) power series coefficients.
;
A4675:
        DB      003
        DB      121,203,086,094,127
        DB      100,011,155,019,128
        DB      021,147,056,118,128
        DB      029,059,170,056,130
;*****
;RND(n) function.
;
A4696:
        JP      NZ,A7939
        LD      DE,(A16165)
        LD      BC,(A16190)

```

```
LD HL,(A16192)
LD A,D
OR A
JR Z,A4770
BIT 7,E
JR Z,A4723
EX DE,HL
LD BC,(A16163)
```

A4723:

```
LD (A16171),BC
LD (A16173),HL
LD BC,1
LD DE,0
LD HL,A4552
CALL A4780
LD (A16190),A
CALL A4780
LD (A16191),A
CALL A4780
LD (A16192),A
CALL A4780
LD (A16193),A
LD BC,(A16190)
LD HL,(A16192)
```

A4770:

```
XOR A
LD (A16151),A
LD D,128
LD E,A
JP A1784
```

;*****

A4780:

```
LD A,(HL)
PUSH HL
CALL A1968
POP HL
EX AF,AF'
INC HL
```

```

        RET
;*****
;Push FPA1 to stack with string check.
;
A4789:
        POP     HL
        LD      A,L
        EX     AF,AF'
        LD      A,H
        LD      HL,(A16165)
        PUSH   HL
        LD      HL,(A16163)
        PUSH   HL
        LD      HL,(A16161)
        PUSH   HL
        INC     L
        DEC     L
        JR     NZ,A4813
        LD      H,A
        EX     AF,AF'
        LD      L,A
        JP     (HL)
;*****
A4813:
        PUSH   AF
        LD      HL,(A16162)
        PUSH   HL
        LD      HL,4
        ADD    HL,SP
        LD      A,L
        EX     (SP),HL
        LD      (HL),A
        INC    HL
        EX     (SP),HL
        LD      A,H
        EX     (SP),HL
        LD      (HL),A
        POP    HL

```

```

        POP    HL
        EX     AF,AF'
        LD     L,A
        JP     (HL)
;*****
;Pop FPA1 from stack with string check.
;
A4835:
        POP    HL
        LD     A,L
        EX     AF,AF'
        LD     A,H
        POP    HL
        INC    L
        DEC    L
        LD     (A16161),HL
        POP    HL
        LD     (A16163),HL
        POP    HL
        LD     (A16165),HL
        JR     NZ,A4859
A4855:
        LD     H,A
        EX     AF,AF'
        LD     L,A
        JP     (HL)
;*****
A4859:
        LD     HL,(A16162)
        LD     (HL),34
        INC    HL
        LD     (HL),63
        JR     A4855
;*****
;Load FPA1 to FPA2 with string check.
;
A4869:
        LD     HL,(A16165)

```

```
LD      (A16174),HL
LD      HL,(A16163)
LD      (A16172),HL
LD      HL,(A16161)
LD      (A16170),HL
INC     L
DEC     L
RET     Z
LD      HL,(A16171)
LD      (HL),43
INC     HL
LD      (HL),63
RET
```

```
;*****
```

```
;Load FPA2 to FPA1 with string check.
```

```
;
```

```
A4899:
```

```
LD      HL,(A16174)
LD      (A16165),HL
LD      HL,(A16172)
LD      (A16163),HL
LD      HL,(A16170)
LD      (A16161),HL
INC     L
DEC     L
RET     Z
LD      HL,(A16162)
LD      (HL),34
INC     HL
LD      (HL),63
RET
```

```
;*****
```

```
;Get number from crunch code.  On entry, A=number  
type.
```

```
;
```

```
A4929:
```

```
CP      147
JP      NC,A4958
```



```
DB 002 ;
DB 004 ;
DB 006 ;
DB 006 ;
DB 006 ;
DB 006 ;
DB 008 ;
DB 010 ;
DB 012 ;
```

```
;*****
```

```
;Get integer 0-9.
```

```
;
```

```
A4982:
```

```
DB 230 ;AND 55 ;clears carry, FPA1
;I hate overlapping code!
```

```
A4983:
```

```
SCF ;sets carry, FPA2
LD H,0
LD A,(DE)
RLA
AND 31
RRA
LD L,A
JP A2407
```

```
;*****
```

```
;Get integer 10-255.
```

```
;
```

```
A4995:
```

```
DB 230 ;AND 55 ;clears carry, FPA1
;I hate overlapping code!
```

```
A4996:
```

```
SCF ;sets carry, FPA2
EXX
DEC C
EXX
INC DE
LD A,(DE)
LD H,0
```

```

        LD     L,A
        JP     A2407
;*****
;Get integer 256-65535.
;
A5008:
        DB     230           ;AND 55 ;clears carry, FPA1
                               ;I hate overlapping code!
A5009:
        SCF           ;sets carry, FPA2
        EXX
        DEC     C
        DEC     C
        EXX
        INC     DE
        LD     A,(DE)
        LD     L,A
        INC     DE
        LD     A,(DE)
        LD     H,A
        JP     A2407
;*****
;Load variable to FPA1.
;
A5023:
        PUSH   BC
        LD     HL,A16162
        PUSH   HL
        CALL  A6559
;*****
;FPA1 variable routine vector table.
;
        DW     A5059 ;floating point
        DW     A5078 ;%
        DW     A5095 ;$
        DW     A5135 ;FN
        DW     A5135 ;command
;*****

```

```

;Load variable to FPA2.
;
A5041:
    PUSH    BC
    LD      HL,A16171
    PUSH    HL
    CALL    A6560
;*****
;FPA2 variable routine vector table.
;
    DW      A5059 ;floating point
    DW      A5078 ;%
    DW      A5095 ;$
    DW      A5141 ;FN
    DW      A5141 ;command
;*****
;Load FP variable from (BC) to (HL).
;
A5059:
    EX      DE,HL
    EX      (SP),HL
    LD      D,B
    LD      E,C
    EX      DE,HL
    LDI
    LDI
    LDI
    LDI
    LDI
    POP     DE
    POP     BC
    XOR     A
    RET
;*****
;Load % variable from (BC) to (HL).
;
A5078:
    LD      A,(BC)

```

```
EX    AF,AF'
INC   BC
LD    A,(BC)
POP   HL
LD    BC,4
ADD   HL,BC
POP   BC
PUSH  DE
LD    D,A
EX    AF,AF'
LD    E,A
JP    A2418
```

```
;*****
```

```
;Load $ variable from (BC) to (HL).
```

```
;
```

```
A5095:
```

```
LD    A,(BC)
LD    L,A
INC   BC
LD    A,(BC)
LD    H,A
INC   HL
INC   HL
LD    A,(HL)
EX    DE,HL
EX    (SP),HL
PUSH  HL
CALL  A7264
POP   BC
LD    A,L
LD    (BC),A
INC   BC
LD    A,H
LD    (BC),A
INC   HL
INC   HL
LD    A,(HL)
OR    A
```

```

        LD      C,A
        LD      B,0
        INC     DE
        INC     HL
        EX      DE,HL
        JR      Z,A5129
        LDIR
A5129:
        POP     DE
        POP     BC
        LD      A,255
        SCF
        RET
;*****
;Execute variable command or FN.
;
A5135:
        POP     HL
        CALL    A5167
        POP     BC
        RET
;*****
;Variable command or FN interpreter.
;
A5141:
        POP     HL
        LD      (A16141),A
        CALL    A4789
        LD      A,(A16141)
        CALL    A5167
        LD      (A16161),A
        LD      B,A
        CALL    A4869
        CALL    A4835
        LD      A,B
        POP     BC
        RET
;*****

```

;Evaluate variable command or FN.

;

A5167:

INC B

DEC B

JP Z,A7933

JP A29477 ;patch to deal with

multi-argument functions

;*****

NOP

;*****

;Variable commands.

;

A5176:

PUSH BC

EXX

DEC C

EXX

INC DE

AND 32

JR NZ,A5207

;*****

;Numeric commands.

;

A5185:

CALL A5939

OR A

POP HL

PUSH DE

CALL A5938

A5194:

POP DE

EXX

DEC C

EXX

INC DE

LD A,(DE)

CP 184

```

        JP      NZ ,A7948
        XOR     A
        RET
;*****
;String commands.
;
A5207:
        CALL   A5939
        OR     A
        POP    HL
        CALL   A5938
        LD     A ,255
        RET
;*****
;FN(n) function.
;
A5218:
        CP     200
        JP     NZ ,A7948
        EXX
        DEC    C
        EXX
        INC    DE
        PUSH   BC
        CALL   A5939
        POP    HL
        OR     A
        JP     NZ ,A7939
        EXX
        DEC    C
        EXX
        INC    DE
        LD     A , (DE)
        CP     184
        JP     NZ ,A7948
        PUSH   DE
        LD     A , (HL)
        LD     BC ,4

```

```
ADD    HL,BC
LD     C,(HL)
INC    C
SUB    C
SUB    6
EXX
PUSH   BC
LD     C,A
EXX
INC    HL
ADD    HL,BC
EX     DE,HL
CALL   A6559
```

```
;*****
```

```
;FN vector table.
```

```
;Only FP variables are allowed as FN arguments,
```

```
;so last 4 vectors in table are omitted.
```

```
;
```

```
        DW    A5269           ;FP
```

```
;*****
```

```
A5269:
```

```
INC     DE
INC     DE
LD      A,E
EX      AF,AF'
LD      A,D
LD      E,C
LD      D,B
LD      HL,-5
ADD     HL,SP
LD      SP,HL
LD      BC,5
EX      DE,HL
LDIR
DEC     HL
EX      DE,HL
LD      HL,A16166
LD      BC,5
```



```
LDDR
INC    DE
PUSH  DE
LD     D,A
EX     AF,AF'
LD     E,A
CALL  A5939
POP    DE
LD     BC,5
LD     HL,0
ADD   HL,SP
LDIR
LD     SP,HL
EXX
POP    BC
EXX
POP    DE
XOR   A
RET
```

```
;*****
```

```
;Move string from crunch code to string space
address in FPA1.
```

```
;
```

```
A5322:
```

```
LD     HL,A16162
JP     A5331
```

```
;*****
```

```
A5328:
```

```
LD     HL,A16171
```

```
A5331:
```

```
EXX
DEC    C
LD     A,C
EXX
EX     DE,HL
INC   HL
PUSH  BC
LD     B,0
```

```

LD      C, (HL)
SUB     C
EXX
LD      C, A
EXX
LD      A, C
PUSH   HL
LD      H, D
LD      L, E
CALL   A7264
EX      DE, HL
LD      (HL), E
INC     HL
LD      (HL), D
POP     HL
LD      A, B
OR      C
JP      Z, A5369
INC     HL
INC     DE
INC     DE
INC     DE
LDIR
DEC     HL
A5369:
EX      DE, HL
POP     BC
LD      A, 255
RET
;*****
;Load FP number from (DE) to FPA1.
;
A5374:
LD      HL, A16162
JP      A5383
;*****
A5380:
LD      HL, A16171

```

A5383:

```
    EXX
    LD    A,C
    SUB   5
    LD    C,A
    EXX
    INC   DE
    PUSH  BC
    EX    DE,HL
    LDI
    LDI
    LDI
    LDI
    LDI
    EX    DE,HL
    DEC   DE
    POP   BC
    XOR   A
    RET
```

;*****

;Move string from (HL) to (DE).

;

A5407:

```
    LD    B,0
    LD    C,(HL)
    INC   HL
    LD    A,B
    OR    C
    RET   Z
    LDIR
    RET
```

;*****

;Addition. +

;

A5417:

```
    JR    NZ,A5426
    CALL  A1583
    POP   DE
```

```

        POP     BC
        XOR     A
        RET
;*****
A5426:
        LD     HL,(A16171)
        LD     DE,(A16162)
        INC    HL
        INC    HL
        INC    DE
        INC    DE
        LD     A,(DE)
        ADD    A,(HL)
        JP    C,A7945
        PUSH   HL
        LD     HL,A16162
        CALL  A7264
        LD     (A16162),HL
        INC    HL
        INC    HL
        INC    HL
        EX    DE,HL
        CALL  A5407
        POP    HL
        CALL  A5407
        POP    DE
        POP    BC
        LD     A,255
        RET
;*****
;Subtraction.  -
;
A5468:
        CALL  A1575
        POP    DE
        POP    BC
        XOR    A
        RET

```

```
;*****  
;Multiplication.  *
```

```
;
```

```
A5475:
```

```
    CALL  A1888  
    POP   DE  
    POP   BC  
    XOR   A  
    RET
```

```
;*****
```

```
;Division.  \
```

```
;
```

```
A5482:
```

```
    CALL  A2018  
    POP   DE  
    POP   BC  
    XOR   A  
    RET
```

```
;*****
```

```
;Exponentiation.  ^
```

```
;
```

```
A5489:
```

```
    CALL  A3696  
    POP   DE  
    POP   BC
```

```
A5494:
```

```
    XOR   A  
    RET
```

```
;*****
```

```
;Logical AND.
```

```
;
```

```
A5496:
```

```
    LD    A, (A16166)  
    OR    A  
    JP    Z, A5624  
    LD    A, (A16175)  
    OR    A  
    JP    Z, A5624
```

```

        JP      A5594
;*****
;Logical OR.
;
A5513:
        LD      A,(A16166)
        LD      B,A
        LD      A,(A16175)
        OR      B
        JP      Z,A5624
        JP      A5594
;*****
;Less than.  <
;
A5527:
        JR      NZ,A5536
        CALL   A2488
A5532:
        JR      C,A5594
        JR      A5624
;*****
A5536:
        CALL   A5636
        JR      A5532
;*****
;Greater than.  >
;
A5541:
        JR      NZ,A5552
        CALL   A2488
A5546:
        JR      C,A5624
        JR      Z,A5624
        JR      A5594
;*****
A5552:
        CALL   A5636
        JR      A5546

```

```

;*****
;Less than or equal to.  <=
;
A5557:
        JR      NZ,A5568
        CALL   A2488
A5562:
        JR      C,A5594
        JR      Z,A5594
        JR      A5624
;*****
A5568:
        CALL   A5636
        JR      A5562
;*****
;Greater than or equal to.  >=
;
A5573:
        JR      NZ,A5582
        CALL   A2488
A5578:
        JR      NC,A5594
        JR      A5624
;*****
A5582:
        CALL   A5636
        JR      A5578
;*****
;Not equal to.  <>
;
A5587:
        JR      NZ,A5612
        CALL   A2488
A5592:
        JR      Z,A5624
A5594:
        LD      A,129
        LD      (A16166),A

```

```

        LD      HL,0
        LD      (A16162),HL
        LD      (A16164),HL
        POP     DE
        POP     BC
        XOR     A
        RET
;*****
A5612:
        CALL   A5636
        JR     A5592
;*****
;Equal to.  =
;
A5617:
        JR     NZ,A5631
        CALL   A2488
A5622:
        JR     Z,A5594
A5624:
        XOR     A
        LD      (A16166),A
        POP     DE
        POP     BC
        RET
;*****
A5631:
        CALL   A5636
        JR     A5622
;*****
;Compare strings pointed to by FPA1 and FPA2.
;On exit, ZF=1 if equal, ZF=0 if not equal.
;
A5636:
        LD      HL,(A16162)
        EX     DE,HL
        LD      HL,(A16171)
        INC    HL

```



```

        INC     HL
        INC     DE
        INC     DE
        CALL    A11839
        EX      AF,AF'
        XOR     A
        EX      AF,AF'
        RET
;*****
;Execute (, -, NOT for FPA1.
;
A5654:
        LD      A, (DE)
        CP      183
        JP      NZ,A5669
        CALL    A5939
        EXX
        DEC     C
        EXX
        INC     DE
        RRCA
        RET
;*****
A5669:
        CP      161
        JP      NZ,A5687
        CALL    A7299
        CALL    A5837
A5680:
        DB      230                ;AND 55 ;clears CF, FPA1
                                        ;I hate overlapping code!
A5681:
        SCF                                ;sets CF, FPA2
        CALL    A2312
        XOR     A
        RET
;*****
A5687:

```

```

        CP      173
        JP      NZ,A5837
        CALL   A7299
        CALL   A5837
A5698:
        DB      230           ;AND 55 ;clears CF, FPA1
                                ;I hate overlapping code!
A5699:
        SCF                    ;sets CF, FPA2
        CALL   A2326
        XOR    A
        RET
;*****
;Execute (, -, NOT for FPA2.
;
A5705:
        LD      A,(DE)
        CP      183
        JP      NZ,A5732
        PUSH   BC
        CALL   A4789
        CALL   A5939
        LD      B,A
        CALL   A4869
        CALL   A4835
        LD      A,B
        POP    BC
        EXX
        DEC    C
        EXX
        INC    DE
        RET
;*****
A5732:
        CP      161
        JP      NZ,A5746
        CALL   A7299
        CALL   A5861

```

```

        JP      A5681
;*****
A5746:
        CP      173
        JP      NZ,A5861
        CALL    A7299
        CALL    A5861
        JP      A5699
;*****
;Check for math symbol in crunch code.
;
A5760:
        EXX
        LD      A,C
        EXX
        OR      A
        JR      Z,A5792
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A,(DE)
        SUB     160
        CP      13
        JR      NC,A5788
        LD      C,A
        ADD     A,C
        ADD     A,C
        LD      BC,A5797
        ADD     A,C
        LD      C,A
        RET     NC
        INC     B
        RET
;*****
A5788:
        EXX
        INC     C

```

```

        EXX
        DEC     DE
A5792:
        LD      BC,A5836
        XOR     A
        RET
;*****
;Math symbol vector table.
;
A5797:
        DB      5
        DW      A5417 ;      +, order=5
        DB      5
        DW      A5468 ;      -, order=5
        DB      4
        DW      A5475 ;      *, order=4
        DB      4
        DW      A5482 ;      /, order=4
        DB      0
        DW      A5489 ;      ^, order=0
        DB      6
        DW      A5527 ;      <, order=6
        DB      6
        DW      A5541 ;      >, order=6
        DB      6
        DW      A5557 ;      =<, order=6
        DB      6
        DW      A5573 ;      =>, order=6
        DB      6
        DW      A5587 ;      <>, order=6
        DB      6
        DW      A5617 ;      =, order=6
        DB      7
        DW      A5496 ;      AND, order=7
        DB      8
        DW      A5513 ;      OR, order=8
A5836:
        DB      255 ;      end of equation

```

```

;*****
;Load FPA1 from crunch code.
;
A5837:
    EXX
    DEC    C
    EXX
    INC    DE
    LD     A,(DE)
    CALL   A4929
;*****
;Vector table used by 4929 routine.  Enters with
carry clear=FPA1.
;
    DW     A4982 ;Get 0-9 integer
    DW     A4995 ;Get 10-255 integer
    DW     A5008 ;Get 256=65535 integer
    DW     A5023 ;Load variable to FPA1
    DW     A6345 ;RET
    DW     A5322 ;Move string from crunch code to
FPA1
    DW     A5374 ;Load FP number from (DE) to FPA1
    DW     A5654 ;Execute (,-,NOT for FPA1
;*****
;Load FPA2 from crunch code.
;
A5861:
    EXX
    DEC    C
    EXX
    INC    DE
    LD     A,(DE)
    CALL   A4929
;*****
;Vector table used by 4929 routine.  Enters with
carry set=FPA2.
;
    DW     A4983 ;Get 0-9 integer.

```

```

        DW      A4996 ;Get 10-255 integer
        DW      A5009 ;Get 256-65535 integer
        DW      A5041 ;Load variable to FPA2
        DW      A6345 ;RET
        DW      A5328 ;Move string from crunch code to
FPA2
        DW      A5380 ;Load FP number from (DE) to FPA2
        DW      A5705 ;Execute (,-,NOT for FPA2
;*****
;Evaluate equation.
;
A5885:
        CALL    A5861
        LD      (A16170),A
        LD      H,B
        LD      L,C
        CALL    A5760
        JP      Z,A5927
A5899:
        LD      A,(BC)
        CP      (HL)
        JP      NC,A5927
        PUSH   HL
        CALL    A4789
        CALL    A4899
        CALL    A5885
        LD      (A16161),A
        CALL    A4869
        CALL    A4835
        POP    HL
        JP      A5899
;*****
A5927:
        LD      A,(A16161)
        OR     A
        INC    HL
        LD      A,(HL)
        INC    HL

```

```

        LD      H, (HL)
        LD      L, A
        PUSH   BC
        PUSH   DE
A5938:
        JP      (HL)
;*****
;Get equation from crunch code.
;
A5939:
        EXX
        LD      A, C
        EXX
        OR      A
        RET     Z
        CALL   A7299
        CALL   A5837
        LD      (A16161), A
        EX     AF, AF'
        CALL   A5760
        JP     Z, A5972
A5960:
        CALL   A5885
        LD      (A16161), A
        EX     AF, AF'
        LD      A, (BC)
        INC    A
        JP     NZ, A5960
A5972:
        EX     AF, AF'
        SCF
        RET
;*****
;Stack setup.
;
A5975:
        JP     A28609      ;patch for WHILE/WEND
;*****

```

```

NOP
A5979:
LD     BC,A7900
PUSH  BC
LD     IX,0
ADD   IX,SP
PUSH  IX
POP   IY
JP    (HL)
;*****
;Find first line number address.
;
A5994:
CALL  A8141
EXX
LD    HL,(A16093)
LD    DE,(A16089)
ADD   HL,DE
LD    (A16101),HL
EX    DE,HL
LD    A,B
AND   248
OR    64
LD    B,A
;*****
;Find next line # address.
;
A6016:
BIT   6,B
JP    Z,A28622
LD    DE,(A16101)
LD    A,L
SUB   E
LD    A,H
SBC  A,D
JR    NC,A6050
INC  HL
INC  HL

```



```
LD     E, (HL)
INC    HL
LD     D, (HL)
INC    HL
LD     A, (DE)
LD     C, A
BIT    7, B
PUSH   DE
EXX
POP    DE
JR     NZ, A6082
RET
```

```
;*****
```

```
;END command.
```

```
;
```

```
A6047:
```

```
EXX
INC    DE
INC    DE
```

```
A6050:
```

```
BIT    6, B
JR     Z, A6063
RES    2, B
LD     (A16122), DE
LD     (A16124), HL
```

```
A6063:
```

```
EXX
INC    SP
INC    SP
JP     A16038
```

```
;*****
```

```
;TRACE routine (after TRACE is activated).
```

```
;TRACE command entry is at A6336.
```

```
;
```

```
A6069:
```

```
INC    DE
LD     A, (DE)
OR     A
```

```
PUSH DE
EXX
POP DE
JR Z,A6016
LD C,A
BIT 7,B
EXX
RET Z
```

A6082:

```
EXX
BIT 6,B
JR Z,A6110
PUSH DE
PUSH HL
DEC HL
DEC HL
DEC HL
DEC HL
LD A,35
CALL A11994
XOR A
CALL A12961
LD A,32
CALL A11994
POP HL
POP DE
LD A,(DE)
```

A6110:

```
EXX
RET
```

;*****

;Execute a command.

;

A6112:

```
PUSH DE
EXX
LD A,B
AND 156
```

```
LD      B,A
AND     16
JR      Z,A6136
LD      HL,(A16089)
CALL   A9485
LD      DE,(A16093)
ADD     HL,DE
LD      (A16101),HL
```

A6136:

```
LD      HL,65535
POP     DE
BIT     3,B
RES     3,B
EXX
CALL   NZ,A8144
EXX
BIT     2,B
EXX
CALL   NZ,A5975
DEC     DE
JP      A6216
```

;*****

;RUN or RUN line# command.

;

A6159:

```
CALL   A5975
CALL   A11907
EXX
LD      A,C
EXX
OR      A
PUSH   DE
PUSH   AF
CALL   A5994
POP    AF
JP     Z,A6190
POP    DE
EXX
```

```

        RES     6,B
        EXX
        LD      HL,A6216
        PUSH   HL
        JP      A8342
;*****
;Execution loop.
;
A6190:
        POP    HL
A6191:
        JP     A28955      ;SB1.x patch
;*****
        NOP
A6195:
        EXX
        CALL   Z,A6222
        INC   DE
        LD    A,(DE)
        LD    HL,A6423-2  ;base of command vector
table-2
        CALL   A27407      ;another SB1.x patch
        NOP
        ADD   HL,BC
        LD    A,(HL)
        INC   HL
        LD    H,(HL)
        LD    L,A
        CALL   A5938
A6216:
        CALL   A6069
        JP     A6191
;*****
A6222:
        CALL   A17310
        INC   A
        RET   NZ
        LD    HL,A17003

```

```

        LD      A, (A16135)
        XOR     (HL)
        LD      (A16136), A
        LD      A, (A16134)
        CP      (HL)
        RET     NZ
A6242:
        LD      A, 19
        JP      A7950
;*****
;LET variable=expression command.
;
A6247:
        CALL   A6555
;*****
;LET variable=expression command vector table.
;FN and variable commands give Incorrect Function
Usage errors.
;
        DW     A6260 ;FP
        DW     A6293 ;%
        DW     A6314 ;$
        DW     A7942 ;FN
        DW     A7942 ;command
;*****
;LET FP variable=expression.
;
A6260:
        PUSH   BC
        EXX
        DEC    C
        EXX
        INC    DE
        CALL   A5939
A6268:
        POP    HL
        LD     BC, (A16162)
        LD     (HL), C

```

```

        INC     HL
        LD      (HL),B
        INC     HL
        LD      BC,(A16164)
        LD      (HL),C
        INC     HL
        LD      (HL),B
        INC     HL
        LD      A,(A16166)
        LD      (HL),A
        RET
;*****
A6290:
        PUSH   BC
        JR     A6301
;*****
;LET % variable=expression.
;
A6293:
        PUSH   BC
        EXX
        DEC    C
        EXX
        INC    DE
        CALL   A5939
A6301:
        CALL   A2354
        LD     B,H
        LD     C,L
        POP    HL
        JP     C,A7936
        LD     (HL),C
        INC    HL
        LD     (HL),B
        RET
;*****
;LET $ variable=expression
;

```

A6314:

```
PUSH BC
EXX
DEC C
EXX
INC DE
CALL A5939
POP BC
LD HL, (A16162)
LD A, L
LD (BC), A
INC BC
LD A, H
LD (BC), A
DEC BC
LD (HL), C
INC HL
LD (HL), B
RET
```

;*****

;TRACE command.

;

A6336:

```
EXX
SET 7, B
EXX
RET
```

;*****

;NOTRACE command.

;

A6341:

```
EXX
RES 7, B
EXX
```

A6345:

```
RET ;common RET used by many routines
```

;*****

;BREAK command.

```

;
A6346:
    EXX
    RES    4,B
    EXX
    RET
;*****
;NOBREAK command.
;
A6351:
    EXX
    SET    4,B
    EXX
    RET
;*****
;NEW command.
;
A6356:
    CALL   A5975
    CALL   A11892
    LD     HL,NVARS                ;number of variable
commands
    LD     (A16107),HL            ;save it
    CALL   A8141
    EXX
    SET    2,B
    EXX
    JP     A16035
;*****
;STOP command.
;
A6378:
    EXX
    INC    DE
    INC    DE
    EXX
    LD     A,20
    JP     A7950

```


;*****

;CONT command.

;

A6387:

```
    EXX
    LD    A,B
    AND   68
    JP    NZ,A6417
    SET   6,B
    LD    DE,(A16122)
    PUSH DE
    LD    HL,(A16124)
    LD    A,(DE)
    LD    C,A
    EXX
    POP   DE
    LD    A,D
    OR    E
    JP    Z,A6418
    DEC   DE
    JP    A6216
```

;*****

A6417:

```
    EXX
```

A6418:

```
    LD    A,21
    JP    A7950
```

;*****

;Primary word vector table. Token order.

;

A6423:

```
    DW    A6247           ;implicit LET
    DW    A8427           ;GOSUB
    DW    A8342           ;GOTO
    DW    A8957           ;INPUT
    DW    A6247           ;LET
    DW    A8811           ;NEXT
    DW    A7854           ;PRINT, ?
```

DW	A9499	; READ
DW	A8419	; REM
DW	A8557	; FOR
DW	A7705	; IF
DW	A8419	; DATA
DW	A6942	; DIM
DW	A8381	; ON
DW	A8114	; ONERR
DW	A6378	; STOP
DW	A8477	; RETURN
DW	A6047	; END
DW	A8244	; DEF
DW	A8141	; CLEAR
DW	A28986	; RESUME
DW	A6356	; NEW
DW	A8493	; POP
DW	A6159	; RUN
DW	A7407	; LIST
DW	A6336	; TRACE
DW	A6341	; NOTRACE
DW	A7555	; DEL
DW	A30169	; CALL
DW	A6387	; CONT
DW	A8109	; CLRERR
DW	A9378	; GET
DW	A31963	; POKE
DW	A29088	; RESTORE
DW	A11090	; HOME
DW	A11358	; DRAW
DW	A11412	; XDRAW
DW	A11050	; FLASH
DW	A11055	; INVERSE
DW	A11060	; NORMAL
DW	A33631	; TEXT
DW	A31465	; GR
DW	A31471	; HGR
DW	A11080	; HGR2
DW	A11233	; HLIN


```

SCF                                ;sets carry, FPA2
EX  AF,AF'
EX  DE,HL
LD  A,(HL)
AND  3
LD  B,A
INC  HL
LD  C,(HL)
INC  HL
LD  D,0
LD  E,(HL)
ADD  HL,DE
EX  DE,HL
LD  A,L
EXX
INC  A
SUB  C
CPL
LD  C,A
EXX
LD  H,B
LD  L,C
ADD  HL,BC
ADD  HL,HL
ADD  HL,BC
LD  BC,(A16095)
ADD  HL,BC
LD  A,(HL)
INC  HL
LD  C,(HL)
INC  HL
LD  B,(HL)
INC  B
DEC  B
JR  Z,A6646
AND  252
JR  NZ,A6613

```

A6605:

```

        POP     HL
A6606:
        EX     AF,AF'
        LD     A,(HL)
        INC   HL
        LD     H,(HL)
        LD     L,A
        EX     AF,AF'
        JP     (HL)
;*****
A6613:
        JP     M,A6635
        BIT   3,A
        JP     NZ,A6781
        BIT   5,A
A6623:
        DEC   HL
A6624:
        LD     B,H
        LD     C,L
A6626:
        POP   HL
        INC   HL
        INC   HL
A6629:
        JR    Z,A6606
        INC   HL
        INC   HL
        JR    A6606
;*****
A6635:
        POP   HL
        PUSH  BC
        LD   BC,6
        ADD  HL,BC
        POP  BC
        BIT  6,A
        JR  A6629

```

;*****

A6646:

```
AND    252
JR     NZ,A6679
PUSH   HL
LD     HL,5
CALL   A7311
LD     HL,(A16109)
LD     (HL),0
DEC    HL
DEC    HL
DEC    HL
DEC    HL
LD     B,H
LD     C,L
DEC    HL
LD     (A16109),HL
POP    HL
LD     (HL),B
DEC    HL
LD     (HL),C
XOR    A
JR     A6605
```

;*****

A6679:

```
JP     M,A6635
BIT    3,A
JR     NZ,A6729
BIT    5,A
JR     Z,A6623
LD     BC,A16210
LD     (HL),B
DEC    HL
LD     (HL),C
JR     A6624
```

;*****

A6698:

```
EX     DE,HL
```

```
LD      A, (HL)
AND     3
LD      B, A
INC     HL
LD      C, (HL)
INC     HL
LD      D, 0
LD      E, (HL)
ADD     HL, DE
EX      DE, HL
LD      A, L
EXX
INC     A
SUB     C
CPL
LD      C, A
EXX
LD      H, B
LD      L, C
ADD     HL, BC
ADD     HL, HL
ADD     HL, BC
LD      BC, (A16095)
ADD     HL, BC
RET
```

```
;*****
```

```
A6729:
```

```
PUSH   DE
PUSH   HL
PUSH   AF
CALL   A7173
POP    AF
LD     B, C
LD     HL, 1
LD     DE, 11
```

```
A6743:
```

```
CALL   A7066
JP     C, A7927
```

```
CALL A7299
PUSH DE
DJNZ A6743
LD E,A
AND 48
LD A,E
LD DE,2
JR NZ,A6767
LD DE,5
```

A6767:

```
CALL A7066
JP C,A7927
CALL A7097
POP HL
LD (HL),B
DEC HL
LD (HL),C
POP DE
```

A6781:

```
PUSH AF
EXX
DEC C
EXX
INC DE
LD A,(DE)
CP 183
JP NZ,A6925
EX AF,AF'
JR C,A6829
CALL A6846
```

A6798:

```
POP AF
PUSH DE
LD D,A
EX AF,AF'
LD A,D
AND 48
LD D,H
```



```

        LD      E,L
        JP      NZ,A6812
        ADD    HL,DE
        ADD    HL,HL
A6812:
        ADD    HL,DE
        ADD    HL,BC
        LD     B,H
        LD     C,L
        POP   DE
        JR     NZ,A6823
        EX    AF,AF'
        JP    A6605
;*****
A6823:
        EX    AF,AF'
        BIT   5,A
        JP    A6626
;*****
A6829:
        CALL  A4789
        CALL  A6846
        LD   (A16139),HL
        CALL A4835
        LD   HL,(A16139)
        JR   A6798
;*****
A6846:
        LD   HL,0
        PUSH HL
        LD   H,B
        LD   L,C
        LD   B,(HL)
        INC  HL
        JP   A6867
;*****
A6857:
        EX   (SP),HL

```

EXX
DEC C
EXX
INC DE
LD A, (DE)
CP 185
JR NZ, A6925

A6867:

PUSH BC
PUSH HL
CALL A5939
JR NC, A6925
OR A
JR NZ, A6925
POP HL
POP BC
LD C, E
LD A, D
LD E, (HL)
INC HL
LD D, (HL)
INC HL
EX (SP), HL
CALL A7066
JR C, A6928
PUSH HL
PUSH AF
CALL A2354
JR C, A6928
POP AF
INC H
DEC H
JP M, A6928
EX DE, HL
DEC HL
SBC HL, DE
JR C, A6928
POP HL

```

        ADD    HL,DE
        LD     D,A
        LD     E,C
        DJNZ  A6857
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        CP   184
        POP  BC
        RET  Z
A6925:
        LD    A,1
        DB   33           ;LD HL,2366 for entry here
(garbage)
                                ;I hate overlapping code!
A6928:
        LD    A,9
A6930:
        JP   A7950
;*****
;DIM array().  Entry is at A6942.
;
A6933:
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        CP   185
        JR   NZ,A6925
A6942:
        EXX
        DEC   C
        EXX
        INC   DE
        CALL A6698

```

```
LD      B, (HL)
INC     HL
LD      A, (HL)
INC     HL
OR      (HL)
JP      NZ, A7924
OR      B
JP      M, A7942
DEC     HL
PUSH    HL
EXX
DEC     C
EXX
INC     DE
LD      A, (DE)
CP      183
JR      NZ, A6925
LD      HL, 1
LD      C, 0
```

A6977:

```
PUSH    BC
PUSH    HL
CALL    A5939
POP     HL
POP     BC
JR      NC, A6928
OR      A
JR      NZ, A6928
CALL    A7299
PUSH    DE
EX      DE, HL
CALL    A2354
EX      DE, HL
JR      C, A6928
INC     D
DEC     D
JP      M, A6928
INC     DE
```

```
CALL A7066
JP C,A7927
EX DE,HL
EX (SP),HL
EX DE,HL
INC C
EXX
DEC C
EXX
INC DE
LD A,(DE)
CP 185
JP Z,A6977
CP 184
JR NZ,A6925
LD A,B
PUSH DE
LD E,A
AND 48
LD A,E
LD DE,2
JR NZ,A7044
LD DE,5
```

A7044:

```
CALL A7066
JP C,A7927
POP DE
CALL A7097
POP HL
LD (HL),C
INC HL
LD (HL),B
EXX
LD A,C
EXX
OR A
RET Z
JP A6933
```

```
;*****  
;Multiply HL*DE. Result in HL. CF=1 if overflow  
(>65535).
```

```
;
```

```
A7066:
```

```
    PUSH    DE  
    PUSH    BC  
    PUSH    AF  
    LD      C,H  
    LD      A,L  
    LD      HL,0  
    LD      B,16
```

```
A7076:
```

```
    SRL     C  
    RRA  
    JP      NC,A7085  
    ADD     HL,DE  
    JR      C,A7092
```

```
A7085:
```

```
    EX      DE,HL  
    ADD     HL,HL  
    EX      DE,HL  
    DJNZ    A7076  
    LD      A,H  
    RLA
```

```
A7092:
```

```
    POP     BC  
    LD      A,B  
    POP     BC  
    POP     DE  
    RET
```

```
;*****
```

```
;Something with strings.
```

```
;
```

```
A7097:
```

```
    EX      (SP),HL  
    LD      (A16141),HL  
    LD      (A16139),DE
```

```
POP    HL
LD     B,C
PUSH  BC
PUSH  AF
LD     A,C
SCF
ADC    A,A
LD     C,A
LD     B,0
ADD   HL,BC
CALL  A7311
OR     A
SBC   HL,BC
POP   AF
BIT   5,A
LD    B,H
LD    C,L
LD    HL,(A16109)
LD    DE,0
JR    Z,A7138
LD    DE,A16210
```

A7138:

```
LD    (HL),D
CPD
JP    PO,A7150
LD    (HL),E
CPD
JP    PE,A7138
```

A7150:

```
POP   BC
```

A7151:

```
POP   DE
LD    (HL),D
DEC   HL
LD    (HL),E
DEC   HL
DJNZ  A7151
LD    (HL),C
```

```

        LD      B,H
        LD      C,L
        DEC     HL
        LD      (A16109),HL
        LD      DE,(A16139)
        LD      HL,(A16141)
        JP      (HL)
;*****
;Check DATA length.
;On exit, C=number of commas+1 (thus the number of
data elements).
;Error if more than 256 commas.
;
A7173:
        LD      BC,0
        JR      A7186
;*****
A7178:
        INC     DE
        LD      A,(DE)
        INC     DE
        ADD     A,E
        LD      E,A
        JR      NC,A7186
        INC     D
A7186:
        LD      A,(DE)
        INC     DE
        CP      185
        JR      Z,A7241
        CP      184
        JR      Z,A7259
        CP      183
        JR      Z,A7251
        CP      138
        JP      C,A7186
        JR      Z,A7237
        CP      139

```



```
JR      Z,A7236
CP      144
JR      C,A7178
CP      146
JR      Z,A7233
JP      NC,A7186
LD      A,(DE)
INC     DE
ADD     A,E
LD      E,A
JP      NC,A7186
INC     D
JP      A7186
```

```
;*****
```

```
A7233:
```

```
INC     DE
INC     DE
INC     DE
```

```
A7236:
```

```
INC     DE
```

```
A7237:
```

```
INC     DE
JP      A7186
```

```
;*****
```

```
A7241:
```

```
LD      A,B
DEC     A
JR      NZ,A7186
```

```
A7245:
```

```
INC     C
JR      NZ,A7186
JP      A6925
```

```
;*****
```

```
A7251:
```

```
INC     B
JR      NZ,A7186
LD      A,14
JP      A7950
```

;*****

A7259:

```
    DEC    B
    JR     NZ,A7186
    INC    C
    RET
```

;*****

;Make string definition.

;On entry, A=string length, HL=address of variable name.

;On exit, HL=start of string space and (A16111)=end of string space.

;

A7264:

```
    PUSH   BC
    PUSH   HL
    LD     C,A
    LD     H,0
    ADD    A,3
    LD     L,A
    JR     NC,A7275
    INC    H
```

A7275:

```
    CALL   A7311
    LD     A,C
    LD     BC,(A16111)
    ADD    HL,BC
    LD     (A16111),HL
    LD     H,B
    LD     L,C
    POP    BC
    LD     (HL),C
    INC    HL
    LD     (HL),B
    INC    HL
    LD     (HL),A
    DEC    HL
    DEC    HL
```

```

        POP    BC
        RET
;*****
;Check stack.  If SP drops below 53760, then Stack
Overflow error.
;
A7299:
        PUSH   HL
        LD     HL,11776      ;not an address, but a size
                               ;need to find out why it has
this value
        ADD   HL,SP
        POP   HL
        RET   C
A7306:
        LD    A,7
        JP    A7950
;*****
;Check string space.
;If string table is too long, a garbage collection
is done.
;If string table is still too long, Out of Memory
error.
;
A7311:
        PUSH   BC
A7312:
        PUSH   HL
        LD     BC,(A16111)
        ADD   HL,BC
        JP    C,A7332
        LD     BC,(A16109)
        SBC   HL,BC
        POP   HL
        POP   BC
        RET   C
        PUSH  BC
        PUSH  HL

```

```

A7332:
    CALL    A10209        ;part of FRE to do garbage
collection
    POP     HL
    PUSH    HL
    LD      BC,(A16111)
    ADD     HL,BC
    JP      C,A7927
    LD      BC,(A16109)
    SBC     HL,BC
    POP     HL
    POP     BC
    RET     C
    JP      A7927
;*****
;Print program.  Used by LIST command.
;
A7357:
    PUSH    DE
    CALL    A12128
    LD      HL,(A16089)
    LD      BC,(A16091)
    JR      A7381
;*****
A7370:
    CALL    A13459
    PUSH    BC
    PUSH    HL
    CALL    A6222
    POP     HL
    POP     BC
    DEC     BC
A7381:
    LD      A,B
    OR      C
    JR      NZ,A7370
    POP     DE
    RET

```

```
;*****  
;Part of LIST command.  
;  
A7387:
```

```
    EXX  
    LD    A,C  
    EXX  
    OR    A  
    JR    Z,A7357  
    EXX  
    DEC   C  
    EXX  
    INC   DE  
    LD    A,(DE)  
    CALL  A1519  
    LD    HL,0  
    JP    A7465
```

```
;*****  
;LIST command.  
;  
A7407:
```

```
    EXX  
    LD    A,C  
    EXX  
    OR    A  
    JR    Z,A7357  
    LD    HL,(A16091)  
    LD    A,H  
    OR    L  
    JP    Z,A8419  
    EXX  
    DEC   C  
    EXX  
    INC   DE  
    LD    A,(DE)  
    CP    185  
    JR    Z,A7387  
    CP    161
```

JR Z,A7387
CALL A1519
EXX
LD A,C
EXX
OR A
JP Z,A7535
EXX
DEC C
EXX
INC DE
LD H,B
LD L,C
EXX
LD A,C
EXX
OR A
JP Z,A7549
EXX
DEC C
EXX
INC DE
LD A,(DE)
CALL A1519

A7465:

PUSH HL
OR A
SBC HL,BC
POP HL
JP Z,A7535
RET NC
LD (A16207),HL
PUSH BC
CALL A12528
POP BC
LD (A16207),BC
PUSH HL
CALL A12528

```

    POP    BC
    LD     A,L
    LD     L,C
    LD     C,A
    LD     A,H
    LD     H,B
    LD     B,A
    JP     C,A7511
    PUSH  HL
    OR     A
    SBC   HL,BC
    POP   HL
    JR    Z,A7539
    DEC   BC
    DEC   BC
    DEC   BC
    DEC   BC
A7511:
    PUSH  DE
A7512:
    PUSH  BC
    PUSH  HL
    CALL  A6222
    POP   HL
    POP   BC
    CALL  A13459
    PUSH  HL
    OR    A
    SBC  HL,BC
    POP  HL
    JP   C,A7512
    JP   Z,A7512
    POP  DE
    RET
;*****
A7535:
    LD   (A16207),BC
A7539:

```

```

        CALL    A12528
        RET     NC
        PUSH   DE
        CALL   A13459
        POP    DE
        RET
;*****
A7549:
        LD     BC,65535
        JP     A7465
;*****
;DEL command.
;
A7555:
        EXX
        LD     A,C
        EXX
        OR     A
        RET    Z
        LD     HL,(A16091)
        LD     A,H
        OR     L
        JP     Z,A8419
        EXX
        DEC    C
        EXX
        INC    DE
        LD     A,(DE)
        CP    185
        JP     Z,A7948
        CP    161
        JP     Z,A7948
        CALL  A1519
        EXX
        LD     A,C
        EXX
        OR     A
        JP     Z,A7695

```



```
LD      H,B
LD      L,C
EXX
DEC     C
LD      A,C
DEC     C
EXX
OR      A
INC     DE
INC     DE
JP      Z,A7948
LD      A,(DE)
CALL   A1519
LD      A,B
LD      B,H
LD      H,A
LD      A,C
LD      C,L
LD      L,A
PUSH   HL
OR      A
SBC    HL,BC
POP    HL
JP      Z,A7695
RET    C
PUSH   DE
PUSH   HL
PUSH   BC
LD      (A16207),HL
CALL   A12528
JR      NC,A7684
JR      A7674
```

```
;*****
```

```
A7638:
```

```
PUSH   HL
PUSH   BC
LD      (A16207),HL
CALL   A12528
```

```
LD     DE,(A16089)
PUSH  HL
OR     A
SBC   HL,DE
POP   HL
JR    Z,A7689
DEC   HL
DEC   HL
DEC   HL
LD    D,(HL)
DEC   HL
LD    E,(HL)
EX    (SP),HL
EX    DE,HL
PUSH  HL
OR     A
SBC   HL,DE
POP   HL
EX    DE,HL
EX    (SP),HL
JR    C,A7689
```

A7674:

```
CALL  A12786
LD    HL,(A16091)
LD    A,H
OR    L
JR    Z,A7689
```

A7684:

```
POP   BC
POP   HL
JP    A7638
```

;*****

A7689:

```
POP   BC
POP   HL
POP   DE
JP    A16035
```

;*****

A7695:

```
PUSH  DE
LD    H,B
LD    L,C
CALL  A12779
POP   DE
JP    A16035
```

;*****

;IF {condition} command.

;Evaluates {condition}. If {condition} is true
(A<>0), it returns

;to the execution loop if THEN follows {condition};
if no THEN it

;assumes GOTO. If {condition} is false (A=0), then
it GOTOS the

;next line number.

;

A7705:

```
CALL  A5939
OR    A
JP    NZ,A7723
LD    A,(A16166)
OR    A
JP    Z,A7734
DEC   A
JP    Z,A7734
```

A7723:

```
EXX
DEC   C
EXX
INC   DE
LD    A,(DE)
CP    178
RET   Z
JP    A8342 ;GOTO
```

;*****

A7734:

```
EXX
```

```

        CALL  A6016
        DEC   DE
        RET
;*****
;TAB(n) function.  Only works as part of a PRINT
statement.
;
A7740:
        EXX
        DEC   C
        DEC   C
        EXX
        INC   DE
        INC   DE
        CALL  A1500
        CALL  A26177
        CP    L
        JP    NC,A7787
        NEG
        ADD   A,L
        LD    L,A
        DEC   L
        JP    NZ,A7779
        JP    A7787
;*****
;SPC(n) function.  Only works as part of a PRINT
statement.
;
A7767:
        EXX
        DEC   C
        DEC   C
        EXX
        INC   DE
        INC   DE
        CALL  A1500
        OR    L
        JR    Z,A7787

```

```

A7779:
    LD      A,32
A7781:
    CALL   A11994
    DEC    L
    JR     NZ,A7781
A7787:
    EXX
    DEC    C
    EXX
    INC    DE
    LD     A,(DE)
    CP     184
    JP     Z,A7892
    JP     A7948
;*****
A7800:
    EXX
    DEC    C
    EXX
    INC    DE
    LD     A,(DE)
    CP     185
    JP     Z,A7875
    CP     186
    JP     Z,A7892
    CP     140
    JP     NZ,A7837
    EXX
    DEC    C
    EXX
    INC    DE
    LD     A,(DE)
    OR     A
    JP     Z,A7767
    DEC    A
    JP     Z,A7740
    EXX

```

```

        INC     C
        EXX
        DEC     DE
A7837:
        EXX
        INC     C
        EXX
        DEC     DE
        CALL    A5939
        JP      NC,A7800
        OR      A
        JP      NZ,A7864
        CALL    A1553
;*****
;PRINT command.
;
A7854:
        EXX
        LD      A,C
        EXX
        OR      A
        JP      NZ,A7800
        JP      A12128
;*****
A7864:
        LD      HL,(A16162)
        INC     HL
        INC     HL
        CALL    A12110
        JP      A7854
;*****
A7875:
        CALL    A26177
        AND     15
        SUB     16
        LD      C,A
        LD      A,32
A7885:

```

```

        CALL    A11994
        INC     C
        JP      NZ,A7885
A7892:
        EXX
        LD      A,C
        EXX
        OR      A
        JP      NZ,A7800
        RET
;*****
;Print error strings.
;
A7900:
        LD      HL,A7900
        PUSH   HL
        LD      IX,0
        ADD    IX,SP
        PUSH   IX
        POP    IY
;*****
;Error code table.
;Fall-through into a series of meaningless LD
BC,nnnn statements.
;Have I mentioned already that I hate overlapping
code?
;
        DB      1
A7915:
        LD      A,17          ;Illegal Mode
        DB      1
A7918:
        LD      A,11          ;Divide By Zero
        DB      1
A7921:
        LD      A,5           ;Overflow
        DB      1
A7924:

```

```

        LD      A,10          ;Redimensioned Array
        DB      1
A7927:
        LD      A,6          ;Out Of Memory
        DB      1
A7930:
        LD      A,3          ;Out Of Data
        DB      1
A7933:
        LD      A,15         ;Undefined Function
        DB      1
A7936:
        LD      A,4          ;Illegal Quantity
        DB      1
A7939:
        LD      A,12         ;Type Mismatch
        DB      1
A7942:
        LD      A,16         ;Incorrect Function Usage
        DB      1
A7945:
        LD      A,13         ;String Too Long
        DB      1
A7948:
        LD      A,1          ;Next Without FOR
A7950:
        LD      (A16128),A    ;entry here prints error
message for code in A
        EXX
        LD      (A16122),DE
        LD      (A16124),HL
        SET    2,B           ;clear GOSUBs and FOR/NEXT
loops
        EXX
        CP      20           ;was it STOP or END?
        JR      NC,A7975     ;YES
        EXX
        BIT    0,B           ;NO, so is ONERR enabled?

```



```

        EXX
        JP      NZ,A8082      ;YES, so do RESUME
A7975:
        CALL   A24419        ;CLOSE files after error
        CALL   A12128        ;print CR
        LD     A,63          ;"?"
        CALL   A11994        ;print it
        LD     A,(A16128)    ;get error code
        CP     19            ;was it ^C?
        JR     NZ,A7995      ;NO
        LD     A,20          ;YES, so make it into Break
A7995:
        JP     A28861        ;^C fix
;*****
A7998:
        LD     B,A           ;make into a counter
        LD     HL,A1152      ;base of error string table
        OR     A             ;is it error 0? (NEXT
without FOR, the 1st string)
        JR     Z,A8014      ;YES, so skip offsetting
A8005:
        LD     A,(HL)        ;NO, so get length of string
        INC   HL            ;point to 1st character
        ADD   A,L           ;offset into table
        LD     L,A
        JR     NC,A8012
        INC   H
A8012:
        DJNZ  A8005         ;keep offsetting
A8014:
        CALL  A12110        ;print error string at HL
        LD   A,(A16128)    ;get error number
        CP   20            ;do we need to print
"Error"?
        JR   NC,A8030      ;NO, so skip it
A8024:
        LD   HL,A8097      ;YES, so get "Error" string
        CALL A12110        ;print it

```

A8030:

```
    EXX
    BIT     6,B           ;are we in program mode?
    EXX
    JR      Z,A8055      ;NO immediate, so skip
    LD      HL,A8104     ;YES, so get "In" string
    CALL    A12110      ;print it
```

```
    EXX
    PUSH   HL
```

```
    EXX
```

```
    POP    HL
```

```
    DEC    HL
```

```
    DEC    HL
```

```
    DEC    HL
```

```
    LD     D,(HL)
```

```
    DEC    HL
```

```
    LD     E,(HL)
```

```
    CALL   A12967      ;print the line number of
```

the error

A8055:

```
    CALL   A12128      ;print CR
```

```
    LD     HL,0
```

```
    LD     (A16162),HL
```

```
    LD     (A16171),HL
```

```
    LD     A,(A16128)  ;get error code
```

```
    CP     20          ;was it Break?
```

```
    JP     NZ,A16035   ;NO, so reset stack and exit
```

to prompt

```
    EXX
```

```
    RES    2,B         ;YES, so leave GOSUBs and
```

FOR/NEXT loops intact

```
    EXX
```

```
    JP     A16038      ;exit to prompt, all stacks
```

and pointers saved

```
;*****
```

A8082:

```
    JP     A8313       ;RESUME entry
```

```
;*****
```

```

A8085:
    EXX
    LD    BC,A6216    ;return address in execution
loop
    PUSH BC
    LD    BC,(A16126) ;get line# to RESUME to
    JP    A8350      ;part of GOTO
;*****
A8097:
    DB    6
    DB    " Error"
A8104:
    DB    4
    DB    " In "
;*****
;CLRERR command.
;
A8109:
    EXX
    RES   0,B        ;disable ONERR
    EXX
    RET
;*****
;ONERR GOTO command.
;
A8114:
    EXX
    BIT   6,B        ;are we in program mode?
    JR    Z,A8137    ;NO, immediate mode, so
Illegal Mode error
    DEC   C          ;YES, so skip GOTO token
    DEC   C
    EXX
    INC   DE
    INC   DE
    LD    A,(DE)
    CALL A1519      ;get line number
    LD    (A16126),BC ;save it

```

```

        EXX
        SET    0,B           ;enable ONERR
        EXX
        RET
;*****
A8137:
        EXX
        JP    A7915         ;Illegal Mode error
;*****
;CLEAR command.
;
A8141:
        CALL  A9482
A8144:
        LD    HL,(A16089)
        DEC   HL
        LD    (A16109),HL
        LD    HL,(A16095)
        INC   HL
        LD    BC,VTABLEN   ;length of permanent
variable table (orig. 355)
                                ;this is computed at the end
of the listing
        ADD   HL,BC
        LD    BC,(A16107)
        LD    A,B
        OR    A
        JR    NZ,A8172
        LD    A,NVARS      ;# of permanent variables
(orig. 71)
                                ;make sure to change this if
you add new var. cmds.!
        CP    C
        JR    Z,A8194
A8172:
        NEG
        ADD   A,C
        LD    C,A

```

```

        JR      C,A8179
        DEC     B
A8179:
        XOR     A
        PUSH   DE
        LD     D,A
        LD     E,3
A8184:
        LD     (HL),A
        INC    HL
        LD     (HL),A
        CPI
        ADD    HL,DE
        JP     PE,A8184
        POP    DE
A8194:
        DEC    HL
        LD     A,(A16097)
        CP     L
        JR     NZ,A8207
        LD     A,(A16098)
        CP     H
        JR     Z,A8234
A8207:
        PUSH   DE
        LD     D,H
        LD     E,L
        LD     BC,(A16097)
        PUSH   BC
        LD     (A16097),HL
        LD     HL,(A16099)
        OR     A
        SBC    HL,BC
        LD     B,H
        LD     C,L
        POP    HL
        LDIR
        LD     (A16099),DE

```

```

        POP     DE
A8234:  LD      HL, (A16099)
        LD      (A16115), HL
        LD      (A16111), HL
        RET
;*****
;DEF FN(n)={statement} command.
;
A8244:  EXX
        DEC     C
        BIT     6, B           ;are we in program mode?
        EXX
        INC     DE
        JP      Z, A7915       ;NO, so Illegal Mode error
        CALL   A6698
        LD      A, (HL)
        AND     48
        JP      NZ, A7948
        INC     HL
        PUSH   HL
        LD      HL, A7948
        PUSH   HL
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A, (DE)
        CP     183
        RET    NZ
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A, (DE)
        SUB    140
        CP     4

```

```

RET    NC
CALL   A6698
LD     A, (HL)
AND    252
RET    NZ
EXX
DEC    C
EXX
INC    DE
LD     A, (DE)
CP     184
RET    NZ
POP    HL
POP    HL
EXX
PUSH   DE
EXX
POP    BC
LD     (HL), C
INC    HL
LD     (HL), B
JP     A8419

```

```
;*****
```

```
;RESUME or RESUME line# command.
```

```
;
```

```
A8313:
```

```

CP     19                ;was it ^C?
JP     Z, A7975          ;YES, so don't RESUME
EXX
BIT    1, B              ;NO, so was ONERR executing?
JR     NZ, A8328         ;YES, so ignore
SET    1, B              ;NO, so make ONERR execute

```

```
now
```

```
JP     A8085            ;GOTO the line#
```

```
;*****
```

```
A8328:
```

```

RES    0, B              ;disable ONERR
EXX

```

```

        JP      A7975
;*****
A8334:
        DW      0      ;DE storage
A8336:
        DW      0      ;HL storage
A8338:
        DW      0      ;IX storage
A8340:
        DW      0      ;IY storage
;*****
;GOTO line# command.
;
A8342:
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A,(DE)
        CALL   A1519
A8350:
        LD      (A16207),BC
        CALL   A12528
        JP      NC,A8376
A8360:
        PUSH   HL
        EXX
        POP    HL
        SET    6,B
        CALL   A6016
        DEC   DE
        RET
;*****
A8370:
        POP    HL
        POP    HL
        POP    IX
        POP    IY

```



```

A8376:
    LD    A,8
    JP    A7950
;*****
;ON value GOTO/GOSUB line# command.
;
A8381:
    CALL  A1500
    LD    A,L
    OR    A
    JP    Z,A8419
    EXX
    DEC   C
    EXX
    INC   DE
    LD    A,(DE)
    CP    174
    JP    NZ,A8516
A8399:
    LD    A,L
    DEC   A
    JP    Z,A8342
    DEC   L
A8405:
    EXX
    DEC   C
    EXX
    INC   DE
    LD    A,(DE)
    RET   Z
    CP    185
    JP    NZ,A8405
    JP    A8399
;*****
;REM or DATA commands.
;
A8419:
    EXX

```

```
LD    A,C
EXX
ADD   A,E
LD    E,A
RET   NC
INC   D
RET
```

```
;*****
```

```
;GOSUB line# command.
```

```
;
```

```
A8427:
```

```
CALL  A7299
PUSH  IY
PUSH  IX
EXX
PUSH  HL
EXX
EXX
DEC   C
EXX
INC   DE
LD    A,(DE)
CALL  A1519
```

```
A8445:
```

```
PUSH  DE
LD    (A16207),BC
CALL  A12528
JP    NC,A8370
LD    DE,A8482
PUSH  DE
LD    IX,0
ADD   IX,SP
PUSH  HL
EXX
POP   HL
SET   6,B
CALL  A6016
JP    A6191
```

```

;*****
;RETURN or RETURN line# command.
;
A8477:
    LD     SP,IX           ;points SP at return address
    LD     A,2             ;code for RETURN without
GOSUB error, if needed
    RET                    ;returns to the address on
the stack
;*****
;This is the address on the stack above.
;
A8482:
    JP     A29066         ;check for RETURN line#
;*****
    NOP
A8486:
    POP    HL
    EXX
    POP    IX
    POP    IY
    RET
;*****
;POP command.
;
A8493:
    LD     SP,IX
    POP    HL
    LD     BC,A8482
    PUSH  HL
    OR     A
    SBC   HL,BC
    POP    HL
    JR     NZ,A8513
    POP    HL
    POP    HL
    POP    IX
    POP    IY

```

```

        RET
;*****
A8513:
        LD     A,2
        JP     (HL)
;*****
;GOSUB handler for ON value GOSUB line#
;
A8516:
        LD     A,L
        DEC    A
        JP     Z,A8536
        DEC    L
A8522:
        EXX
        DEC    C
        EXX
        INC    DE
        LD     A,(DE)
        RET    Z
        CP     185
        JP     NZ,A8522
        JP     A8516
;*****
A8536:
        EXX
        DEC    C
        EXX
        INC    DE
        LD     A,(DE)
        CALL  A1519
        CALL  A8419
        PUSH  IY
        PUSH  IX
        EXX
        PUSH  HL
        EXX
        JP     A8445

```

```

;*****
;FOR FP variable=value1 TO value2 command.
;
A8557:
    CALL    A7299
    PUSH    IX
    PUSH    IY
    CALL    A6555
;*****
;FOR command vector table.
;Only FP variables are allowed; others give Syntax
Error.
;
    DW      A8577 ;FP
    DW      A7948 ;%
    DW      A7948 ;$
    DW      A7948 ;FN
    DW      A7948 ;command
;*****
;FOR FP variables.
;
A8577:
    OR      A
    JP      NZ,A7948
    PUSH    BC
    PUSH    BC
    PUSH    IY
    POP     HL
    PUSH    IX
    POP     BC
    PUSH    HL
    OR      A
    SBC    HL,BC
    POP     HL
    JP      NC,A8644
A8597:
    LD      BC,16
    ADD     HL,BC

```

```
POP    BC
PUSH   BC
INC    HL
LD     A,B
CP     (HL)
DEC    HL
JR     NZ,A8621
LD     A,C
CP     (HL)
JP     NZ,A8621
LD     SP,HL
POP    BC
PUSH   BC
PUSH   BC
JP     A8644
```

```
;*****
```

```
A8621:
```

```
INC    HL
INC    HL
LD     C,(HL)
INC    HL
LD     A,(HL)
INC    HL
INC    HL
CP     (HL)
DEC    HL
JR     NZ,A8634
LD     A,C
CP     (HL)
```

```
A8634:
```

```
JR     NC,A8644
DEC    HL
LD     A,(HL)
DEC    HL
LD     L,(HL)
LD     H,A
JP     A8597
```

```
;*****
```

A8644:

```
EXX
DEC    C
EXX
INC    DE
CALL   A5939
POP    HL
LD     BC,(A16162)
LD     (HL),C
INC    HL
LD     (HL),B
INC    HL
LD     BC,(A16164)
LD     (HL),C
INC    HL
LD     (HL),B
INC    HL
LD     A,(A16166)
LD     (HL),A
EXX
DEC    C
EXX
INC    DE
CALL   A5939
LD     HL,(A16165)
PUSH   HL
LD     HL,(A16163)
PUSH   HL
LD     A,(A16162)
PUSH   AF
INC    SP
EXX
LD     A,C
EXX
OR     A
JP     Z,A8738
EXX
DEC    C
```

```

    EXX
    INC     DE
    CALL   A5939
    LD     HL,(A16165)
    PUSH  HL
    LD     HL,(A16163)
    PUSH  HL
    LD     A,(A16162)
    PUSH  AF
A8718:
    INC     SP
    EXX
    PUSH  HL
    SET   5,B
    EXX
    PUSH  DE
    LD     HL,A8753
    PUSH  HL
    LD     IY,0
    ADD   IY,SP
    JP    A6216
;*****
A8738:
    LD     BC,33024      ;not an address
                                ;need to figure out what it
is
    PUSH  BC
    LD     B,0
    PUSH  BC
    PUSH  BC
    JP    A8718
;*****
A8749:
    EXX
    JP    A7948
;*****
;NEXT routines
;
```


A8753:

```
    EXX
    LD    A,B
    AND   96
    JR    Z,A8749
    LD    A,C
    EXX
    OR    A
    JP    Z,A8822
    PUSH DE
    EXX
    PUSH BC
    EXX
    CALL A6555
```

;*****

;NEXT command vector table.

;Only FP variables are allowed; others give Syntax
Error.

;

```
    DW    A8782        ;FP
    DW    A7948        ;%
    DW    A7948        ;$
    DW    A7948        ;FN
    DW    A7948        ;command
```

;*****

;NEXT FP variables.

;

A8782:

```
    OR    A
    JP    NZ,A7948
    LD    A,(IY+17)
    CP    B
    JR    NZ,A8798
    LD    A,(IY+16)
    CP    C
    JR    Z,A8816
```

A8798:

```
    EXX
```

```

        POP     BC
        EXX
        POP     DE
        LD      HL,16
        ADD     HL,SP
        LD      SP,HL
        POP     IY
        POP     IX
;*****
;NEXT command entry.
;
A8811:
        LD      SP,IY
        LD      A,0
        RET
;*****
A8816:
        LD      H,B
        LD      L,C
        POP     BC
        POP     BC
        JR      A8830
;*****
A8822:
        LD      HL,14
        ADD     HL,SP
        LD      A,(HL)
        INC     HL
        LD      H,(HL)
        LD      L,A
A8830:
        PUSH    DE
        PUSH    HL
        LD      DE,A16162
        LDI
        LDI
        LDI
        LDI

```

```
LDI
LD    HL,8
ADD   HL,SP
LD    DE,A16171
LDI
LDI
LDI
LDI
LDI
EX    (SP),HL
PUSH  HL
CALL  A1583
POP   DE
LD    HL,A16162
LDI
LDI
LDI
LDI
LDI
POP   HL
LD    DE,A16171
LDI
LDI
LDI
LDI
LDI
CALL  A2488
POP   DE
PUSH  AF
BIT   7,(IY+9)
JR    NZ,A8932
POP   AF
JR    Z,A8911
JR    NC,A8936
```

A8911:

```
POP   DE
EXX
POP   HL
```

```

        INC     HL
        LD      A,H
        OR      L
        DEC     HL
        JR      Z,A8922
        SET     6,B
A8922:
        PUSH   HL
        EXX
        PUSH   DE
        LD      HL,A8753
        PUSH   HL
        JP      A6216
;*****
A8932:
        POP    AF
        JP     NC,A8911
A8936:
        LD     HL,16
        ADD   HL,SP
        LD     SP,HL
        POP   IY
        POP   IX
        EXX
        LD     A,C
        EXX
        OR    A
        RET   Z
        EXX
        DEC   C
        EXX
        INC   DE
        JP    A8811
;*****
;INPUT or LINPUT commands.
;
A8957:
        EXX

```

```

DEC    C
EXX
INC    DE
LD     A, (DE)
CP     145
JP     NZ, A8998
EXX
DEC    C
EXX
INC    DE
LD     A, (DE)
LD     H, D
LD     L, E
CALL   A12110
LD     A, (HL)
EXX
NEG
ADD    A, C
LD     C, A
EXX
LD     A, (HL)
ADD    A, E
LD     E, A
JP     NC, A8991
INC    D
A8991:
EXX
DEC    C
EXX
INC    DE
JP     A9007
;*****
A8998:
EXX
INC    C
EXX
DEC    DE
A9002:

```

```

        LD     A,63
        CALL  A11994
A9007:
        PUSH  DE
        CALL  A12159
        LD    B,D
        LD    C,E
        LD    A,(DE)
        OR    A
        JP    Z,A9029
        INC  DE
        LD    A,(DE)
        PUSH  HL
        LD    HL,A16134
        CP    (HL)
        POP  HL
        JP    Z,A6242
A9029:
        POP  DE
        LD    A,(BC)
        INC  BC
        OR    A
        JR    Z,A9043
        JR    A9042
;*****
A9037:
        LD    A,(BC)
        OR    A
        JP    Z,A9002
A9042:
        SCF
A9043:
        PUSH  BC
        PUSH  AF
        LD    (A16145),DE
        EXX
        LD    A,C
        EXX

```

```

        JP      A28769          ;check for LINPUT as well.
;*****
A9055:
        CALL   A6555
;*****
;INPUT vector table.
;FNs and variable commands give Illegal Function
Usage Error.
;
        DW     A9068 ;FP
        DW     A9336 ;%
        DW     A9229 ;$
        DW     A7942 ;FN
        DW     A7942 ;command
;
;*****
;INPUT FP variables.
;
A9068:
        POP    AF
        POP    HL
        JP     NC,A9172
        PUSH   DE
        LD     D,H
        LD     E,L
A9076:
        LD     A,(DE)
        CP    32
        JP    NZ,A9086
        INC   DE
        JP    A9076
;*****
A9086:
        PUSH   BC
        CALL  A2576
        POP    HL
        JP    C,A7936
        LD    B,D

```

```
LD      C,E
LD      DE,(A16162)
LD      (HL),E
INC     HL
LD      (HL),D
INC     HL
LD      DE,(A16164)
LD      (HL),E
INC     HL
LD      (HL),D
INC     HL
LD      A,(A16166)
LD      (HL),A
POP     DE
```

A9117:

```
LD      A,(BC)
CP      44
JR      Z,A9147
OR      A
JP      Z,A9135
CP      32
JP      NZ,A9172
INC     BC
JP      A9117
```

;*****

A9135:

```
EXX
LD      A,C
EXX
OR      A
RET     Z
EXX
DEC     C
EXX
INC     DE
JP      A9002
```

;*****

A9147:


```
INC    BC
EXX
LD     A,C
EXX
OR     A
JR     Z,A9161
EXX
DEC    C
EXX
INC    DE
JP     A9037
```

```
;*****
```

```
A9161:
```

```
LD     HL,A9205
CALL   A12110
CALL   A12128
RET
```

```
;*****
```

```
A9171:
```

```
POP    DE
```

```
A9172:
```

```
EXX
BIT    0,B
EXX
JR     NZ,A9200
LD     HL,A9220
CALL   A12110
LD     DE,(A16145)
LD     A,(A16147)
EXX
LD     C,A
EXX
CALL   A12128
JP     A9002
```

```
;*****
```

```
A9200:
```

```
LD     A,18
JP     A7950
```

```

;*****
A9205:
    DB    14
    DB    "?Extra Ignored"
A9220:
    DB    8
    DB    "?Reenter"
;*****
;INPUT $ variables.
;
A9229:
    POP   AF
    POP   HL
    PUSH  BC
    JR    NC,A9243
A9234:
    LD    A,(HL)
    CP    32
    JR    NZ,A9251
    INC   HL
    JP    A9234
;*****
A9243:
    LD    (A16137),DE
    LD    D,H
    LD    E,L
    JR    A9276
;*****
A9251:
    OR    A
    JR    Z,A9243
    CP    34
    JP    Z,A9320
    PUSH HL
    DB    62
                                ;LD A,35 in this frame
                                ;I hate overlapping code!
A9261:
    INC   HL

```

```
LD    A, (HL)
CP    44
JR    Z, A9271
OR    A
JP    NZ, A9261
```

A9271:

```
LD    (A16137), DE
POP   DE
```

A9276:

```
OR    A
SBC   HL, DE
LD    A, L
POP   BC
LD    H, B
LD    L, C
CALL  A7264
LD    A, L
LD    (BC), A
INC   BC
LD    A, H
LD    (BC), A
INC   HL
INC   HL
LD    A, (HL)
INC   HL
OR    A
EX    DE, HL
JR    Z, A9304
LD    C, A
LD    B, 0
LDIR
```

A9304:

```
LD    B, H
LD    C, L
LD    DE, (A16137)
LD    A, (BC)
CP    34
JP    NZ, A9117
```

```

        INC     BC
        JP      A9117
;*****
A9320:
        INC     HL
        PUSH    HL
A9322:
        LD      A, (HL)
        CP      34
        JP      Z, A9271
        OR      A
        JP      Z, A9271
        INC     HL
        JP      A9322
;*****
;INPUT % variables.
;
A9336:
        POP     AF
        EX      DE, HL
        EX      (SP), HL
        JP      NC, A9171
        PUSH    BC
A9343:
        LD      A, (HL)
        CP      32
        JR      NZ, A9352
        INC     HL
        JP      A9343
;*****
A9352:
        LD      D, H
        LD      E, L
        CALL    A2576
        JP      C, A7936
        CALL    A2354
        JP      C, A7936
        POP     BC

```

```
LD    A,L
LD    (BC),A
INC   BC
LD    A,H
LD    (BC),A
LD    B,D
LD    C,E
POP   DE
JP    A9117
```

```
;*****
```

```
;GET variable command.
```

```
;
```

```
A9378:
```

```
CALL  A12137
PUSH  AF
CALL  A6555
```

```
;*****
```

```
;GET command vector table.
```

```
;FNs and variable commands give Incorrect Function  
Usage Error.
```

```
;
```

```
DW    A9395      ;FP
DW    A9395      ;%
DW    A9464      ;$
DW    A7942      ;FN
DW    A7942      ;command
```

```
;
```

```
;*****
```

```
;GET FP or % variables.
```

```
;
```

```
A9395:
```

```
LD    L,A
POP   AF
CP    46
JR    Z,A9428
CP    69
JR    Z,A9428
CP    101
```

```
JR    Z,A9428
CP    45
JR    Z,A9428
CP    43
JR    Z,A9428
CP    48
JP    C,A7948
CP    58
JP    NC,A7948
DB    254
```

```
;CP 175  in this frame
;I hate overlapping code!
```

A9428:

```
XOR   A
AND   15
BIT   4,L
JR    NZ,A9459
PUSH  BC
LD    H,0
LD    L,A
OR    A
CALL  A2407
POP   HL
LD    BC,(A16165)
XOR   A
LD    (HL),A
INC   HL
LD    (HL),A
INC   HL
LD    (HL),A
INC   HL
LD    (HL),C
INC   HL
LD    (HL),B
RET
```

;*****

A9459:

```
LD    (BC),A
INC   BC
```

```

        XOR    A
        LD     (BC),A
        RET
;*****
;GET $ variables.
;
A9464:
        LD     H,B
        LD     L,C
        LD     A,1
        CALL  A7264
        LD     A,L
        LD     (BC),A
        INC   BC
        LD     A,H
        LD     (BC),A
        INC   HL
        INC   HL
        INC   HL
        POP   AF
        LD     (HL),A
        RET
;*****
;RESTORE or RESTORE line# commands.
;
A9482:
        LD     HL,(A16089)
A9485:
        LD     (A16117),HL
        XOR   A
        LD     (A16121),A
        LD     (A16119),A
        LD     (A16120),A
        RET
;*****
;READ variable command.
;
A9499:

```

```
LD    A, (A16121)
OR    A
JP    NZ, A9578
LD    HL, (A16117)
LD    BC, (A16119)
OR    B
OR    C
JP    NZ, A9536
```

A9518:

```
LD    BC, (A16101)
PUSH  HL
OR    A
SBC   HL, BC
POP   HL
JP    NC, A7930
INC   HL
INC   HL
LD    C, (HL)
INC   HL
LD    B, (HL)
INC   HL
```

A9536:

```
LD    A, (BC)
OR    A
JP    Z, A9518
INC   BC
EX    AF, AF'
LD    A, (BC)
CP    12
JP    Z, A9559
EX    AF, AF'
ADD   A, C
LD    C, A
JP    NC, A9536
INC   B
JP    A9536
```

;*****

A9559:


```
EX    AF,AF'
INC   BC
INC   BC
LD    A,(BC)
OR    A
INC   BC
LD    (A16119),BC
LD    (A16117),HL
JP    Z,A9536
LD    (A16121),A
```

A9578:

```
CALL  A6555
```

```
;*****
```

```
;READ command vector table.
```

```
;FNs and variable commands give Illegal Function
Usage Error.
```

```
;
```

```
DW    A9778           ;FP
DW    A9591           ;%
DW    A9607           ;$
DW    A7942           ;FN
DW    A7942           ;command
```

```
;
```

```
;*****
```

```
;READ % variables.
```

```
;
```

A9591:

```
PUSH  DE
CALL  A9817
POP   DE
CALL  A6290
JP    A9805
```

```
;*****
```

A9602:

```
LD    D,H
LD    E,L
JP    A9656
```

```
;*****
```

```

;READ $ variables.
;
A9607:
    PUSH    DE
    PUSH    BC
    LD      HL,(A16119)
    LD      A,(A16121)
    LD      C,A
A9616:
    LD      A,(HL)
    CP      32
    JP      NZ,A9630
    DEC     C
    JP      Z,A9767
    INC     HL
    JP      A9616
;*****
A9630:
    CP      44
    JP      Z,A9602
    CP      34
    JP      Z,A9713
    LD      D,H
    LD      E,L
    LD      (A16143),HL
A9645:
    DEC     C
    JP      Z,A9705
    INC     HL
    LD      A,(HL)
    CP      44
    JP      NZ,A9645
A9656:
    INC     HL
    DEC     C
    LD      (A16119),HL
A9661:
    LD      A,C

```

```
LD    (A16121),A
OR    A
SBC   HL,DE
DEC   L
```

A9669:

```
LD    A,L
POP   HL
CALL  A7264
LD    E,(HL)
INC   HL
LD    D,(HL)
DEC   HL
LD    A,L
LD    (DE),A
INC   DE
LD    A,H
LD    (DE),A
INC   HL
INC   HL
LD    A,(HL)
OR    A
JP    Z,A9701
INC   HL
LD    DE,(A16143)
LD    C,A
LD    B,0
EX    DE,HL
LDIR
```

A9701:

```
POP   DE
JP    A9805
```

;*****

A9705:

```
INC   HL
LD    (A16119),HL
INC   HL
JP    A9661
```

;*****

A9713:

```
DEC    C
JP     Z,A9767
INC    HL
LD     D,H
LD     E,L
LD     (A16143),HL
```

A9723:

```
LD     A,(HL)
CP     34
JP     Z,A9737
DEC    C
JP     Z,A9705
INC    HL
JP     A9723
```

;*****

A9737:

```
PUSH  HL
```

A9738:

```
DEC    C
JP     Z,A9758
INC    HL
LD     A,(HL)
CP     44
JP     Z,A9757
CP     32
JP     Z,A9738
JP     A7948
```

;*****

A9757:

```
DEC    C
```

A9758:

```
INC    HL
LD     (A16119),HL
POP    HL
INC    HL
JP     A9661
```

;*****

```
A9767:
    LD    A,C
    LD    (A16121),A
    LD    (A16119),HL
    LD    L,A
    JP    A9669
```

```
;*****
```

```
;READ FP variables.
```

```
;
```

```
A9778:
    PUSH  DE
    CALL  A9817
    LD    H,B
    LD    L,C
    LD    BC,(A16162)
    LD    (HL),C
    INC   HL
    LD    (HL),B
    INC   HL
    LD    BC,(A16164)
    LD    (HL),C
    INC   HL
    LD    (HL),B
    INC   HL
    LD    A,(A16166)
    LD    (HL),A
    POP   DE
```

```
A9805:
    EXX
    LD    A,C
    EXX
    OR    A
    RET   Z
    EXX
    DEC   C
    EXX
    INC   DE
    JP    A9499
```

;*****

A9817:

```
PUSH BC
LD A, (A16121)
LD C, A
LD DE, (A16119)
```

A9826:

```
LD A, (DE)
CP 32
JP NZ, A9840
DEC C
JP Z, A9957
INC DE
JP A9826
```

;*****

A9840:

```
CP 44
JP Z, A9970
LD A, C
LD (A16121), A
LD (A16119), DE
```

A9853:

```
LD A, (DE)
CP 44
JP Z, A9908
INC DE
DEC C
JP NZ, A9853
LD HL, (A16119)
LD DE, A16247
LD A, (A16121)
LD C, A
LD B, 0
LDIR
LD (A16119), HL
LD A, B
LD (DE), A
LD (A16121), A
```

```
LD DE,A16247
CALL A2576
JP C,A7936
POP BC
```

A9896:

```
LD A,(DE)
OR A
RET Z
INC DE
CP 32
JP Z,A9896
JP A7939
```

;*****

A9908:

```
LD DE,(A16119)
CALL A2576
JP C,A7936
LD H,D
LD L,E
LD DE,(A16119)
LD (A16119),HL
OR A
SBC HL,DE
LD A,(A16121)
SUB L
LD C,A
LD HL,(A16119)
```

A9938:

```
LD A,(HL)
CP 44
JP Z,A9976
CP 32
JP NZ,A7948
DEC C
JP Z,A9978
INC HL
JP A9938
```

;*****

```

A9957:
    LD    A,C
    LD    (A16121),A
    LD    (A16119),DE
    LD    (A16166),A
    POP   BC
    RET
;*****
A9970:
    XOR   A
    LD    (A16166),A
    LD    H,D
    LD    L,E
A9976:
    INC   HL
    DEC   C
A9978:
    LD    (A16119),HL
    LD    A,C
    LD    (A16121),A
    POP   BC
    RET
;*****
;Get memory address (2's complement integer) from
crunch code into HL.
;
A9987:
    CALL  A5939
A9990:
    LD    A,(A16166)
    OR    A
    JR    Z,A10038
    CP    128
    JR    C,A10038
    CP    145
    JP    NC,A7936
    CALL  A10675
    LD    HL,(A16164)

```



```

        LD      A, (A16166)
        BIT    7, H
        JR     NZ, A10031
        SET    7, H
A10020:
        SUB    145
A10022:
        INC    A
        RET    Z
        SRL    H
        RR     L
        JP     A10022
;*****
A10031:
        LD     BC, A2392
        PUSH  BC
        JP     A10020
;*****
A10038:
        LD     HL, 0
        RET
;*****
;CALL address command.
;
A10042:
        CALL  A9987
        CALL  A7299
        PUSH  DE
        EXX
        PUSH  DE
        PUSH  BC
        PUSH  HL
        EXX
        PUSH  IY
        PUSH  IX
A10058:
        CALL  A5938
A10061:

```

```

        POP     IX
        POP     IY
        EXX
        POP     HL
        POP     BC
        POP     DE
        EXX
        POP     DE
        XOR     A
        RET
;*****
;USR command.
;
A10073:
        CALL    A7299
        PUSH   DE
        EXX
        PUSH   DE
        PUSH   BC
        PUSH   HL
        EXX
        PUSH   IY
        PUSH   IX
        LD     HL,(A16130)
        JR     A10058
;*****
;PEEK(address) function.
;
A10091:
        JP     NZ,A7939
        CALL   A9990
A10097:
        LD     A,(HL)
        LD     L,A
        XOR   A
        LD     H,A
        JP     A2407
;*****

```

```

;Parse vector table.
;
A10104:
    DB      1
    DW      A10107
;*****
A10107:
    CALL    A14875
    CALL    A15939
    CALL    A14875
    CALL    A14555
    RET     NC
    JP      A14875
;*****
    NOP
    NOP
    NOP
;*****
;WAIT port,value1,value2 command.
;
A10126:
    CALL    A1500
    PUSH    HL
    EXX
    DEC     C
    EXX
    INC     DE
    CALL    A1500
    EXX
    LD      A,C
    EXX
    OR      A
    JP      Z,A10155
    EXX
    DEC     C
    EXX
    INC     DE
    PUSH    HL

```

```

        CALL    A1500
        LD      A,L
        POP    HL
        LD      H,A
A10155:
        POP    BC
A10156:
        IN  A,(C)
        XOR  L
        AND  H
        JP   Z,A10156
        RET
;*****
;
;
A10164:
        CALL  A9987
        LD   (A16130),HL
        RET
;*****
;
;
A10171:
        CALL  A9987
        LD   (A16766),HL
        RET
;*****
;
;
A10178:
        AND  2
        RET  NZ
        JP  A7915
;*****
;
;
A10184:
        LD   HL,A28080

```

```

        LD     C, (HL)
        INC   C
        JP    A30945
;*****
;FRE or FRE(dummy) function.
;
A10192:
        CALL  A10209
        LD    HL, (A16109)
        LD    BC, (A16111)
        OR    A
        SBC   HL, BC
        OR    A
        JP    A2407
;*****
;String space garbage collection routine.
;
A10209:
        PUSH  DE
        LD    HL, (A16099)
        LD    (A16113), HL
        LD    D, H
        LD    E, L
        LD    HL, (A16115)
        LD    (A16115), DE
A10225:
        LD    BC, (A16111)
        PUSH  HL
        OR    A
        SBC   HL, BC
        POP   HL
        JP    NC, A10301
        LD    E, (HL)
        INC   HL
        LD    D, (HL)
        DEC   HL
        LD    A, (DE)
        CP    L

```

```
JP      NZ,A10288
INC     DE
LD      A,(DE)
CP      H
JP      NZ,A10288
LD      A,(A16114)
LD      (DE),A
DEC     DE
LD      A,(A16113)
LD      (DE),A
LD      E,L
LD      D,H
INC     HL
INC     HL
LD      B,0
LD      A,(HL)
ADD     A,3
JP      NC,A10274
INC     B
```

A10274:

```
LD      C,A
LD      HL,(A16113)
EX      DE,HL
LDIR
LD      (A16113),DE
JP      A10225
```

;*****

A10288:

```
INC     HL
INC     HL
LD      A,(HL)
INC     HL
ADD     A,L
LD      L,A
JP      NC,A10225
INC     H
JP      A10225
```

;*****

```
A10301:
    LD     HL,(A16113)
    LD     (A16111),HL
    POP   DE
    RET
;*****
;VAL(ASCII of number) function.
;
```

```
A10309:
    JP     Z,A7939
    LD     HL,(A16162)
    INC   HL
    INC   HL
    LD     A,(HL)
    OR    A
    JP     Z,A1803
    INC   HL
    LD     C,A
    LD     B,0
    PUSH  DE
    LD     DE,A16247
    LDIR
    XOR   A
    LD     (DE),A
    LD     DE,A16246
```

```
A10337:
    INC   DE
    LD     A,(DE)
    CP    32
    JR    Z,A10337
    CALL  A2576
    POP   DE
    RET   NC
    JP    A7921
```

```
;*****
;ASC(string) function.
;
A10351:
```

```

        JP      Z,A7939
        LD      HL,(A16162)
        INC     HL
        INC     HL
        LD      A,(HL)
        OR      A
        JP      Z,A7936
A10364:
        INC     HL
        LD      L,(HL)
        LD      H,0
        JP      A2407
;*****
;CHR$(n) function.
;
A10371:
        JP      NZ,A7939
        CALL   A2354
        JP      C,A7936
        LD      A,H
        OR      H
        JP      NZ,A7936
        LD      C,L
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A,(DE)
        SUB     184
        JP      NZ,A7948
        INC     A
        LD      HL,A16162
        CALL   A7264
        LD      (A16162),HL
        INC     HL
        INC     HL
        INC     HL
        LD      (HL),C

```



```

        RET
;*****
;STR$(number) function.
;
A10411:
        JP      NZ,A7939
        EXX
        DEC     C
        EXX
        INC     DE
        LD      A,(DE)
        CP      184
        JP      NZ,A7948
        PUSH   DE
        CALL   A3260
A10428:
        LD      A,(A16246)
        LD      HL,A16162
        CALL   A7264
        LD      (A16162),HL
        INC     HL
        INC     HL
        LD      C,(HL)
        INC     HL
        LD      DE,A16247
        EX      DE,HL
        LD      B,0
        LDIR
        POP     DE
        RET
;*****
;LEN(string) function.
;
A10454:
        JP      Z,A7939
        LD      HL,(A16162)
        INC     HL
        JP      A10364

```

;*****

;Check string length. Used by LEFT\$, RIGHT\$, and
MID\$ functions.

;

A10464:

```
JP      Z,A7939
LD      HL,A16162
LD      C,(HL)
INC     HL
LD      B,(HL)
POP     HL
PUSH    BC
PUSH    HL
LD      HL,2
ADD     HL,SP
LD      A,L
LD      (BC),A
INC     BC
LD      A,H
LD      (BC),A
EXX
DEC     C
EXX
INC     DE
CALL    A1500
LD      C,L
OR      C
JP      Z,A7936
LD      A,C
POP     BC
POP     HL
PUSH    HL
PUSH    BC
LD      C,A
INC     HL
INC     HL
LD      A,(HL)
CP      C
```

```

        RET
;*****
;LEFT$(string,count) function.
;
A10508:
        CALL  A10464
        JP    NC,A10515
A10514:
        LD    C,A
A10515:
        DEC   C
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        SUB   184
        JP    Z,A10616
        JP    A7948
;*****
;RIGHT$(string,count) function.
;
A10529:
        CALL  A10464
        JP    C,A10514
        LD    H,A
        SUB   C
        LD    C,A
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        CP    184
        JP    NZ,A7948
        LD    A,C
        LD    C,H
        JP    A10615

```

;*****

A10553:

POP AF
JP C,A10658
LD B,A

A10558:

LD A,C
LD C,B
JP A10614

;*****

;MID\$(string,position,length) function.

;

A10563:

CALL A10464
PUSH AF
EXX
DEC C
EXX
INC DE
LD A,(DE)
CP 184
JP Z,A10553
PUSH BC
CALL A1500
POP BC
LD B,L
EXX
DEC C
EXX
INC DE
LD A,(DE)
CP 184
JP NZ,A7948
POP AF
JP C,A10658
PUSH AF
CP B
JP C,A10553

```

SUB    B
CP     C
JP     C,A10553
POP    AF
LD     A,C
ADD    A,B
LD     B,C
LD     C,A
LD     A,B
DEC    C
A10614:
DEC    A
A10615:
DEC    C
;*****
;Cut string.  On entry, A=left boundary, C=right
boundary.
;
A10616:
LD     B,A
LD     HL,A16162
LD     A,C
SUB    B
INC    A
JR     Z,A10662
CALL  A7264
LD     (A16162),HL
INC    HL
INC    HL
LD     C,(HL)
INC    HL
EX     DE,HL
EX     (SP),HL
PUSH  HL
INC    HL
INC    HL
INC    HL
LD     A,B

```

```

        LD     B,0
        ADD   A,L
        LD     L,A
        JR    NC,A10649
        INC   H
A10649:
        LDIR
        POP   HL
        POP   DE
A10653:
        XOR   A
        LD    (HL),A
        INC   HL
        LD    (HL),A
        RET
;*****
A10658:
        XOR   A
        LD    HL,A16162
A10662:
        CALL  A7264
        LD    (A16162),HL
        POP   HL
        JP    A10653
;*****
;INT(number) function.
;
A10672:
        JP    NZ,A7939
A10675:
        LD    A,(A16166)
        OR    A
        RET   Z
        CP    160
        RET   NC
        LD    BC,(A16164)
        BIT   7,B
        SET   7,B

```

```
JR      Z,A10709
LD      HL,(A16162)
PUSH   HL
LD      HL,(A16164)
PUSH   HL
LD      HL,(A16165)
PUSH   HL
LD      HL,A10771
PUSH   HL
```

A10709:

```
CP      129
JP      C,A1802
CP      144
JP      C,A10760
LD      HL,(A16162)
```

A10722:

```
SRL    B
RR     C
RR     H
RR     L
INC    A
CP     160
JP     NZ,A10722
PUSH  HL
```

A10737:

```
SLA    L
RL     H
RL     C
RL     B
JP     P,A10737
RES    7,B
LD     (A16164),BC
LD     (A16162),HL
POP    HL
LD     A,L
RET
```

;*****

A10760:

```

        CALL    A2354
        PUSH   HL
        OR     A
        CALL   A2407
        POP    HL
        LD     A,L
        RET
;*****
A10771:
        LD     HL,A16165
        SET   7,(HL)
        POP   HL
        LD    (A16174),HL
        POP   HL
        LD    (A16173),HL
        POP   HL
        LD    (A16171),HL
        PUSH  AF
        CALL  A2488
        JR    Z,A10812
        LD    HL,33152                ;not an address
                                        ;need to figure out

what it is
        LD    (A16174),HL
        LD    HL,0
        LD    (A16171),HL
        LD    (A16172),HL
        CALL  A1583
A10812:
        POP   AF
        RET
;*****
;ERRNUM or ERRNUM(dummy) function.
;
A10814:
        LD    A,(A16128)
        LD    L,A
A10818:

```



```

        LD      H,0
A10820:
        OR      A
        JP      A2407
;*****
;Parse vector table.
;
A10824:
        DB      2
        DW      A1117 ;
        DW      A14875 ;
;*****
        DB      0,103,9 ;garbage leftovers
;*****
;SPEED=value command.
;
A10832:
        EXX
        DEC     C
        EXX
        INC     DE
        CALL    A1500
        LD      A,L
        LD      (A16129),A
        RET
;*****
;POS or POS(dummy) function.
;
A10844:
        JP      NZ,A7948
A10847:
        CALL    A26177
        LD      L,A
        LD      H,0
        OR      A
        JP      A2407
;*****
;VPOS or VPOS(dummy) function.

```

```

;
A10857:
        JP      NZ,A7948
A10860:
        CALL   A26184
        LD     L,A
        LD     H,0
        OR     A
        JP     A2407
;*****
;LOMEM: address command.
;
A10870:
        EXX
        DEC    C
        EXX
        INC    DE
A10874:
        CALL   A9987
        PUSH  HL
        CALL   A8144
        LD     HL,(A16095)
        POP   BC
        OR     A
        SBC   HL,BC
        RET   Z
        PUSH  DE
        JP    NC,A10949
        LD     HL,(A16099)
        LD     DE,(A16095)
        OR     A
        SBC   HL,DE
        LD     D,H
        LD     E,L
        ADD   HL,BC
        LD     B,H
        LD     C,L
        LD     HL,(A16109)

```

```
DEC    HL
OR     A
SBC    HL,BC
JP     C,A7927
LD     A,D
LD     D,B
LD     B,A
LD     A,E
LD     E,C
LD     C,A
LD     HL,(A16099)
EX     DE,HL
LD     (A16099),HL
LD     (A16115),HL
LD     (A16111),HL
EX     DE,HL
INC    BC
LDDR
INC    DE
LD     (A16095),DE
JP     A10992
```

;*****

A10949:

```
LD     HL,LOMEM
DEC    HL
OR     A
SBC    HL,BC
JP     NC,A7927
LD     D,B
LD     E,C
LD     HL,(A16099)
LD     BC,(A16095)
OR     A
SBC    HL,BC
LD     B,H
LD     C,L
LD     HL,(A16095)
LD     (A16095),DE
```

```

        LDIR
        EX      DE,HL
        LD      (A16099),HL
        LD      (A16111),HL
        LD      (A16115),HL
A10992:
        LD      HL,(A16107)
        LD      D,H
        LD      E,L
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,DE
        LD      DE,(A16095)
        ADD     HL,DE
        LD      (A16097),HL
        POP     DE
        RET
;*****
;HIMEM: address command.
;
A11010:
        EXX
        DEC     C
        EXX
        INC     DE
A11014:
        CALL    A9987
        PUSH   HL
        CALL    A8144
        POP    HL
        LD     BC,(A16089)
        PUSH   HL
        OR     A
        SBC   HL,BC
        POP    HL
        JP     NC,A7927
        LD     BC,(A16111)
        PUSH   HL

```

```

        OR      A
        SBC    HL,BC
        POP    HL
        JP     C,A7927
        LD     (A16109),HL
        RET

;*****
;FLASH command.
;
A11050:
        LD     HL,A35107
        JR     A11093
;*****
;INVERSE command.
;
A11055:
        LD     HL,A35087
        JR     A11093
;*****
;NORMAL command.
;
A11060:
        LD     HL,A35127
        JR     A11093
;*****
;TEXT command.
;
A11065:
        LD     HL,A18453
        JR     A11093
;*****
;GR command.
;
A11070:
        LD     HL,A18492
        JR     A11093
;*****
;HGR command.

```

```

;
A11075:
        LD      HL,A25484
        JR      A11093
;*****
;HGR2 command.
;
A11080:
        LD      HL,A25370
        JR      A11093
;*****
;SHLOAD command.
;
A11085:
        LD      HL,A6345           ;vector to RET
        JR      A11093
;*****
;HOME command.
;
A11090:
        LD      HL,A19304
A11093:
        PUSH   DE
        CALL   A5938
        POP    DE
        RET
;*****
;COLOR=value command.
;
A11099:
        CALL   A11200
        LD     (A16776),A
        RET
;*****
;HCOLOR=value command.
;
A11106:
        CALL   A11200

```

```

        LD      (A16777),A
        RET
;*****
;COLOR(2)=value command.
;Sets screen border color.
;
A11113:
        CALL   A11200
        LD      (A17059),A
        LD      (A18607),A
        LD      (A25431),A
        RET
;*****
;COLOR(3)=value command.
;Sets NORMAL text (foreground) color.
;
A11126:
        CALL   A11200           ;get value 0-15 in A and L
        CALL   A11209           ;move to hi nibble
        LD     L,A              ;save in L
        LD     A,(A17115)       ;get current NORMAL colors
        AND    15              ;wipe old hi nibble
A11138:
        OR     L                ;swap in new value
        LD     (A17115),A       ;save it
        LD     (A18711),A
        LD     (A25568),A
        RET
;*****
;COLOR(4)=value command.
;Sets NORMAL background color.
;
A11149:
        CALL   A11200           ;get value 0-15 in A and L
        LD     A,(A17115)       ;get current NORMAL colors
        AND    240             ;wipe old lo nibble
        JR     A11138          ;swap in new value, save and
exit

```

```

;*****
;COLOR(5)=value command.
;Sets INVERSE text (foreground) color.
;
A11159:
    CALL    A11200        ;get value 0-15 in A and L
    CALL    A11209        ;move to hi nibble
    LD      L,A           ;save it
    LD      A,(A17126)    ;get current INVERSE colors
    AND     15            ;wipe old hi nibble
A11171:
    OR      L             ;swap in new value
    LD      (A17126),A    ;save it
    RET
;*****
;COLOR(6)=value command.
;Sets INVERSE background color.
;
A11176:
    CALL    A11200        ;get value 0-15 in A and L
    LD      A,(A17126)    ;get current INVERSE colors
    AND     240           ;wipe old lo nibble
    JR      A11171        ;swap in new value, save and
exit
;*****
;COLOR(7)=value command.
;Sets HGR/HGR2 graphics screen color.
;
A11186:
    CALL    A11200        ;get value 0-15 in A and L
    CALL    A11209        ;move to hi nibble
    OR      L             ;make lo and hi nibbles
identical
    LD      (A18633),A    ;save it
    LD      (A25471),A
    RET
;*****
;Get COLOR argument 0-15 in A and L.

```



```

;
A11200:
        CALL    A11214        ;get argument 0-255 in A and
L
        CP      16
        RET     C
        JP      A7936
;*****
;Move lo nibble of A to hi nibble.  Multiplies A*16.
;
A11209:
        ADD     A,A
        ADD     A,A
        ADD     A,A
        ADD     A,A
        RET
;*****
;Skip token and get argument 0-255 in A and L.
;
A11214:
        EXX
        DEC     C
        EXX
        INC     DE
        CALL    A1500
        LD      A,L
        RET
;*****
;PLOT command.
;
A11223:
        CALL    A18728        ;GR coordinate checker
        LD      B,L
        PUSH    DE
        CALL    A19102
        POP     DE
        RET
;*****

```

```

;HLIN command.
;
A11233:
    CALL    A18753        ;VLIN/HLIN coordinate
checker
    PUSH    DE
    LD      E,A
    CALL    A18805
    POP     DE
    RET
;*****
;VLIN command.
;
A11243:
    CALL    A18753        ;VLIN/HLIN coordinate
checker
    PUSH    DE
    LD      E,A
    CALL    A18940
    POP     DE
    RET
;*****
;PROMPT=value command.
;
A11253:
    CALL    A11214        ;get value 0-255 in A and L
    LD      (A1146),A    ;save it
    RET
;*****
;BEEP command.
;
A11260:
    LD      A,7
    LD      HL,A17242
    JP     A11093
;*****
;SCRN(x,y) function.
;

```

A11268:

```
CALL  A2354
JP    C,A7936
LD    A,H
OR    A
JP    NZ,A7936
LD    A,39
CP    L
JP    C,A7936
PUSH  HL
EXX
DEC   C
EXX
INC   DE
LD    A,(DE)
CP    185
JP    NZ,A7948
CALL  A1500
LD    A,39
CP    L
JP    C,A7936
POP   BC
LD    B,L
POP   HL
EX    DE,HL
EX    (SP),HL
PUSH  DE
CALL  A19195
LD    L,A
XOR   A
LD    H,A
JP    A2407
```

;*****

;HTAB value command.

;

A11320:

```
CALL  A1500
LD    C,L
```

```

        PUSH    DE
        CALL    A26191
        POP     DE
        RET
;*****
;VTAB value command.
;
A11330:
        CALL    A1500
        LD      A,L
        OR      A
        JP      Z,A7936
        LD      A,24
        CP      L
        JP      C,A7936
        LD      C,L
        PUSH    DE
        CALL    A26219
        POP     DE
        RET
;*****
;DRAW shape command.  Entry is at A11358.
;
A11351:
        POP     BC
        PUSH    DE
        CALL    A26317
        POP     DE
        RET
;*****
A11358:
        CALL    A1500
        PUSH    HL
        EXX
        LD      A,C
        EXX
        OR      A
        JR      Z,A11351

```

```
EXX
DEC    C
EXX
INC    DE
CALL   A1500
LD     A,255
CP     L
JP     C,A7936
PUSH   HL
EXX
DEC    C
EXX
INC    DE
CALL   A1500
LD     A,191
CP     L
JP     C,A7936
POP    BC
LD     B,L
POP    HL
PUSH   DE
LD     E,L
CALL   A26588
POP    DE
RET
```

```
;*****
```

```
;XDRAW shape command.  Entry is at A11412.
```

```
;
```

```
A11405:
```

```
POP    BC
PUSH   DE
CALL   A26297
POP    DE
RET
```

```
;*****
```

```
A11412:
```

```
CALL   A1500
PUSH   HL
```

```
EXX
LD    A,C
EXX
OR    A
JR    Z,A11405
EXX
DEC   C
EXX
INC   DE
CALL  A1500
LD    A,255
CP    L
JP    C,A7936
PUSH  HL
EXX
DEC   C
EXX
INC   DE
CALL  A1500
LD    A,191
CP    L
JP    C,A7936
POP   BC
LD    B,L
POP   HL
PUSH  DE
LD    E,L
CALL  A26884
POP   DE
RET
```

```
;*****
```

```
;ROT=value command.
```

```
;
```

```
A11459:
```

```
EXX
DEC   C
EXX
INC   DE
```

```

        CALL  A1500
        LD    C,L
        PUSH  DE
        CALL  A29988      ;patch to save ROT value
        POP   DE
        RET
;*****
;SCALE=value command.
;
A11473:
        EXX
        DEC   C
        EXX
        INC   DE
        CALL  A1500
        LD    C,L
        PUSH  DE
        CALL  A26333
        POP   DE
        RET
;*****
;HPLOT command.
;
A11487:
        EXX
        DEC   C
        EXX
        INC   DE
        LD    A,(DE)
        CP    180
        JP    Z,A11581
        EXX
        INC   C
        EXX
        DEC   DE
        CALL  A1500
        LD    A,255
        CP    L

```

```
JP      C,A7936
PUSH   HL
EXX
DEC     C
EXX
INC     DE
CALL   A1500
LD     A,191
CP     L
JP     C,A7936
POP    BC
LD     B,L
EXX
LD     A,C
EXX
OR     A
JP     Z,A11613
PUSH   BC
EXX
DEC     C
EXX
INC     DE
CALL   A1500
LD     A,255
CP     L
JP     C,A7936
PUSH   HL
EXX
DEC     C
EXX
INC     DE
CALL   A1500
LD     A,191
CP     L
JP     C,A7936
POP    BC
EX     DE,HL
EX     (SP),HL
```



```
LD    D,E
LD    E,C
LD    B,H
LD    C,L
CALL  A25686
```

A11571:

```
POP   DE
EXX
LD    A,C
EXX
OR    A
RET   Z
EXX
DEC   C
EXX
INC   DE
```

A11581:

```
CALL  A1500
LD    A,255
CP    L
JP    C,A7936
PUSH  HL
EXX
DEC   C
EXX
INC   DE
CALL  A1500
LD    A,191
CP    L
JP    C,A7936
POP   BC
LD    B,L
PUSH  DE
CALL  A25797
JP    A11571
```

;*****

A11613:

```
PUSH  DE
```

```

        CALL  A25601
        POP   DE
        RET
;*****
;PDL(n) function.
;
A11619:
        JP    NZ,A7939
        CALL  A2354
        JP    C,A7936
        LD    A,H
        OR    A
        JP    NZ,A7936
        LD    A,L
        CP    16
        JP    NC,A7936
        LD    C,A
        PUSH  DE
        CALL  A26904
        POP   DE
        LD    L,A
        XOR   A
        LD    H,A
        JP    A2407
;*****
;INVERSE fix for TEXT40.
;CPLs the bit patterns for the upper ASCII set
(128-255) and
;saves them back to VRAM.
;
A11651:
        LD    A,(A17988)
        CP    40
        JR    Z,A11662
        LD    HL,0
        RET
;*****
A11662:

```

```

        LD      DE,0          ;VRAM address
                                ;pattern generators for low
ASCII 0-127
        LD      B,128        ;counter
A11667:
        PUSH   BC            ;save it
        PUSH   DE            ;save the VRAM address
        LD     HL,A11718     ;buffer
        LD     BC,8          ;8 bytes to read from VRAM
        CALL  READ_VRAM      ;EOS read VRAM
        LD     HL,A11718     ;buffer
        LD     B,8           ;counter
A11683:
        LD     A,(HL)        ;get the bit pattern
        CPL                    ;swap 1s and 0s
        LD     (HL),A        ;save it back
        INC   HL             ;point to next
        DJNZ  A11683         ;do all 8
        POP   HL             ;restore VRAM address
        PUSH  HL             ;and save it again
        LD   DE,1024         ;offset to pattern
generators for upper ASCII 128-255
        ADD   HL,DE
        EX   DE,HL          ;into DE
        LD   HL,A11718      ;buffer
        LD   BC,8           ;8 bytes to write to VRAM
        CALL WRITE_VRAM     ;EOS write VRAM
        POP  HL             ;restore low ASCII VRAM
address
        LD   DE,8           ;offset to next character
pattern
        ADD  HL,DE
        EX  DE,HL          ;into DE
        POP  BC            ;restore counter
        DJNZ A11667        ;do all 128
        POP  HL            ;clear old RET address
        JP   A17163        ;exit
;*****

```

```

;VRAM buffer for CPL of bitmaps in TEXT40 INVERSE.
;
A11718:
    DS 8
;*****
;SERIAL port,baudrate,stats command.
;
A11726:
    CALL  A1500
    LD    A,L
    PUSH  DE
    LD    B,5
    LD    DE,4
    LD    HL,A30680
A11739:
    CP    (HL)
    JP    Z,A30740
    ADD   HL,DE
    DJNZ  A11739
    LD    B,2
    LD    HL,A33871
    JP    A30731
;*****
;MIB2 patches to SERIAL command.
;
A11754:
    LD    A,(A28074) ;port to initialize
    LD    HL,A33897 ;MIB2 data table
    CP    68 ;is it Eve/Orphanware or
ADAMlink modem?
    RET   C ;NO, it's MIB2
    LD    HL,A28070 ;YES, so use
Eve/Orphanware/ADAMlink data table
    RET
;*****
;More MIB2 patches to SERIAL command.
;
A11767:

```

```

        LD      A, (A28074)    ;port to initialize
        CP      68             ;is it Eve/Orphanware or
ADAMlink modem?
        RET     NC             ;YES, so do
Eve/Orphanware/ADAMlink init
        JP      A35497        ;NO, so do MIB2 init
;*****
A11776:
        INC     DE
A11777:
        LD      A, (DE)
        OR      A
        RET     Z
        CP      33
        JR      C, A11776
        OR      A
        RET
;*****
A11786:
        LD      B, (HL)
        LD      A, (DE)
        LD      C, A
        INC     HL
        INC     DE
A11791:
        LD      A, C
        CP      B
        JP      C, A11797
        LD      C, B
A11797:
        LD      B, 0
        PUSH   AF
        LD      A, C
        OR      A
        JP      Z, A11818
A11805:
        LD      A, (DE)
        CP      (HL)

```

```

        JP      NZ,A11820
A11810:
        INC    DE
        INC    HL
        DEC    BC
        LD     A,B
        OR     C
        JP     NZ,A11805
A11818:
        POP    AF
        RET
;*****
A11820:
        EX     AF,AF'
        JP     C,A11835
        LD     A,(DE)
        XOR    32
        CP     (HL)
        JP     NZ,A11835
        EX     AF,AF'
        JP     A11810
;*****
A11835:
        EX     AF,AF'
        POP    BC
        RET
;*****
A11838:
        DB     230                ;AND 55 ;clear CF
                                   ;I hate overlapping code!
A11839:
        SCF                                ;set CF
        EX     AF,AF'
        PUSH   BC
        PUSH   DE
        PUSH   HL
        CALL   A11786
        POP    HL

```

```

        POP    DE
        POP    BC
        RET
;*****
A11851:
        DB    230           ;AND 55 ;clear CF
                               ;I hate overlapping code!
A11852:
        SCF                               ;set CF
        EX    AF,AF'
        PUSH  BC
        PUSH  DE
        PUSH  HL
        CALL  A11791
        POP   HL
        POP   DE
        POP   BC
        RET
;*****
A11864:
        LD    A,(DE)
A11865:
        BIT   5,A
        JP    NZ,A11877
A11870:
        CP    65
        CCF
        RET   NC
        CP    91
        RET
;*****
A11877:
        CP    97
        CCF
        RET   NC
        CP    123
        RET
;*****

```

```

A11884:
    LD    A, (DE)
A11885:
    CP    48
    CCF
    RET   NC
    CP    58
    RET
;*****
A11892:
    LD    HL, 53632
    LD    (A16089), HL
    LD    HL, 0
    LD    (A16091), HL
    LD    (A16093), HL
A11907:
    LD    HL, 64320    ;loword random number seed
    LD    (A16190), HL ;save it
    LD    HL, 53905    ;hiword random number seed
    LD    (A16192), HL ;save it
    RET
;*****
A11920:
    DB    230          ;AND 55 ;clear CF
                                ;I hate overlapping code!
A11921:
    SCF                ;set CF
    PUSH AF
    CALL A19884
    CALL A27477
    CALL A12110
    EX   DE, HL
    CALL A12128
    LD   DE, A16246
    OR  A
    SBC HL, DE
    LD  B, L
    INC B

```



```

        JR      A11951
;*****
A11946:
        LD      A,32
        CALL   A11994
A11951:
        DJNZ   A11946
        LD      A,94
        CALL   A11994
        CALL   A12128
        POP    AF
        POP    HL
A11963:
        PUSH   AF
        PUSH   HL
        LD      BC,A12485
        OR     A
        SBC    HL,BC
        POP    HL
        LD      B,255
        JP     NZ,A32047
        JP     A32046
;*****
A11980:
        DB     9
        DB     " Expected"
;*****
A11990:
        POP    HL
A11991:
        OR     A
        JR     A11963
;*****
A11994:
        PUSH   AF
        PUSH   BC
        PUSH   DE
        PUSH   HL

```

```

        LD      HL, (A16201)
A12001:
        OR      A
        CALL   NZ, A5938
        POP    HL
        POP    DE
        POP    BC
        POP    AF
A12009:
        RET
;*****
A12010:
        PUSH   AF
        CALL   A32157
        POP    AF
        PUSH   AF
        CP     13
        JR     Z, A12031
        LD     A, (A16177)
        DEC   A
        JR     NZ, A12039
        LD     A, 13
        CALL   A32157
A12031:
        LD     A, 10
        CALL   A32157
        LD     A, (A16176)
A12039:
        LD     (A16177), A
        POP    AF
A12043:
        PUSH   AF
        LD     A, (A16129)
A12047:
        LD     B, A
A12048:
        INC   B
        JR     NZ, A12048

```

```
INC    A
JR     NZ,A12047
POP    AF
JP     A19471
```

```
;*****
```

```
A12058:
```

```
EXX
DEC    C
EXX
INC    DE
CALL   A1500
LD     A,7
CP     L
JP     C,A7936
```

```
A12071:
```

```
ADD    HL,HL
LD     BC,A16213
ADD    HL,BC
LD     A,(HL)
INC    HL
LD     H,(HL)
LD     L,A
LD     (A16201),HL
RET
```

```
;*****
```

```
A12084:
```

```
EXX
DEC    C
EXX
INC    DE
CALL   A1500
LD     A,7
CP     L
JP     C,A7936
```

```
A12097:
```

```
ADD    HL,HL
LD     BC,A16229
ADD    HL,BC
```

```
LD    A, (HL)
INC   HL
LD    H, (HL)
LD    L, A
LD    (A16197), HL
RET
```

```
;*****
```

```
A12110:
```

```
LD    A, (HL)
OR    A
RET   Z
PUSH  BC
PUSH  HL
LD    C, A
```

```
A12116:
```

```
INC   HL
LD    A, (HL)
CALL  A11994
DEC   C
JP    NZ, A12116
POP   HL
POP   BC
RET
```

```
;*****
```

```
A12128:
```

```
PUSH  HL
LD    HL, A1150
CALL  A12110
POP   HL
RET
```

```
;*****
```

```
A12137:
```

```
PUSH  BC
PUSH  HL
PUSH  DE
LD    HL, (A16197)
CALL  A5938
POP   DE
```

```

        POP    HL
        POP    BC
        RET
;*****
A12150:
        CALL   A12128
        LD     HL,A1145
        JP     A12110
;*****
A12159:
        CALL   A29152
        PUSH  AF
        XOR   A
        LD    (A17006),A
        LD    A,(A17008)
        OR   A
        JR   NZ,A12181
        LD    A,(A17004)
        PUSH AF
        XOR   A
        LD    (A17004),A
A12181:
        LD    DE,A16245
        LD    A,254
        LD    (DE),A
        PUSH HL
        PUSH BC
        EX   DE,HL
        PUSH HL
        LD    C,(HL)
A12192:
        INC   HL
        INC   HL
        LD    B,0
A12196:
        CALL   A12137
        CP    32
        JR    C,A12247

```

```

        CP      128
        JR      NC,A12247
A12207:
        LD      (HL),A
        CALL   A12420
A12211:
        INC     B
        INC     HL
        LD      A,B
        CP      C
        JP      C,A12196
        PUSH   AF
A12219:
        POP     AF
        DEC     B
A12221:
        LD      (HL),0
        LD      A,B
        POP     DE
        INC     DE
        LD      (DE),A
        POP     BC
        POP     HL
        CALL   A12435
        LD      A,(A17008)
        OR      A
        JR      NZ,A12242
        POP     AF
        LD      (A17004),A
A12242:
        POP     AF
        JP      A29169
;*****
A12246:
        RET
;*****
A12247:
        PUSH   HL

```

```
PUSH BC
LD HL,A12369
LD BC,17
CPIR
JR NZ,A12289
LD HL,A12386
ADD HL,BC
ADD HL,BC
POP BC
EX AF,AF'
LD A,(HL)
INC HL
LD H,(HL)
LD L,A
EX AF,AF'
EX (SP),HL
RET
```

```
;*****
```

```
A12273:
```

```
LD A,148
JR A12283
```

```
;*****
```

```
A12277:
```

```
LD A,151
JR A12283
```

```
;*****
```

```
A12281:
```

```
SUB 4
```

```
A12283:
```

```
CALL A12420
JP A12196
```

```
;*****
```

```
A12289:
```

```
POP BC
POP HL
LD (HL),A
JP A12207
```

```
;*****
```

A12295:

```
LD    A,B
OR    A
JP    Z,A12196
DEC   B
DEC   HL
LD    A,8
CALL  A12420
JP    A12196
```

;*****

A12310:

```
LD    A,(A16955)
AND   127
LD    (HL),A
LD    A,161
CALL  A12420
JP    A12211
```

;*****

A12324:

```
CALL  A18360
LD    A,D
AND   3
NEG
ADD   A,4
```

A12334:

```
PUSH  AF
LD    (HL),32
INC   B
INC   HL
LD    A,B
CP    C
JP    NC,A12219
LD    A,32
CALL  A12420
POP   AF
DEC   A
JR    NZ,A12334
JP    A12196
```


;*****

A12356:

```
POP    HL
PUSH   HL
LD     A,92
CALL   A12420
CALL   A12435
JP     A12192
```

;*****

;Control character table--screen printing.

;

A12369:

```
DB     13      ;CR
DB     8       ;BS
DB     9       ;HT
DB     161     ;right arrow
DB     163     ;left arrow
DB     148     ;insert (old ^N)
DB     151     ;delete (old ^O)
DB     24      ;^X cancel
DB     160     ;up arrow
DB     162     ;down arrow
DB     128     ;home
DB     12      ;^L clear screen and home
DB     164     ;control up arrow
DB     166     ;control down arrow
DB     167     ;control left arrow
DB     165     ;control right arrow
DB     0       ;^2 null
```

;*****

;Control character vector table--screen printing.

;

A12386:

```
DW     A12196      ;^2 null
DW     A12281      ;control right arrow
DW     A12281      ;control left arrow
DW     A12281      ;control down arrow
DW     A12281      ;control up arrow
```

```
DW      A12283      ;^L clear screen and home
DW      A12283      ;home
DW      A12283      ;down arrow
DW      A12283      ;up arrow
DW      A12356      ;^X cancel
DW      A12277      ;delete (old ^O)
DW      A12273      ;insert (old ^N)
DW      A12295      ;left arrow
DW      A12310      ;right arrow
DW      A12324      ;HT
DW      A12295      ;BS
DW      A12221      ;CR
```

```
;*****
```

```
A12420:
```

```
    PUSH  AF
    PUSH  HL
    PUSH  BC
    PUSH  DE
    LD    HL,(A16203)
    CALL  A5938
    POP   DE
    POP   BC
    POP   HL
    POP   AF
    RET
```

```
;*****
```

```
A12435:
```

```
    PUSH  HL
    PUSH  BC
    LD    HL,A1150
    LD    B,(HL)
```

```
A12441:
```

```
    INC   HL
    LD    A,(HL)
    CALL  A12420
    DJNZ  A12441
    POP   BC
    POP   HL
```

```

        RET
;*****
A12451:
        LD      HL,0
A12454:
        LD      B,0
        LD      A,(DE)
        AND     15
        LD      C,A
        ADD     HL,BC
        JR      C,A12482
        INC     DE
        CALL    A11884
        RET     NC
        ADD     HL,HL
        JR      C,A12482
        LD      B,H
        LD      C,L
        ADD     HL,HL
        JR      C,A12482
        ADD     HL,HL
        JR      C,A12482
        ADD     HL,BC
        JR      NC,A12454
A12482:
        CALL    A11920
;*****
A12485:
        DB      14
        DB      "Number Too Big"
;*****
A12500:
        PUSH    BC
        LD      H,D
        LD      L,E
        LD      B,1
        JR      A12517
;*****

```

A12507:

```
INC    A
LD     C,A
ADD    A,B
LD     B,A
LD     A,L
ADD    A,C
LD     L,A
JR     NC,A12517
INC    H
```

A12517:

```
LD     A,(HL)
OR     A
JP     NZ,A12507
LD     A,B
LD     (A16206),A
POP    BC
RET
```

;*****

A12528:

```
PUSH  DE
LD     BC,0
LD     HL,(A16093)
EX     DE,HL
```

A12536:

```
LD     H,D
LD     L,E
PUSH  HL
OR     A
SBC   HL,BC
POP    HL
JP     C,A12594
JP     Z,A12594
ADD   HL,BC
RR     H
LD     A,L
RRA
AND   252
```

```

        LD     L,A
        PUSH  HL
        PUSH  DE
        EX    DE,HL
        LD    HL,(A16089)
        ADD   HL,DE
A12564:
        LD    E,(HL)
        INC  HL
        LD    D,(HL)
        LD    HL,(A16207)
        PUSH HL
        OR   A
        SBC  HL,DE
        POP  HL
        POP  DE
        JP   C,A12590
        JP   Z,A12601
        POP  BC
        INC  BC
        INC  BC
        INC  BC
        INC  BC
        JP   A12536
;*****
A12590:
        POP  DE
        JP   A12536
;*****
A12594:
        LD    HL,(A16089)
        ADD   HL,BC
        OR   A
        POP  DE
        RET
;*****
A12601:
        LD    HL,(A16089)

```

```
POP DE
ADD HL,DE
SCF
POP DE
RET
```

```
;*****
```

```
A12609:
```

```
PUSH DE
CALL A12786
POP DE
CALL A12528
RET
```

```
;*****
```

```
A12618:
```

```
EXX
SET 2,B
SET 3,B
EXX
PUSH IX
PUSH DE
CALL A12528
CALL C,A12609
LD B,H
LD C,L
LD DE,(A16093)
LD HL,(A16089)
ADD HL,DE
SBC HL,BC
PUSH HL
LD H,B
LD L,C
LD DE,(A16089)
SBC HL,DE
PUSH HL
LD BC,4
LD A,(A16205)
ADD A,C
LD C,A
```

```
RL      B
LD      HL, (A16089)
LD      D, H
LD      E, L
SBC     HL, BC
LD      BC, (A16099)
INC     BC
PUSH    HL
OR      A
SBC     HL, BC
POP     HL
JP      C, A12768
LD      (A16089), HL
EX      DE, HL
POP     BC
LD      A, B
OR      C
JP      Z, A12697
LDIR
```

A12697:

```
PUSH    DE
POP     IX
LD      A, (A16207)
LD      (DE), A
INC     DE
LD      A, (A16208)
LD      (DE), A
LD      C, 3
EX      DE, HL
ADD     HL, BC
EX      DE, HL
POP     BC
LD      A, B
OR      C
JP      Z, A12722
LDIR
```

A12722:

```
LD      (IX+2), E
```

```

LD      (IX+3),D
LD      A,(A16205)
LD      C,A
LD      B,0
POP     HL
LDIR
LD      HL,(A16091)
INC     HL
LD      (A16091),HL
ADD     HL,HL
ADD     HL,HL
LD      (A16093),HL
LD      HL,(A16099)
LD      (A16111),HL
LD      (A16115),HL
LD      HL,(A16089)
DEC     HL
LD      (A16109),HL
POP     IX
RET
;*****
A12768:
POP     HL
POP     HL
POP     HL
POP     IX
LD      HL,A1235
JP      A11991
;*****
A12779:
LD      (A16207),HL
A12782:
CALL    A12528
RET     NC
A12786:
EXX
SET     2,B
SET     3,B

```



```
EXX
PUSH IX
PUSH IY
INC HL
INC HL
LD E, (HL)
INC HL
LD D, (HL)
PUSH DE
CALL A12500
LD BC, (A16089)
LD HL, (A16093)
ADD HL, BC
LD B, D
LD C, E
EX DE, HL
SBC HL, DE
PUSH HL
LD H, B
LD L, C
LD D, B
LD E, C
DEC DE
DEC A
LD B, 0
LD C, A
ADD HL, BC
EX DE, HL
POP BC
LD A, B
OR C
JP Z, A12838
LDDR
```

A12838:

```
PUSH DE
PUSH HL
POP IX
POP IY
```

```
POP    HL
LD     BC, (A16091)
LD     DE, -3
ADD    IX, DE
ADD    IY, DE
```

A12856:

```
LD     A, B
OR     C
JP     Z, A12927
LD     D, (IX+3)
LD     E, (IX+2)
EX     DE, HL
PUSH   HL
OR     A
SBC    HL, DE
POP    HL
EX     DE, HL
JP     C, A12889
JP     NZ, A12898
LD     DE, -4
ADD    IX, DE
DEC    BC
JP     A12856
```

;*****

A12889:

```
LD     A, (A16206)
ADD    A, E
LD     E, A
JP     NC, A12898
INC    D
```

A12898:

```
LD     (IY+3), D
LD     (IY+2), E
LD     A, (IX+1)
LD     (IY+1), A
LD     A, (IX+0)
LD     (IY+0), A
LD     DE, -4
```

```

        ADD     IX,DE
        ADD     IY,DE
        DEC     BC
        JP      A12856
;*****
A12927:
        LD      DE,4
        ADD     IY,DE
        LD      (A16089),IY
        DEC     IY
        LD      (A16109),IY
        LD      HL,(A16091)
        DEC     HL
        LD      (A16091),HL
        ADD     HL,HL
        ADD     HL,HL
        LD      (A16093),HL
        POP     IY
        POP     IX
        RET
;*****
A12959:
        LD      A,(A16148)           ;does nothing to CF
                                       ;I hate overlapping
code!

A12961      EQU      A12959+2

;;A12961:
;;      CCF           ;complement CF (0-->1
or 1-->0)

        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        DB      254           ;CP 175 ;leaves A
alone

```

```

code!
A12967:
    XOR A
A12968:
    LD     (A16209),A
    PUSH  BC
    PUSH  HL
    LD     HL,A262
    LD     A,6
    OR     A
    JR     A13012
;*****
A12981:
    POP   AF
    DEC   A
    JR    Z,A13034
    JR    C,A13001
    INC   E
    DEC   E
    JR    NZ,A13001
    PUSH AF
    CP    1
    JR    Z,A13003
    LD    A,(A16209)
    JR    A13006
;*****
A13001:
    SCF
    PUSH AF
A13003:
    LD    A,E
    OR    48
A13006:
    CALL A11994
    POP  AF
    EX  DE,HL
    POP HL

```

;I hate overlapping

;A=0 in this frame

```

A13012:
    LD     C, (HL)
    INC   HL
    LD     B, (HL)
    INC   HL
    PUSH  HL
    EX    DE, HL
    PUSH  AF
    LD     E, 0

A13021:
    LD     A, H
    CP    B
    JR    NZ, A13027
    LD     A, L
    CP    C

A13027:
    JR    C, A12981
    SBC  HL, BC
    INC  E
    JR    A13021
;*****
A13034:
    POP  HL
    POP  HL
    POP  BC
    RET
;*****
A13038:
    LD     A, (A16148)
    OR    A
    JP    NZ, A13056
    LD     A, (DE)
    CP    7
    JR    NZ, A13056
    DEC  C
    LD     A, 63
    JP    A11994
;*****

```

A13056:
DEC C
LD A, (DE)
CP 1
JR Z, A13069
LD A, (A16148)
CALL A11994
LD A, (DE)

A13069:
CALL A13379
JP A12110

;*****

A13075:
DEC C
INC DE
LD A, (DE)
LD B, A
AND 224
RL A
JP C, A13092

A13086:
LD HL, A1064
JP A11991

;*****

A13092:
JP NZ, A13312
LD A, B
CP 138
JP C, A13134
INC DE
JP Z, A13142
CP 139
JP Z, A13153
CP 144
JP Z, A13273
CP 145
JP C, A13164
JP Z, A13285

```

        CP      146
        JP      Z,A13477
        LD      HL,A1064
        CALL   A11991
A13134:
        AND    15
        OR     48
        CALL   A11994
        RET
;*****
A13142:
        LD     A,(DE)
        PUSH  DE
        LD     D,0
        LD     E,A
        CALL  A12967
        POP   DE
        DEC   C
        RET
;*****
A13153:
        EX    DE,HL
        LD    E,(HL)
        INC  HL
        LD    D,(HL)
        CALL A12967
        EX    DE,HL
        DEC  C
        DEC  C
        RET
;*****
A13164:
        LD    A,B
        AND  3
        LD    H,A
        LD    A,(DE)
        LD    L,A
        PUSH DE

```

```
LD      D,H
LD      E,L
ADD     HL,HL
ADD     HL,HL
ADD     HL,DE
LD      DE,(A16095)
ADD     HL,DE
LD      A,(HL)
LD      D,A
LD      A,D
AND     64
CALL    NZ,A13264
LD      A,D
AND     3
JR      Z,A13244
LD      B,A
INC     HL
INC     HL
```

A13197:

```
INC     HL
LD      A,(HL)
CALL    A11994
DJNZ   A13197
```

A13204:

```
LD      B,D
POP     DE
INC     DE
LD      A,(DE)
DEC     C
OR      A
JR      Z,A13223
LD      H,B
LD      B,A
```

A13214:

```
INC     DE
DEC     C
LD      A,(DE)
CALL    A11994
```



```

        DJNZ  A13214
        LD    B,H
A13223:
        DEC  C
        LD   A,B
        AND  48
        CP   32
        JR   NZ,A13236
        LD   A,36
        JP   A11994
;*****
A13236:
        CP   16
        RET  NZ
        LD   A,37
        JP   A11994
;*****
A13244:
        INC  HL
        INC  HL
        INC  HL
        LD   A,(HL)
        INC  HL
        LD   H,(HL)
        LD   L,A
        LD   A,C
        LD   BC,(A16097)
        ADD  HL,BC
        LD   C,A
        CALL A12110
        JP   A13204
;*****
A13264:
        PUSH HL
        LD   HL,A15151
        CALL A12110
        POP  HL
        RET

```

;*****

A13273:

```
EX    DE,HL
CALL  A12110
LD    B,0
LD    C,(HL)
ADD   HL,BC
EX    DE,HL
LD    C,0
RET
```

;*****

A13285:

```
EX    DE,HL
LD    A,34
CALL  A11994
LD    B,(HL)
DEC   C
LD    A,C
SUB   B
LD    C,A
CALL  A12110
LD    A,34
CALL  A11994
LD    A,C
LD    C,B
LD    B,0
ADD   HL,BC
LD    C,A
EX    DE,HL
RET
```

;*****

A13312:

```
LD    A,B
CALL  A13361
LD    A,32
PUSH  AF
CALL  A31197
LD    A,B
```

```
CALL A13389
CALL A12110
POP AF
CALL A1092
LD A,B
CP 178
RET NZ
NOP
NOP
INC DE
JP A13413
```

```
;*****
```

```
A13343:
```

```
CP 185
JP NZ,A13353
LD A,32
JP A11994
```

```
;*****
```

```
A13353:
```

```
CP 186
```

```
A13355:
```

```
RET NZ
LD A,32
JP A11994
```

```
;*****
```

```
A13361:
```

```
LD A,(A16148)
OR A
LD A,B
JR Z,A13375
CP 171
```

```
A13370:
```

```
CCF
RET NC
CP 181
RET
```

```
;*****
```

```
A13375:
```

```

        CP      171
        JR      A13370
;*****
A13379:
        PUSH   BC
        CALL   A27421
        LD     BC,3
        JP     A13396
;*****
A13389:
        PUSH   BC
        CALL   A27431
        LD     BC,1
A13396:
        PUSH   DE
        LD     D,0
A13399:
        CP     (HL)
        ADD    HL,BC
        JP     Z,A13410
        LD     E,(HL)
        INC    HL
        ADD    HL,DE
        JP     A13399
;*****
A13410:
        POP    DE
        POP    BC
        RET
;*****
A13413:
        LD     A,(DE)
        LD     C,A
        INC    DE
        CALL   A13038
        INC    C
        DEC    C
        RET    Z

```

```
LD    A, (DE)
CALL  A30499
NOP
```

A13427:

```
CALL  A13075
LD    A, C
OR    A
JP    NZ, A13427
RET
```

;*****

A13436:

```
LD    A, (DE)
LD    H, 0
LD    L, A
ADD   HL, DE
INC   HL
PUSH  HL
CALL  A13413
POP   DE
LD    A, (DE)
OR    A
RET   Z
LD    HL, A1148
CALL  A12110
JP    A13436
```

;*****

A13459:

```
PUSH  BC
CALL  A1485
LD    E, (HL)
INC   HL
LD    D, (HL)
INC   HL
PUSH  HL
CALL  A13436
POP   HL
CALL  A12128
POP   BC
```

```

RET
;*****
A13477:
LD    A, (DE)
LD    (A16162), A
INC   DE
LD    A, (DE)
LD    (A16163), A
INC   DE
LD    A, (DE)
LD    (A16164), A
INC   DE
LD    A, (DE)
LD    (A16165), A
INC   DE
LD    A, (DE)
LD    (A16166), A
PUSH  DE
PUSH  BC
CALL  A1553
POP   BC
POP   DE
LD    A, C
SUB   5
LD    C, A
RET
;*****
A13513:
LD    HL, (A16115)
LD    C, (HL)
INC   HL
LD    B, (HL)
DEC   HL
LD    A, (BC)
CP    L
JR    NZ, A13582
INC   BC
LD    A, (BC)

```

```
CP      H
JR      NZ,A13582
DEC     BC
INC     HL
INC     HL
LD      A,(HL)
LD      H,B
LD      L,C
CALL   A7264
LD      C,(HL)
INC     HL
LD      B,(HL)
DEC     HL
LD      A,L
LD      (BC),A
INC     BC
LD      A,H
LD      (BC),A
INC     HL
INC     HL
LD      A,(HL)
OR      A
JR      Z,A13572
INC     HL
PUSH   DE
EX      DE,HL
LD      HL,(A16115)
INC     HL
INC     HL
INC     HL
LD      C,A
LD      B,0
LDIR
LD      (A16115),HL
POP    DE
RET
```

```
;*****
```

```
A13572:
```

```

        LD     HL, (A16115)
        INC   HL
        INC   HL
A13577:
        INC   HL
A13578:
        LD     (A16115), HL
        RET
;*****
A13582:
        INC   HL
        INC   HL
        LD     A, (HL)
        INC   HL
        ADD   A, L
        LD     L, A
        JR    NC, A13578
        JR    A13577
;*****
A13592:
        PUSH  HL
        PUSH  BC
        PUSH  AF
        LD    L, A
        LD    H, 0
        CALL  A7311
        LD    HL, (A16115)
        LD    A, (A16111)
        CP    L
        JR    NZ, A13616
        LD    A, (A16112)
        CP    H
        JR    Z, A13645
A13616:
        EXX
        BIT   2, B
        EXX
        JR    NZ, A13645

```



```

A13622:
    INC    HL
    INC    HL
    LD     C, (HL)
    POP    AF
    SUB    3
    JR     C, A13631
    SUB    C
A13631:
    PUSH   AF
    CALL   A13513
    POP    AF
    JR     C, A13656
    PUSH   AF
    LD     HL, (A16115)
    JP     A13622
;*****
A13645:
    POP    AF
    LD     C, A
    LD     B, 0
    ADD    HL, BC
    LD     (A16115), HL
    LD     (A16111), HL
A13656:
    POP    BC
    POP    HL
    RET
;*****
A13659:
    CALL   A11777
    RET    Z
    PUSH   DE
    LD     BC, 1
    CALL   A11864
    JP     C, A13745
    INC    B
    CALL   A11884

```

```

        JP      C,A13734
        INC    B
        CP     58
        JR     Z,A13731
        AND    252
        CP     60
        JR     NZ,A13730
        LD     A,(DE)
        INC    A
        AND    3
        JR     Z,A13761
        INC    DE
        BIT    0,A
        JR     Z,A13720
        EX     DE,HL
        XOR    63
        CP     (HL)
        JP     Z,A13715
        LD     A,61
        CP     (HL)
        JP     NZ,A13729
A13715:
        EX     DE,HL
A13716:
        INC    C
        POP    DE
        SCF
        RET
;*****
A13720:
        LD     A,(DE)
        AND    253
        CP     60
        JP     Z,A13716
A13728:
        DB     62                ;LD A,235 ;in this
frame
A13729:

```

```

        DB      235          ;EX DE,HL  ;in this
frame
A13730:
        SCF          ;clear CF in this
frame
                                ;I hate overlapping
code!
A13731:
        POP      DE
        RET
;*****
A13733:
        INC      C
A13734:
        INC      DE
        CALL    A11884
        JP      C,A13733
        POP      DE
        SCF
        RET
;*****
A13744:
        INC      C
A13745:
        INC      DE
        CALL    A11864
        JP      C,A13744
        CALL    A11884
        JP      C,A13744
        POP      DE
        SCF
        RET
;*****
A13761:
        LD      B,0
        POP      DE
        SCF
        RET

```

;*****

A13766:

LD A,3
LD HL,A272
JP A13779

;*****

A13774:

LD A,1
LD HL,A818

A13779:

PUSH BC
PUSH DE
LD D,0
LD E,A
LD B,C

A13785:

LD A,(HL)
OR A
JP Z,A27441
ADD HL,DE
LD C,(HL)
INC HL
EX DE,HL
EX (SP),HL
CALL A11851
EX (SP),HL
EX DE,HL
JP NZ,A13812

A13803:

DEC HL
OR A
SBC HL,DE
SCF
LD A,(HL)
POP DE
POP BC
RET

;*****

```

A13812:
    LD     A,C
    ADD    A,L
    LD     L,A
    JP     NC,A13785
    INC    H
    JP     A13785
;*****
A13822:
    POP    DE
    POP    BC
    XOR    A
    RET
;*****
A13826:
    CALL   A13766
    RET    C
    JP     A13774
;*****
A13833:
    LD     HL,0
    LD     (A16105),HL
    LD     HL,A16503
    LD     (A16103),HL
A13845:
    CALL   A13946
    CALL   A13969
    JP     C,A13845
    JP     Z,A13900
    CALL   A11920
;*****
A13860:
    DB     39
    DB     "Meaning Of Line Unclear"
    DB     13      ;CR
    DB     " ':' Expected ?"
;*****
A13900:

```

```

        XOR    A
        CALL  A14530
        LD    HL,(A16103)
        LD    BC,A16503
        OR    A
        SBC  HL,BC
        LD    A,H
        OR    A
        JP    Z,A13935
A13918:
        CALL  A11920
;*****
        DB    13
        DB    "Line Too Long"
;*****
A13935:
        LD    A,L
        LD    (A16205),A
        CALL  A14483
        LD    DE,A16503
        RET
;*****
A13946:
        PUSH  BC
        LD    BC,(A16103)
        XOR  A
        CALL  A14530
        CALL  A13992
        LD    HL,(A16103)
        OR    A
        SBC  HL,BC
        LD    A,L
        DEC  A
        LD    (BC),A
        POP  BC
        RET
;*****
A13969:

```

```

        CALL  A11777
        RET   Z
        CP    58
        JP    Z,A13980
        OR    A
        RET
;*****
A13980:
        INC   DE
        CALL  A11777
        RET   Z
        CP    58
        JP    Z,A13980
        SCF
        RET
;*****
A13992:
        PUSH  BC
        CALL  A13659
        JP    NC,A14080
        LD    A,B
        OR    A
        JP    Z,A14028
        LD    A,(DE)
        CP    38
        JR    Z,A14028
A14009:
        CALL  A11920
;*****
A14012:
        DB    15
        DB    "Illegal Command"
;*****
A14028:
        CALL  A13766
        JP    NC,A14048
        CALL  A14530
        CALL  A14549

```

```

        INC     HL
        LD      A, (HL)
        INC     HL
        LD      H, (HL)
        LD      L, A
        JP      A14056
;*****
A14048:
        LD      A, 1
        CALL   A14530
        LD      HL, A938
A14056:
        LD      A, (HL)
        OR      A
        JP      Z, A14080
        INC     HL
A14062:
        LD      C, (HL)
        INC     HL
        LD      B, (HL)
        INC     HL
        PUSH   AF
        PUSH   HL
        PUSH   BC
        LD      HL, A14074
        EX     (SP), HL
        JP      (HL)
;*****
A14074:
        POP    HL
        POP    AF
        DEC    A
        JP      NZ, A14062
A14080:
        POP    BC
        RET
;*****
A14082:

```



```
PUSH BC
LD HL, (A16107)
PUSH HL
AND 120
LD B, A
LD HL, (A16095)
JR A14148
```

```
;*****
```

```
A14095:
```

```
LD A, (HL)
EX AF, AF'
LD A, (HL)
AND 120
CP B
INC HL
INC HL
INC HL
JP NZ, A14146
PUSH BC
PUSH HL
LD A, C
CP 3
JR C, A14116
LD A, 2
```

```
A14116:
```

```
EX AF, AF'
AND 3
JP NZ, A14135
LD A, C
LD C, (HL)
INC HL
LD B, (HL)
LD HL, (A16097)
ADD HL, BC
LD C, A
LD B, (HL)
INC HL
JR A14138
```

;*****

A14135:

LD B,A
EX AF,AF'
LD C,A

A14138:

CALL A11851
POP HL
POP BC
JP Z,A14160

A14146:

INC HL
INC HL

A14148:

EX (SP),HL
LD A,H
OR L
DEC HL
EX (SP),HL
JP NZ,A14095
INC SP
INC SP
POP BC
RET

;*****

A14160:

DEC HL
DEC HL
DEC HL
LD A,(HL)
EX AF,AF'
POP BC
LD HL,(A16107)
DEC HL
OR A
SBC HL,BC
LD A,H
OR 140

```

        LD      H,A
        POP    BC
        EX     AF,AF'
        SCF
        RET
;*****
A14181:
        LD      HL,0
        LD      (A16105),HL
        CALL   A11920
;*****
        DB     27
        DB     "Too Many Variables Declared"
;*****
A14218:
        LD      HL,0
;*****
;Make variable.  Create variable of type A, name
length C,
;name address DE, vector HL.
;
A14221:
        PUSH   HL
        PUSH   AF
        CALL   A14082
        JP     C,A14383
        LD     HL,(A16107)
        INC    HL
        LD     A,252
        AND    H
        JR     NZ,A14181
        PUSH   DE
        PUSH   BC
        LD     A,5
        CALL   A13592
        LD     HL,(A16099)
        LD     D,H
        LD     E,L

```

```
LD BC, (A16097)
OR A
SBC HL, BC
LD B, H
LD C, L
LD HL, 5
ADD HL, DE
LD (A16099), HL
EX DE, HL
JR Z, A14275
DEC DE
DEC HL
LDDR
INC HL
INC DE
```

A14275:

```
POP BC
LD (A16097), DE
POP DE
LD B, C
LD A, C
CP 3
JR C, A14289
LD B, 2
```

A14289:

```
POP AF
LD (A16137), A
ADD A, 64
LD A, (A16137)
JP M, A14303
OR B
LD C, B
```

A14303:

```
LD (HL), A
ADD A, 64
LD A, C
POP BC
INC HL
```

```
LD      (HL),C
INC     HL
LD      (HL),B
INC     HL
LD      B,0
LD      C,A
JP      P,A14355
PUSH   DE
PUSH   HL
INC     A
CALL   A13592
LD      HL,(A16099)
LD      DE,(A16097)
OR      A
SBC    HL,DE
EX      DE,HL
POP     HL
LD      (HL),E
INC     HL
LD      (HL),D
LD      HL,(A16099)
POP     DE
LD      (HL),C
INC     HL
EX      DE,HL
LDIR
LD      (A16099),DE
JR      A14370
```

```
;*****
```

```
A14355:
```

```
LD      A,(DE)
CALL   A11870
JR      NC,A14363
OR      32
```

```
A14363:
```

```
LD      (HL),A
INC     DE
CPI
```

```

        JP      PE,A14355
A14370:
        LD      HL,(A16107)
        INC     HL
        LD      (A16107),HL
        DEC     HL
        LD      A,140
        OR      H
        LD      H,A
        RET
;*****
A14383:
        POP     AF
        POP     AF
        RET
;*****
A14386:
        LD      HL,0
        LD      (A16105),HL
        CALL    A11920
;*****
        DB      21
        DB      "Statement Too Complex"
;*****
;Hold new variables.
;
A14417:
        PUSH    BC
        PUSH    DE
        LD      HL,(A16105)
        LD      A,H
        OR      L
        JR      NZ,A14442
        LD      A,253
        LD      HL,A16105
        CALL    A7264
        LD      (A16105),HL
        INC     HL

```

```
        INC     HL
        INC     HL
        LD      (HL),0
A14442:
        LD      HL,(A16105)
        INC     HL
        INC     HL
        INC     HL
        LD      D,0
        LD      E,(HL)
        INC     (HL)
        INC     HL
        LD      A,41
        CP      E
        JR      C,A14386
        EX      DE,HL
        ADD     HL,HL
        LD      B,H
        LD      C,L
        ADD     HL,HL
        ADD     HL,BC
        ADD     HL,DE
        POP     DE
        POP     BC
        LD      (HL),B
        INC     HL
        LD      (HL),C
        INC     HL
        LD      (HL),D
        INC     HL
        LD      (HL),E
        INC     HL
        LD      BC,(A16103)
        LD      (HL),B
        INC     HL
        LD      (HL),C
        RET
```

```
;*****
```

A14483:

```
LD    HL, (A16105)
LD    A, H
OR    L
RET   Z
INC   HL
INC   HL
INC   HL
LD    B, (HL)
INC   B
DEC   B
RET   Z
```

A14496:

```
INC   HL
LD    A, (HL)
INC   HL
LD    C, (HL)
INC   HL
LD    D, (HL)
INC   HL
LD    E, (HL)
INC   HL
PUSH  BC
PUSH  HL
CALL  A14218
POP   DE
EX    DE, HL
LD    B, (HL)
INC   HL
LD    C, (HL)
LD    A, D
LD    (BC), A
INC   BC
LD    A, E
LD    (BC), A
POP   BC
DJNZ  A14496
LD    HL, 0
```



```

        LD      (A16105),HL
        RET
;*****
A14530:
        PUSH   HL
        LD     HL,(A16103)
        LD     (HL),A
        INC   HL
        LD     (A16103),HL
        LD     H,A
        LD     A,L
        CP    119
        LD     A,H
        POP   HL
        RET   NZ
        JP    A13918
;*****
A14549:
        LD     B,0
        EX    DE,HL
        ADD   HL,BC
        EX    DE,HL
        RET
;*****
A14555:
        LD     A,185
        JR    A14581
;*****
A14559:
        LD     A,186
        JR    A14581
;*****
A14563:
        LD     A,160
        JR    A14581
;*****
A14567:
        LD     A,161

```

```

        JR      A14581
;*****
A14571:
        LD      A,173
        JR      A14581
;*****
A14575:
        LD      A,183
        JR      A14581
;*****
A14579:
        LD      A,184
A14581:
        PUSH   HL
        LD      H,A
        CALL   A13659
        LD      B,H
        POP    HL
        RET    NC
        CALL   A13774
        RET    NC
        CP     B
        SCF
        CCF
        RET    NZ
        CALL   A14530
        CALL   A14549
        SCF
        RET
;*****
A14605:
        CALL   A14571
        JR      NC,A14622
A14610:
        CALL   A14571
        JR      C,A14610
        CALL   A14628
        CCF

```

```

        JR      NC,A14610
        RET
;*****
A14622:
        CALL   A14628
        RET    NC
        JR     A14610
;*****
A14628:
        CALL   A14563
        JR     NC,A14645
A14633:
        CALL   A14563
        JR     C,A14633
        CALL   A14567
        CCF
        JR     NC,A14633
        RET
;*****
A14645:
        CALL   A14567
        RET    NC
        JR     A14633
;*****
A14651:
        PUSH   BC
        CALL   A14605
        LD     HL,A14673
        JR     NC,A14663
        LD     HL,A14950
A14663:
        PUSH   HL
        CALL   A13659
        RET    NC
        POP   HL
        CALL   A14675
        SCF
A14673:

```

```

        POP     BC
        RET
;*****
A14675:
        DEC     B
        JP      Z,A15653
        INC     B
        JP      Z,A15395
        LD      A,(DE)
        CP      46
        JP      Z,A15653
        CP      34
        JP      Z,A15826
        CALL    A14575
        JR      C,A14702
        POP     BC
        POP     BC
        RET
;*****
A14702:
        CALL    A14777
        PUSH    HL
        CALL    A14579
        JP      NC,A15536
        POP     AF
        AND     1
        RET
;*****
A14716:
        PUSH    BC
        PUSH    HL
        CALL    A13659
        JR      NC,A14739
        CALL    A13774
        JR      NC,A14739
        CP      173
        JR      NC,A14739
        CALL    A14530

```

```

        CALL  A14549
        SCF
A14739:
        POP   HL
        POP   BC
        RET
;*****
;Math priority table.
;
A14742:
        DB    66      ; +
        DB    130     ; -
        DB    129     ; *
        DB    129     ; /
        DB    128     ; ^
        DB    3       ; <
        DB    3       ; >
        DB    3       ; <=
        DB    3       ; >=
        DB    3       ; <>
        DB    3       ; =
        DB    132     ; AND
        DB    133     ; OR
;*****
;Print "Illegal Equation".
;
A14755:
        JR    NC,A14777
A14757:
        CALL  A11920
;*****
        DB    16
        DB    "Illegal Equation"
;*****
;Equation evaluation in parsing.
;
A14777:
        LD    C,15

```

```
CALL A14651
RET NC
LD B,A
LD H,A
JR A14794
```

```
;*****
```

```
A14787:
```

```
CALL A14651
JP NC,A14757
LD H,A
```

```
A14794:
```

```
CALL A11777
PUSH DE
CALL A14716
LD L,15
JR NC,A14840
PUSH HL
SUB 160
LD HL,A14742
ADD A,L
LD L,A
JR NC,A14816
INC H
```

```
A14816:
```

```
LD A,(HL)
POP HL
LD L,A
AND 240
OR H
LD H,A
LD A,15
AND L
LD L,A
JR A14837
```

```
;*****
```

```
A14829:
```

```
PUSH BC
LD B,H
```

```

        LD      C,L
        CALL   A14787
        POP    BC
        LD     A,L
A14837:
        CP     C
        JR     C,A14829
;*****
A14840:
        BIT    3,C
        JR     NZ,A14857
        LD     A,B
        RLCA
        AND    H
        RRA
        JR     C,A14863
        LD     A,B
        XOR    H
        RRA
        JR     C,A14863
        BIT    6,B
A14857:
        POP    BC
        SCF
        RET    NZ
        RES    0,H
        RET
;*****
A14863:
        POP    DE
        LD     HL,A1333
        PUSH  HL
        JP    A11920
;*****
A14871:
        CALL  A14927
        RET
;*****

```

```

A14875:
    CALL    A14927
    CALL    A11921
;*****
    DB      16
    DB      "Numeric Equation"
;*****
;Parse string equation.
;
A14898:
    CALL    A14924
    RET
;*****
A14902:
    CALL    A14924
    CALL    A11921
;*****
    DB      15
    DB      "String Equation"
;*****
;Parse equation.
;
A14924:
    LD      B,1
    DB      33      ;LD HL,6
A14927:
    LD      B,0
    PUSH   BC
    CALL   A14777
    POP    BC
    RET    NC
    LD     A,H
    AND    1
    CP     B
    RET    NZ
    INC   SP
    INC   SP
    RET

```



```
;*****
A14943:
    CALL    A14962
    RET
;*****
A14947:
    CALL    A14962
A14950:
    CALL    A11921
;*****
    DB      8
    DB      "Equation"
;*****
A14962:
    CALL    A14777
    RET     NC
    INC     SP
    INC     SP
    RET
;*****
A14969:
    CALL    A14555
    RET     NC
    JP      A14875
;*****
A14976:
    LD      A,177
    CALL    A14581
    RET     NC
    CALL    A14875
    CALL    A15939
    JP      A14875
;*****
A14991:
    CALL    A15364
    AND     1
    PUSH   AF
    CALL    A15926
```

```

        POP     AF
        JR      Z,A15028
        CALL   A11921
;*****
        DB     13
        DB     "Real Variable"
;*****
;Parse LET.
;
A15020:
        CALL   A15364
        PUSH   AF
        CALL   A15902
        POP    AF
A15028:
RRA
        JP     NC,A14875
        JP     A14902
;*****
A15035:
        CALL   A15048
        CALL   A11921
;*****
        DB     6
        DB     "' THEN' "
;*****
A15048:
        LD     A,174
        CALL   A14581
        JR     C,A15089
        CP     178
        RET    NZ
        POP    HL
        CALL   A14549
        CALL   A13659
        JP     NC,A14009
        DEC   B
        JR     Z,A15083

```

```

        INC     B
        JP      NZ,A14009
        LD      A,178
        CALL    A14530
        JP      A13946
;*****
A15083:
        LD      A,179
        CALL    A14530
        DB      62           ;LD A,225 ;in this frame,
won't POP HL           ;I hate overlapping code!

A15089:
        POP HL           ;gets HL off stack in this
frame
        JP      A15756
;*****
A15093:
        LD      A,176
        CALL    A14581
        RET     NC
        JP      A14875
;*****
A15102:
        LD      A,180
        CALL    A14581
A15107:
        CALL    A14875
        CALL    A15939
        CALL    A14875
        LD      A,180
        CALL    A14581
        JP      C,A15107
        RET
;*****
A15125:
        CALL    A15155
        CALL    A11921

```

```

;*****
    DB      19
    DB      "'FN<function name>'"
A15151:
    DB      2
A15152:
    DB      "FN"
;*****
    NOP
;*****
A15155:
    CALL    A13659
    RET     NC
    LD      A,C
    CP      2
    RET     C
    LD      BC,514           ;not an address
                               ;need to figure out what it
is
    LD      HL,A15152
    CALL    A11851
    RET     NZ
    PUSH   DE
    INC     DE
    INC     DE
    CALL    A13659
    POP     DE
    RET     NC
    INC     B
    DEC     B
    RET     NZ
    POP     HL
    JP      A14991
;*****
A15188:
    CALL    A11921
;*****
    DB      17

```

```

        DB      "'GOTO' or 'GOSUB'"
;*****
A15209:
        LD      A,174
        CALL   A14581
        JR     C,A15223
        LD      A,175
        CALL   A14581
        JR     NC,A15188
A15223:
        CALL   A15756
        CALL   A14555
        JR     C,A15223
        RET
;*****
A15232:
        CALL   A13659
        RET    NC
        DEC    B
        JP     Z,A15767
        JP     A16007
;*****
A15243:
        CALL   A15271
        RET
;*****
A15247:
        CALL   A15271
        CALL   A11921
;*****
        DB     17
        DB     "Line Number Range"
;*****
A15271:
        CALL   A13659
        RET    NC
        DEC    B
        JR     Z,A15299

```

```

        INC     B
        RET     Z
A15280:
        CALL   A14555
        JR     C,A15289
        CALL   A14567
        RET     NC
A15289:
        CALL   A13659
        RET     NC
        DEC     B
        RET     NZ
        POP     HL
        JP     A15767
;*****
A15299:
        POP     HL
        CALL   A15767
        CALL   A15280
        RET
;*****
A15307:
        CALL   A11777
        RET     Z
        PUSH   BC
        LD     C,32
        CP     36
        JR     Z,A15327
        LD     C,16
        CP     37
        JR     Z,A15327
        LD     C,0
        DB     62                ;LD A,19 ;in this frame,
doesn't INC DE                ;I hate overlapping code!
A15327:
        INC DE                ;INC DE in this frame
        CALL   A11777

```

```

        CP      40
        LD      A,C
        POP     BC
        RET     NZ
        OR      8
        RET
;*****
A15339:
        CALL   A11920
;*****
        DB     21
        DB     "Illegal Variable Name"
;*****
A15364:
        CALL   A15386
A15367:
        CALL   A11921
;*****
        DB     8
        DB     "Variable"
;*****
A15379:
        CALL   A15386
        POP     HL
        JP     A15573
;*****
A15386:
        CALL   A13659
        RET     NC
        CCF
        INC     B
        DEC     B
        RET     NZ
        POP     HL
A15395:
        LD     B,0
        LD     A,C
        CP     2

```

```
JR      C,A15429
PUSH   BC
LD     BC,514      ;not an address
LD     HL,A15152
CALL   A11851
POP    BC
JR     NZ,A15429
LD     H,192
INC    DE
INC    DE
CALL   A13659
JR     NC,A15367
INC    B
DEC    B
JR     NZ,A15367
LD     B,H
```

A15429:

```
CALL   A29202
JR     C,A15339
PUSH   DE
LD     A,B
CALL   A14549
```

A15439:

```
LD     B,A
CALL   A15307
OR     B
LD     B,A
EX     DE,HL
EX     (SP),HL
EX     DE,HL
CALL   A14082
JR     C,A15459
PUSH   BC
CALL   A14417
POP    BC
LD     A,B
```

A15459:

```
ADD    A,64
```



```

        JP      P,A15466
        LD      C,1
A15466:
        LD      A,H
        CALL   A14530
        LD      A,L
        CALL   A14530
        DEC    C
        INC    DE
        JR     Z,A15480
        DEC    C
        INC    DE
A15480:
        LD      A,C
        CALL   A14530
        INC    C
        DEC    C
        JR     Z,A15504
A15488:
        LD      A,(DE)
        CALL   A11870
        JR     NC,A15496
        OR     32
A15496:
        CALL   A14530
        INC    DE
        DEC    C
        JP     NZ,A15488
A15504:
        POP    DE
        PUSH   BC
        CALL   A15516
        POP    BC
        XOR    A
        BIT   5,B
        RET   Z
        INC   A
        RET

```

```

;*****
A15516:
    CALL  A14575
    RET   NC
A15520:
    CALL  A14943
    JP    NC,A14950
    CALL  A14555
    JP    C,A15520
A15532:
    CALL  A14579
    RET   C
A15536:
    CALL  A11921
;*****
    DB    3
    DB    " ' ) ' "
;*****
;Parse INPUT.
;
A15543:
    CALL  A11777
    CP    34
    JP    NZ,A15574
    CALL  A15826
    CALL  A14559
    JP    C,A15574
    CALL  A11921
;*****
    DB    3
    DB    " ' ; ' "
;*****
;Parse NEXT.  Includes undocumented NEXT a,b,c
[,...] syntax.
;
A15567:
    CALL  A15379
A15570:

```

```

        CALL    A14555
A15573:
        RET     NC
A15574:
        CALL    A15364
        JP      A15570
;*****
A15580:
        CALL    A14943
A15583:
        CALL    A14559
        JP      C,A15580
        CALL    A14555
        JP      C,A15580
        CALL    A11777
        RET     Z
        CP      58
        RET     Z
        CP      59
        JR      Z,A15583
        CP      44
        JR      Z,A15583
        CP      34
        JR      Z,A15627
        CP      46
        JR      Z,A15627
        CALL    A11884
        JR      C,A15627
        CALL    A11864
        RET     NC
A15627:
        LD      A,186
        CALL    A14530
        JP      A15580
;*****
A15635:
        CALL    A11920
;*****

```

```

        DB      14
        DB      "Number Too Big"
;*****
A15653:
        CALL   A2576
        JR     C,A15635
        LD     A,(A16166)
        OR     A
        JP     Z,A15781
        CP     129
        JP     C,A15683
        CP     144
        JP     NC,A15683
        LD     HL,(A16162)
        LD     A,H
        OR     L
        JP     Z,A15718
A15683:
        LD     A,146
        CALL   A14530
        LD     HL,(A16162)
        LD     A,L
        CALL   A14530
        LD     A,H
        CALL   A14530
        LD     HL,(A16164)
        LD     A,L
        CALL   A14530
        LD     A,H
        CALL   A14530
        LD     A,(A16166)
        CALL   A14530
        XOR    A
        RET
;*****
A15718:
        LD     A,(A16164)
        RRA

```

```
JP      C,A15683
CALL   A2354
SCF
PUSH   HL
CALL   A2407
CALL   A2488
POP    HL
JP     NZ,A15683
LD     A,H
OR     A
JP     P,A15770
LD     A,161
CALL   A14530
CALL   A2392
JP     A15770
```

```
;*****
```

```
A15756:
```

```
LD     HL,A16007
PUSH   HL
CALL   A13659
RET    NC
DEC    B
RET    NZ
POP    HL
```

```
A15767:
```

```
CALL   A12451
```

```
A15770:
```

```
LD     A,H
OR     A
JP     NZ,A15799
LD     A,L
CP     10
JP     NC,A15788
```

```
A15781:
```

```
OR 128
```

```
CALL   A14530
XOR    A
RET
```

;*****

A15788:

```
LD    A,138
CALL  A14530
LD    A,L
CALL  A14530
XOR   A
RET
```

;*****

A15799:

```
LD    A,139
CALL  A14530
LD    A,L
CALL  A14530
LD    A,H
CALL  A14530
XOR   A
RET
```

;*****

A15814:

```
LD    B,58
DB    33
```

B,value cascade

;LD HL,6 ;multiple-frame LD

;B=58

;I hate overlapping code!

A15817:

```
LD    B,0
LD    A,144
EX    AF,AF'
OR    A
JP    A15832
```

;B=0 here

;*****

A15826:

```
LD    A,145
LD    B,34
EX    AF,AF'
SCF
```

A15832:

```

        INC     DE
        EX     AF,AF'
        LD     C,0
        CALL  A14530
        LD     HL,(A16103)
        PUSH  HL
        CALL  A14530
A15846:
        LD     A,(DE)
        OR     A
        JP     Z,A15868
        INC   DE
        CP     B
        JP     Z,A15885
        CP     34
        CALL  Z,A15898
A15861:
        INC   C
        CALL  A14530
        JP     A15846
;*****
A15868:
        LD     A,B
        CP     34
        JR     Z,A15878
A15873:
        POP   HL
        LD     (HL),C
        LD     A,1
        RET
;*****
A15878:
        CALL  A11921
;*****
        DB     3
        DB     39,34,39      ;'''
;*****
A15885:

```

```

        CP      58
        JR      NZ,A15873
        EX      AF,AF'
A15890:
        JR      NC,A15895
        EX      AF,AF'
        JR      A15861
;*****
A15895:
        DEC     DE
        JR      A15873
;*****
A15898:
        EX      AF,AF'
        CCF
        EX      AF,AF'
        RET
;*****
A15902:
        LD      A,170
        CALL   A14581
        RET     C
        JP      A14009
;*****
A15911:
        LD      A,181
        CALL   A14530
        CALL   A11777
        CP      58
        JP      NZ,A16022
        INC     DE
        RET
;*****
A15926:
        LD      A,170
        CALL   A14581
        RET     C
        CALL   A11921

```



```

;*****
    DB      3
    DB      "'=' "
;*****
A15939:
    CALL   A14555
    RET    C
    CALL   A11921
;*****
    DB      3
    DB      "'',"
;*****
A15950:
    LD     A,182
    CALL   A14581
    RET    C
    CALL   A11921
;*****
    DB      3
    DB      "'#' "
;*****
A15963:
    LD     A,180
    CALL   A14581
    RET    C
    CALL   A11921
;*****
    DB      4
    DB      "'TO' "
;*****
A15977:
    LD     A,177
    CALL   A14581
    RET    C
    CALL   A11921
;*****
    DB      4
    DB      "'AT' "

```

```

;*****
A15991:
    LD     A,174
    CALL  A14581
    RET   C
    CALL  A11921
;*****
    DB     6
    DB     "'GOTO'"
;*****
A16007:
    CALL  A11921
;*****
    DB     11
    DB     "Line Number"
;*****
A16022:
    CALL  A11921
;*****
    DB     3
    DB     "':'"
;*****
;Cold boot startup.  Bypassed in SmartBASIC 1.x.
;
    LD     SP,SB10_STACK
    CALL  16481
;*****
;Central loop.
;
A16035:
    CALL  A5975
A16038:
    CALL  A12150
    LD     HL,A16038
    PUSH  HL
    CALL  A12159
    LD     DE,A16247
    CALL  A11777

```

```

    RET     Z
    CALL   A11884
    JP     C,A16068
    CALL   A13833
    POP    HL
    JP     A6112
;*****
A16068:
    CALL   A12451
    LD     (A16207),HL
    CALL   A13969
    JP     Z,A16086
    CALL   A13833
    JP     A12618
;*****
A16086:
    JP     A12782
;*****
A16089:
    DW     53632           ;pointer to start of line
number table
A16091:
    DW     0              ;number of line numbers
A16093:
    DW     0              ;length of line number table
A16095:
    DW     LOMEM          ;pointer to start of
variable table/LOMEM
A16097:
    DW     VASCII         ;pointer to end of variable
table

A16098     EQU     A16097+1

A16099:
    DW     $SPACE        ;pointer to end of variable
command name table
A16101:

```

```

        DW      53632          ;pointer to start of crunch
code table
A16103:
        DW      A16503        ;pointer to end of crunch
code buffer
A16105:
        DW      VASCII        ;pointer to the string of
new variables
A16107:
        DW      NVARs        ;number of variables
A16109:
        DW      53631         ;pointer to start of
variable value table
A16111:
        DW      $SPACE        ;pointer to end of string
space

A16112      EQU      A16111+1

A16113:
        DW      0              ;temporary pointer to end of
string space

A16114      EQU      A16113+1

A16115:
        DW      $SPACE        ;pointer to start of string
space
A16117:
        DW      53632         ;pointer to current DATA
line number
A16119:
        DW      0              ;pointer to current DATA
crunch code

A16120      EQU      A16119+1

A16121:

```

```

        DB      0          ;number of bytes remaining
in DATA crunch code
A16122:
        DW      0          ;storage of DE' for CONT
A16124:
        DW      0          ;storage of HL' for CONT
A16126:
        DW      0          ;line number for ONERR
A16128:
        DB      20         ;command error number
A16129:
        DB      255        ;current SPEED
A16130:
        DW      A6345       ;vector to USR routine (hex
18C9)
A16132:
        DW      A8419       ;vector to & routine
A16134:
        DB      3          ;ASCII code for break
(default ^C)
A16135:
        DB      19         ;ASCII code for pause
(default ^S)
A16136:
        DB      255        ;pause indicator
A16137:
        DS      2          ;temporary storage area
A16139:
        DS      2          ;
A16141:
        DS      2          ;
A16143:
        DS      2          ;
A16145:
        DS      2          ;
A16147:
        DS      1          ;
A16148:

```

```

        DB      32          ;ASCII code for indenting
characters
A16149:
        DW      65535      ;pointer to POKE upper limit
A16151:
        DB      0          ;sign for math operation
results
A16152:
        DS      2          ;temporary FPA data and
pointers
A16154:
        DS      1          ;
A16155:
        DS      1          ;
A16156:
        DS      2          ;
A16158:
        DS      2          ;FPA1 data used in division
A16160:
        DS      1          ;
A16161:
        DB      0          ;FPA1 status byte
A16162:
        DS      1          ;FPA1
A16163:
        DS      1          ;
A16164:
        DS      1          ;
A16165:
        DS      1          ;
A16166:
        DS      1          ;
A16167:
        DS      2          ;FPA2 data used in division
A16169:
        DS      1          ;
A16170:
        DB      0          ;FPA2 status byte

```

```

A16171:
    DS      1          ;FPA2
A16172:
    DS      1          ;
A16173:
    DS      1          ;
A16174:
    DS      1          ;
A16175:
    DS      1          ;
A16176:
    DB      80        ;maximum width of printer
line
A16177:
    DB      80        ;position of head on printer
A16178:
    DB      9         ;temporary FPA for SIN, COS,
etc.
A16179:
    DS      2          ;
A16181:
    DS      2          ;
A16183:
    DS      2          ;temporary FPA for
calculations
A16185:
    DS      2          ;
A16187:
    DS      2          ;
A16189:
    DS      1          ;
A16190:
    DS      1          ;random number seed (4
bytes)
A16191:
    DS      1          ;
A16192:
    DS      1          ;

```

```

A16193:
    DS      1          ;
A16194:
    DB      0          ;sign of floating point
numbers
A16195:
    DW      A17289     ;IN# vector used by READ
A16197:
    DW      A17289     ;IN# vector to receive data
from device
A16199:
    DW      A12043     ;PR# vector used by LOAD,
WRITE
A16201:
    DW      A12043     ;PR# vector to transmit data
to device
A16203:
    DW      A17234     ;vector to screen PRINT
A16205:
    DW      0          ;length of crunch code
buffer

A16206      EQU      A16205+1

A16207:
    DW      0          ;line number to GOTO, GOSUB

A16208      EQU      A16207+1

A16209:
    DB      0          ;temporary ASCII code for
line indenting
A16210:
    DB      255        ;null string
A16211:
    DW      0          ;unused?
A16213:
    DW      A12043     ;PR #0 vector / video screen

```



```

        DW      A12010      ;PR #1 vector / ADAM printer
        DW      A31218      ;PR #2 vector / parallel
printer
        DW      A30897      ;PR #3 vector / serial
printer
        DW      A12043      ;PR #4 vector
        DW      A12043      ;PR #5 vector
        DW      A12043      ;PR #6 vector
        DW      A12043      ;PR #7 vector
A16229:
        DW      A17289      ;IN #0 vector / ADAM
keyboard
        DW      A17289      ;IN #1 vector
        DW      A17289      ;IN #2 vector
        DW      A17289      ;IN #3 vector
        DW      A17289      ;IN #4 vector
        DW      A17289      ;IN #5 vector
        DW      A17289      ;IN #6 vector
        DW      A17289      ;IN #7 vector
;*****
A16245:
        DB      254          ;max length of input buffer
A16246:
        DB      0            ;used length of input buffer
A16247:
        DS      256          ;input buffer
;*****
A16503:
        DS      256          ;crunch code buffer

A16507      EQU      A16503+4

;*****
A16759:
        DS      1            ;
A16760:
        DS      1            ;
A16761:

```

```

        DS      2          ;
A16763:
        DS      2          ;coordinates of last plotted
hi-res point
A16765:
        DB      1          ;current SCALE
A16766:
        DW      A26574     ;pointer to shape table
A16768:
        DB      0,248,8,0,0,8,248,0      ;DRAW and ROT
area
A16776:
        DB      15         ;current COLOR
A16777:
        DB      15         ;current HCOLOR
A16778:
        DS      5          ;PDL buffer player 2
A16783:
        DS      5          ;PDL buffer player 1
A16788:
        DB      1,0,2,255,255      ;BSAVE file header

A16790      EQU      A16788+2
A16791      EQU      A16788+3

A16793:
        DS      4          ;length of file (loword,
hiword)

A16795      EQU      A16793+2

A16797:
        DS      12         ;temporary filename in 1st
FCB
A16809:
        DS      12         ;temporary filename in 2nd
FCB
A16821:

```

```

        DS      1          ;device # for drive
A16822:
        DS      1          ;EOS file #
A16823:
        DB      1          ;another EOS file #
A16824:
        DB      13         ;start of a 3-byte special
file read buffer
                                ;1 and 2-byte values (with
A16825) are taken out of here.
                                ;Maybe this is used by code
to try to interpret H-type
                                ;SmartWriter files? Can
SmartBASIC read a program that
                                ;was edited by SmartWriter?
A16825:
        DB      255        ;byte 2 of special file read
buffer
A16826:
        DB      255        ;byte 3 of special file read
buffer
                                ;??? some count or flag
during file read
                                ;if it's nonzero, we do a
_CV_A for some reason
                                ;if it's zero, we skip the
_CV_A
A16827:
        DS      1          ;temporary file #
A16828:
        DS      1          ;temporary file # for READ
during WRITE
A16829:
        DW      A23209     ;MON/NOMON I vector
A16831:
        DW      A21981     ;MON/NOMON C vector
A16833:
        DW      A21874     ;MON/NOMON L vector

```

```

A16835:
    DW      A23209      ;MON/NOMON 0 vector

A16836      EQU      A16837-1

A16837:
    DS      10          ;1st FCB header

A16838      EQU      A16837+1

A16847:
    DS      10          ;2nd FCB header
A16857:
    DS      14          ;name and length of 1st file
A16871:
    DS      14          ;name and length of 2nd file
A16885:
    DS      26          ;complete directory entry
for file

A16897      EQU      A16885+12 ;attribute byte
A16902      EQU      A16885+17 ;allocated length

A16911:
    DB      255         ;end of entry?
A16912:
    DS      12          ;filename for CATALOG
A16924:
    DS      26          ;directory entry for file to
APPEND ??

A16939      EQU      A16924+15 ;hiword of startblock

A16950:
    DB      255         ;end of entry?
A16951:
    DS      2          ;APPEND temporary storage
;*****

```

```

A16953:
    DB      0          ;cursor character
A16954:
    DB      32         ;space character
A16955:
    DB      0          ;current character
A16956:
    DB      1          ;screen left margin
A16957:
    DB      31         ;screen right margin
A16958:
    DB      0          ;screen top margin
A16959:
    DB      23         ;screen bottom margin
;*****
A16960:
    DS      8          ;pattern buffer
A16968:
    DS      8          ;pattern buffer
    DS      3          ;garbage leftovers from old
TEXT31 buffer?
;*****
;some patch code
;
A16979:
    LD      HL,A18321
    JP      Z,A17440
    LD      HL,A17004
    ADD    A,(HL)
    JP      A17459
;*****
    DB      255        ;old end of TEXT31 buffer
;*****
A16993:
    DB      24         ;# screen lines for HOME
A16994:
    DB      31         ;# screen columns for HOME
A16995:

```

```

        DB      0          ;start line for HOME
A16996:
        DB      1          ;start column for HOME
A16997:
        DW     6144        ;VRAM address of name table
A16999:
        DW      0          ;VRAM address of pattern
table

A17000      EQU     A16999+1

A17001:
        DB      0          ;current cursor line
A17002:
        DB      1          ;current cursor column
A17003:
        DS      1          ;current input byte (last
key pressed or char read)
A17004:
        DB      0          ;ASCII base
A17005:
        DB      0          ;blinking cursor indicator
A17006:
        DB      0          ;cursor ASCII base
A17007:
        DB     255        ;current name table
A17008:
        DB      0          ;current screen graphics
mode
A17009:
        DB     255        ;character print indicator
A17010:
        DB      0          ;character flash indicator
A17011:
        DB     12         ;# of NMI cycles between
INVERSE and NORMAL toggle
A17012:
        DW     6144        ;VRAM address of FLASHing

```

name table

```
A17014:
    DB      0          ;^D indicator
A17015:
    DB      0          ;temporary output storage
A17016:
    DB      0          ;length of ^D buffer
A17017:
    DS      23         ;^D buffer

A17039      EQU      A17017+22      ;end of ^D buffer

A17040:
    DW      A17017      ;pointer to current position
in ^D buffer
A17042:
    DS      2          ;temporary filename pointer
($$$$1 or $$$2)
A17044:
    DS      2          ;default filename pointer
;*****
A17046:
    LD      BC,0
    CALL    WRITE_REGISTER
    LD      BC,384
    CALL    WRITE_REGISTER
    LD      BC,1793      ;

A17059      EQU      $-2          ;color POKE

    CALL    WRITE_REGISTER
    LD      HL,0
    LD      A,3
    CALL    INIT_TABLE
    LD      HL,6144
    LD      A,2
    CALL    INIT_TABLE
    LD      HL,8192
```

```

LD      A,4
CALL   INIT_TABLE
LD      HL,7936
XOR    A
CALL   INIT_TABLE
LD      HL,14336
LD      A,1
CALL   INIT_TABLE
LD      A,208
LD      DE,1
LD      HL,7936
CALL   FILL_VRAM
LD      C,241      ;

```

```

A17115      EQU      $-1      ;color

```

```

CALL   A18776
LD      HL,8192
CALL   FILL_VRAM
LD      A,31      ;

```

```

A17126      EQU      $-1      ;screen width patch

```

```

LD      DE,16
LD      HL,8208
CALL   FILL_VRAM

```

```

A17136:

```

```

CALL   LOAD_ASCII
LD      HL,A17226
LD      DE,248
LD      BC,8
CALL   WRITE_VRAM
CALL   A11651
LD      BC,128
LD      DE,1024
CALL   PUT_ASCII

```

```

A17163:

```

```

LD      A,32

```



```

        LD      DE,768      ;
A17166      EQU    $-2      ;bytecount

        LD      HL,6144
        CALL   FILL_VRAM
        LD      A,32
        LD      DE,768      ;

A17177      EQU    $-2      ;bytecount

        LD      HL,2048
        CALL   FILL_VRAM
        LD      HL,6144
        LD      (A17012),HL
        LD      HL,12
        LD      (A17011),HL
        LD      BC,7703      ;

A17199      EQU    $-1      ;

        LD      DE,256
        LD      HL,6144
        LD      A,0
        EX     AF,AF'
        LD      A,0
        CALL   A17334
        LD      BC,480      ;

A17215      EQU    $-2      ;color

        CALL   WRITE_REGISTER
        LD      A,255
        LD      (A17009),A
        RET
;*****
;Pattern for CHR$(31) to replace ROM pattern.
;Lazer Microsystems programmer Joel Lagerquist says

```

that this was put in by
;fellow programmer Johnny Fitzgerald, who was a big
windsurfing fan.

;(JKL to RFD, spring 1997)

;

A17226:

```
DB    008H    ;    .    B    .
DB    018H    ;    .    BB   .
DB    038H    ;    .   BBB  .
DB    078H    ;    .  BBBB .
DB    0F8H    ;    .BBBBB .
DB    008H    ;    .    B    .
DB    0FEH    ;    .BBBBBBB .
DB    000H    ;    .        .
```

;*****

A17234:

```
LD    B,A
LD    A,(A17008)
CP    3
RET   Z
LD    A,B
```

A17242:

```
JP    A33966
```

;*****

```
NOP
NOP
```

;*****

A17247:

```
OR    A
JR    NZ,A17268
EX    AF,AF'
CP    32
JR    C,A17270
CP    128
JR    NC,A17270
PUSH  HL
LD    HL,A17004
ADD   A,(HL)
```

```

        POP     HL
        CALL   A17448
A17268:
        POP     AF
        RET
;*****
A17270:
        CALL   A17416
        POP     AF
        RET
;*****
A17275:
        DEC     DE
        LD      A,D
        OR      E
        JR      NZ,A17292
        LD      A,(A17005)
        OR      A
        JR      NZ,A17289
        CALL   A17396
A17289:
        LD      DE,1024
A17292:
        CALL   A17310
        INC     A
        JR      NZ,A17275
        LD      A,(A17003)
        CP      16
        RET     NZ
        CALL   A34012
        JP      A17289
;*****
A17310:
        CALL   _END_RD_KBD
        LD      (A17003),A
        LD      A,0
        JR      NC,A17333
        JR      Z,A17328

```

```

        CALL  _START_RD_KBD
        XOR   A
        JR    A17333
;*****
A17328:
        CALL  _START_RD_KBD
        LD    A,255
A17333:
        RET
;*****
A17334:
        INC   B
        INC   C
        LD    (A16993),BC
        LD    (A16995),DE
        LD    (A17001),DE
        LD    (A16997),HL
        LD    (A16999),A
        EX    AF,AF'
        LD    (A17000),A
        LD    A,D
        LD    (A16956),A
        ADD   A,B
        DEC   A
        LD    (A16957),A
        LD    A,E
        LD    (A16958),A
        ADD   A,C
        DEC   A
        LD    (A16959),A
        LD    A,(A16954)
        LD    (A16955),A
        LD    A,(A16953)
        CALL  A18200
        RET
;*****
A17389:
        LD    C,A

```

```

        XOR    A
        LD     (A17009),A
        LD     A,C
        RET
;*****
A17396:
        JP     A33989
;*****
A17399:
        PUSH  IX
        PUSH  IY
        PUSH  DE
        CALL  A17389
        LD     A,(A16953)
        RLA
        LD     A,254
        RRA
        JR     A17501
;*****
A17416:
        PUSH  AF
        PUSH  BC
        PUSH  HL
        PUSH  IX
        PUSH  IY
        PUSH  DE
        CALL  A17389
        LD     HL,A18304
        LD     BC,17           ;@@@ # of control
characters in table
        CPIR
        JP     A16979
;*****
        NOP
        NOP
;*****
A17440:
        ADD   HL,BC

```

```
ADD    HL,BC
LD     B,A
LD     A,(HL)
INC    HL
LD     H,(HL)
LD     L,A
JP     (HL)
```

```
;*****
```

```
A17448:
```

```
PUSH  AF
PUSH  BC
PUSH  HL
PUSH  IX
PUSH  IY
PUSH  DE
CALL  A17389
```

```
A17459:
```

```
CALL  A18200
JP     A31717
```

```
;*****
```

```
A17465:
```

```
INC    H
LD     A,(A16957)
CP     H
JR     NC,A17492
LD     A,(A16956)
LD     H,A
INC    L
LD     A,(A16959)
CP     L
JR     NC,A17492
DEC    L
PUSH  HL
CALL  A18246
CALL  A18112
POP   HL
```

```
A17492:
```

```
LD     (A17001),HL
```

```

A17495:
    CALL    A18249
A17498:
    LD      A, (A16955)
A17501:
    CALL    A18288
    LD      HL, A16968
    CALL    READ_VRAM
    LD      A, (A16953)
    CALL    A18288
    LD      HL, A16960
    CALL    READ_VRAM
    LD      B, 8
    LD      HL, A16960
    LD      DE, A16968
A17530:
    LD      A, (DE)
    XOR     (HL)
    LD      (DE), A
    INC     HL
    INC     DE
    DJNZ   A17530
    EX     DE, HL
    LD      A, (A16953)
    RLA
    LD      A, 254
    RRA
    PUSH   AF
    LD      L, A
    LD      H, 0
    ADD    HL, HL
    ADD    HL, HL
    ADD    HL, HL
    LD     BC, (A16999)
    ADD    HL, BC
    EX     DE, HL
    LD     BC, 8
    CALL   WRITE_VRAM

```

```

        POP     BC
        LD      A, (A17006)
        PUSH   AF
        XOR    A
        LD      (A17006), A
        LD      A, B
        CALL   A18200
        POP    AF
        LD      (A17006), A
A17581:
        LD      A, (A17008)
        OR     A
        JR     NZ, A17594
        LD      A, (A17010)
        OR     A
        CALL   NZ, A123
A17594:
        LD      A, 255
        LD      (A17009), A
        LD      (A17007), A
        POP    DE
        POP    IY
        POP    IX
        POP    HL
        POP    BC
        POP    AF
        RET
;*****
A17611:
        CALL   A18047
        LD      A, (A16956)
        LD      (A17002), A
        CALL   A18246
        LD      B, 10
A17625:
        LD      HL, (A17001)
        LD      A, (A16959)
        JP     A31477

```


;*****

A17634:

INC L
LD A,10
CP B
CALL NZ,A18197
JP A17492

;*****

A17644:

LD A,10
CP B
JR NZ,A17581
CALL A18115
LD HL,(A17001)
JP A17495

;*****

A17658:

LD HL,(A17001)
LD A,(A16958)
JP A31491

;*****

A17667:

DEC L

A17668:

CALL A18197
JP A17492

;*****

A17674:

LD HL,(A17001)
LD A,(A16957)
JP A31505

;*****

A17683:

INC H
JP A17668

;*****

A17687:

LD A,(A16959)

```

        CP      L
        JP      A31519
;*****
A17694:
        INC     L
        LD      A, (A16956)
        LD      H, A
        JP      A17668
;*****
A17702:
        LD      HL, (A17001)
        LD      A, (A16956)
        JP      A31532
;*****
A17711:
        DEC     H
        JP      A17668
;*****
A17715:
        LD      A, (A16958)
        CP      L
        JP      A31546
;*****
A17722:
        DEC     L
        LD      A, (A16957)
        LD      H, A
        JP      A17668
;*****
A17730:
        LD      A, 255
        LD      (A17007), A
        CALL   A17761
        LD      A, (A17008)
        OR      A
        JR      NZ, A17752
        LD      A, 15
        LD      (A17007), A

```

```

        CALL    A17761
A17752:
        LD      A, (A16954)
        LD      (A16955), A
        JP      A17498
;*****
A17761:
        CALL    A18085
        PUSH   DE
        PUSH   BC
        LD      A, (A16955)
        LD      (HL), A
        ADD    HL, BC
        LD      D, H
        LD      E, L
        DEC    HL
        LDDR
        LD      A, (A16953)
        LD      (DE), A
        EX     DE, HL
        POP    BC
        POP    DE
        JP     WRITE_VRAM
;*****
A17786:
        LD      A, 255
        LD      (A17007), A
        CALL   A17812
        LD      A, (A17008)
        OR     A
        JP     NZ, A17498
        LD      A, 15
        LD      (A17007), A
        CALL   A17812
        JP     A17498
;*****
A17812:
        CALL   A18085

```

```
PUSH DE
PUSH BC
PUSH HL
LD D,H
LD E,L
INC HL
LDIR
POP HL
LD A,(A16954)
DEC DE
LD (DE),A
LD A,(HL)
LD (A16955),A
LD A,(A16953)
LD (HL),A
POP BC
POP DE
JP WRITE_VRAM
```

```
;*****
```

```
A17842:
```

```
LD A,(A16993)
LD B,A
LD HL,(A16995)
CALL A17999
LD A,(A16954)
LD (A16955),A
JR A17863
```

```
;*****
```

```
A17860:
```

```
CALL A18197
```

```
A17863:
```

```
LD HL,(A16995)
JP A17492
```

```
;*****
```

```
A17869:
```

```
CALL A18047
JP A17498
```

```
;*****
```

```
A17875:
    CALL    A18047
    LD      HL, (A17001)
    INC     L
    LD      A, (A16958)
    LD      C, A
    LD      A, (A16993)
    ADD     A, C
    SUB     L
    JP      Z, A17498
    LD      B, A
    LD      A, (A16956)
    LD      H, A
    CALL    A17999
    JP      A17498
```

```
;*****
```

```
A17905:
    LD      A, (A69)
    CP      D
    JR      Z, A17914
    JP      NC, A17581
```

```
A17914:
    LD      A, (A70)
    CP      D
    JR      Z, A17923
    JP      C, A17581
```

```
A17923:
    LD      A, (A71)
    CP      E
    JR      Z, A17932
    JP      NC, A17581
```

```
A17932:
    LD      A, (A72)
    CP      E
    JR      Z, A17941
    JP      C, A17581
```

```
A17941:
    EX      DE, HL
```

```

        JP      A17668
;*****
A17945:
        LD      A, (SOUNDPORT)
        LD      C, A
        LD      A, 143
        OUT     (C), A
        LD      A, 17
        OUT     (C), A
        LD      A, 144
        OUT     (C), A
        LD      HL, 1920      ;not an address; delay count
for BEEP
A17964:
        EX      (SP), HL
        EX      (SP), HL
        DEC     HL
        LD      A, H
        OR      L
        JR      NZ, A17964
        LD      A, 159
        OUT     (C), A
        JP      A35041
;*****
A17978:
        PUSH    HL
        PUSH    BC
        JR      A17989
;*****
A17982:
        PUSH    HL
        PUSH    BC
        LD      HL, A33662    ;screen buffer
        LD      B, 32

A17988      EQU    $-1      ;current screen width

A17989:

```

```

        LD      A, (A16954)
A17992:
        LD      (HL), A
        INC    HL
        DJNZ   A17992
        POP    BC
        POP    HL
        RET
;*****
A17999:
        CALL   A17982
        LD      A, (A16994)
        LD      C, A
        PUSH   BC
        PUSH   HL
        LD      A, 255
        LD      (A17007), A
        CALL   A18028
        POP    HL
        POP    BC
        LD      A, (A17008)
        OR     A
        RET    NZ
        LD      A, 15
        LD      (A17007), A
A18028:
        PUSH   BC
        PUSH   HL
        CALL   A18266
        LD      B, 0
        LD      HL, A33662 ; screen buffer
        CALL   WRITE_VRAM
        POP    HL
        INC    L
        POP    BC
        DJNZ   A18028
        RET
;*****

```

A18047:

```
LD    A,255
LD    (A17007),A
CALL  A18065
LD    A,(A17008)
OR    A
RET   NZ
LD    A,15
LD    (A17007),A
```

A18065:

```
CALL  A18085
PUSH  HL
PUSH  BC
LD    B,C
CALL  A17978
POP   BC
POP   HL
LD    A,(A16954)
LD    (A16955),A
JP    WRITE_VRAM
```

;*****

A18085:

```
LD    HL,(A17001)
LD    A,(A16957)
INC   A
SUB   H
LD    C,A
CALL  A18266
LD    HL,A33662    ;screen buffer
LD    B,0
PUSH  BC
PUSH  HL
PUSH  DE
CALL  READ_VRAM
POP   DE
POP   HL
POP   BC
RET
```


;*****

A18112:

CALL A18197

A18115:

LD A,255

LD (A17007),A

CALL A18133

LD A,(A17008)

OR A

RET NZ

LD A,15

LD (A17007),A

A18133:

LD HL,(A16995)

PUSH HL

CALL A18266

POP HL

LD A,(A16993)

LD B,A

DEC B

A18146:

PUSH BC

PUSH DE

INC L

PUSH HL

CALL A18266

PUSH DE

LD A,(A16994)

LD C,A

LD B,0

PUSH BC

LD HL,A33662 ;screen buffer

CALL READ_VRAM

POP BC

POP DE

POP HL

EX (SP),HL

EX DE,HL

```

    PUSH    HL
    LD      HL,A33662    ;screen buffer
    CALL   WRITE_VRAM
    POP     DE
    POP     HL
    POP     BC
    DJNZ   A18146
    CALL   A17982
    LD      HL,A33662    ;screen buffer
    LD      A,(A16994)
    LD      C,A
    JP     WRITE_VRAM
;*****
A18197:
    LD      A,(A16955)
A18200:
    PUSH   HL
    LD      HL,(A17001)
    LD      B,A
    CALL   A18266
    PUSH   DE
    LD      HL,A16960
    LD      (HL),B
    LD      BC,1
    CALL   WRITE_VRAM
    POP    DE
    LD      A,(A17008)
    OR     A
    JR     NZ,A18244
    LD      A,D
    AND    15
    LD      D,A
    LD      A,(A17006)
    LD      HL,A16960
    XOR    (HL)
    LD      (HL),A
    LD      BC,1
    CALL   WRITE_VRAM

```

```

A18244:
    POP    HL
    RET
;*****
A18246:
    LD     HL,(A17001)
A18249:
    LD     A,255
    LD     (A17007),A
    CALL  A18266
    LD     BC,1
    LD     HL,A16955
    JP    READ_VRAM
;*****
A18266:
    LD     E,H
    LD     D,0
    LD     H,D
    ADD   HL,HL
    ADD   HL,HL
    CALL  A28000
    ADD   HL,DE
    LD     DE,(A16997)
    ADD   HL,DE
    EX    DE,HL
    LD     A,(A17007)
    AND   D
    LD     D,A
    RET
;*****
A18288:
    LD     L,A
    LD     H,0
    ADD   HL,HL
    ADD   HL,HL
    ADD   HL,HL
    LD     BC,(A16999)
    ADD   HL,BC

```

```

        EX      DE,HL
        LD      BC,8
        RET
;*****
;Control character matching table.
;
A18304:
        DB      8           ;backspace
        DB      13          ;CR
        DB      10          ;LF
        DB      12          ;^L
        DB      128         ;home
        DB      7           ;bell
        DB      22          ;^V erase to end of line
        DB      24          ;^X erase to bottom of
screen
        DB      28          ;quickmove cursor
        DB      160         ;up arrow
        DB      162         ;down arrow
        DB      163         ;left arrow
        DB      161         ;right arrow
        DB      9           ;htab
        DB      148         ;insert
        DB      151         ;delete
        DB      16          ;^P screen dump to printer
;*****
;Control character vector table.  Reverse order of
above table.
;
A18321:
        DW      A18379      ;18379  ^P
        DW      A17786      ;17786  delete
        DW      A17730      ;17730  insert
        DW      A17581      ;17581  htab
        DW      A17674      ;17674  right arrow
        DW      A17702      ;17702  left arrow
        DW      A17625      ;17625  down arrow
        DW      A17658      ;17658  up arrow

```

DW	A17905	;17905	quickmove cursor
DW	A17875	;17875	^X
DW	A17869	;17869	^V
DW	A17945	;17945	bell
DW	A17860	;17860	home
DW	A17842	;17842	^L
DW	A17625	;17625	LF
DW	A17611	;17611	CR
DW	A17702	;17702	backspace

;*****

A18355:

```
LD DE, (A17001)
RET
```

;*****

A18360:

```
LD DE, (A17001)
LD A, (A16956)
NEG
ADD A, D
LD D, A
LD A, (A16958)
NEG
ADD A, E
LD E, A
RET
```

;*****

A18379:

```
CALL A18197
LD HL, (A16995)
LD BC, (A16993)
LD A, B
LD B, C
LD C, A
```

A18392:

```
PUSH HL
PUSH BC
CALL A18266
LD B, 0
```

```

        PUSH    BC
        LD      HL,A33662    ;screen buffer
        CALL   READ_VRAM
        LD      A,(A16954)
        LD      HL,A33662    ;screen buffer
        POP    BC
        ADD    HL,BC
        DEC    HL
A18415:
        CPD
        JP     PO,A18422
        JR     Z,A18415
A18422:
        INC    HL
        INC    HL
        LD     (HL),3
        LD     HL,A33662    ;screen buffer
        XOR    A
        LD     (A31016),A
        LD     A,(A17988)
        LD     C,A
        CALL  A31049
        NOP
        NOP
        POP    BC
        POP    HL
        INC    L
        DJNZ  A18392
        CALL  A18197
        JP    A17498
;*****
A18453:
        PUSH  IY
        PUSH  IX
        LD   HL,A16953
        LD   A,127
        AND  (HL)
        LD   (HL),A

```

```
LD     HL,A16954
LD     A,127
AND    (HL)
LD     (HL),A
XOR    A
LD     (A17008),A
LD     (A17004),A
LD     (A17005),A
LD     (A17006),A
CALL   A17046
POP    IX
POP    IY
RET
```

```
;*****
```

```
A18492:
```

```
LD     A,1
LD     (A17008),A
XOR    A
LD     (A17005),A
LD     A,128
LD     (A17004),A
LD     HL,A16953
OR     (HL)
LD     (HL),A
LD     A,128
LD     HL,A16954
OR     (HL)
LD     (HL),A
PUSH   IY
PUSH   IX
LD     B,1
LD     C,128
CALL   WRITE_REGISTER
CALL   A18609
CALL   A18609
LD     BC,7683
LD     DE,276
LD     HL,6144
```

```

LD      A,48
EX      AF,AF'
LD      A,0
CALL    A17334
LD      B,1
LD      C,192
CALL    WRITE_REGISTER
POP     IX
POP     IY
RET
;*****
;Lo-res graphics (GR) blocks 6x4.
;
A18564:
        DB      0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH
        DB      0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H
        DB      0C0H,0C0H,0C0H,0C0H,0C0H,0C0H,0C0H,0C0H
;*****
;GR VRAM table allocation data.
;
A18588:
        DB      0
        DW      8064          ;sprite attribute table
        DB      1
        DW      14336         ;sprite generator table
        DB      2
        DW      6144          ;pattern name table
        DB      3
        DW      8192          ;pattern generator table
        DB      4
        DW      0             ;color table
        DB      255          ;end of table
;*****
;GR VDP register data.
;
A18604:
        DB      0,2          ;register 0, data 2
        DB      7,1          ;register 7, data 1

```



```

A18607      EQU    A18604+3

            DB    255          ;end of table
;*****
A18609:
            LD    HL,A18588
            CALL  A19315
            LD    HL,A18604
            CALL  A19334
            LD    A,1
            LD    (A17008),A
            LD    HL,0
            LD    DE,6144
            LD    A,17        ;

A18633      EQU    $-1        ;color

            CALL  FILL_VRAM
            NOP
            NOP
            NOP
            NOP
            NOP
            LD    HL,8200
            LD    B,20
            LD    C,10

A18649:
            PUSH  BC
            PUSH  HL
            EX    DE,HL
            LD    HL,A18564
            LD    BC,24
            CALL  WRITE_VRAM
            POP   HL
            LD    DE,24
            ADD   HL,DE
            POP   BC

```

```

DEC    C
JP     NZ,A18649
LD     DE,16
ADD    HL,DE
LD     C,10
DJNZ   A18649
CALL   A26272
LD     HL,6784
LD     DE,128
LD     A,160
CALL   FILL_VRAM
LD     HL,13568
LD     DE,768
LD     A,255
CALL   FILL_VRAM
LD     HL,5376
LD     DE,768
LD     A,241      ;

```

```

A18711      EQU    $-1      ;color

```

```

CALL   FILL_VRAM
LD     HL,32
LD     BC,96
LD     DE,13568
CALL   PUT_ASCII
RET

```

```

;*****

```

```

;GR coordinate checker.

```

```

;On exit, C=arg1, L=arg2.

```

```

;

```

```

A18728:

```

```

CALL   A1500      ;get arg1 0-255 in A and L
LD     A,39
CP     L          ;is it off the screen?
JP     C,A7936    ;YES, so Illegal Quantity

```

```

error

```

```

PUSH   HL        ;NO, so save it

```

```

    EXX
    DEC    C
    EXX
    INC    DE                ;skip token (,)
    CALL   A1500            ;get arg2 0-255 in A and L
    LD     A,39
    CP     L                ;is it off the screen?
    JP     C,A7936          ;YES, so Illegal Quantity
error
    POP    BC                ;NO, so restore arg1
    RET
;*****
;VLIN row1,row2 AT col/HLIN col1,col2 AT row
coordinate checker.
;On exit, B=arg2, C=arg1, A=L=arg3.
;
A18753:
    CALL   A18728            ;GR coordinate checker
    LD     A,L
    LD     H,A
    LD     L,C
    PUSH   HL                ;H=arg2, L=arg1
    EXX
    DEC    C
    EXX
    INC    DE                ;skip token (AT)
    CALL   A1500            ;get arg3 0-255 in A and L
    LD     A,39
    CP     L                ;is it off the screen?
    JP     C,A7936          ;YES, so Illegal Quantity
error
    LD     A,L                ;NO, so A=arg3
    POP    BC                ;restore B=arg2, C=arg1
    RET
;*****
A18776:
    LD     A,(A17988)
    CP     40

```

```

        JR      Z,A18788
        LD      A,C
        LD      DE,16
        RET
;*****
A18788:
        LD      B,7
        CALL   WRITE_REGISTER
        POP     HL
        JP      A17136
;*****
;PLOT/HLIN vector table.
;
A18797:
        DW      A18885
        DW      A18891
        DW      A18904
        DW      A18922
;*****
A18805:
        LD      A,(A17008)
        DEC     A
        RET     NZ
        PUSH   IX
        PUSH   IY
        LD      A,C
        LD      C,B
        LD      B,A
        LD      D,E
        LD      A,C
        SUB     B
        JP      P,A18828
        LD      E,C
        LD      C,B
        LD      B,E
        NEG
A18828:
        LD      E,B

```

```
LD      B,A
INC     B
PUSH   BC
LD      B,E
LD      C,D
CALL   A19348
POP     BC
LD      C,A
PUSH   BC
```

A18840:

```
LD      HL,A18797
RLCA
LD      C,A
LD      B,0
ADD    HL,BC
PUSH   DE
LD      E,(HL)
INC    HL
LD      D,(HL)
EX     DE,HL
POP    DE
PUSH   DE
JP     (HL)
```

;*****

A18856:

```
POP    DE
POP    BC
DEC    B
JR     Z,A18880
INC    C
PUSH   BC
LD     A,C
CP     4
JR     NZ,A18840
POP    BC
LD     C,0
PUSH   BC
LD     A,C
```

```

        LD     HL,24
        ADD   HL,DE
        EX    DE,HL
        JR    A18840
;*****
A18880:
        POP   IY
        POP   IX
        RET
;*****
A18885:
        CALL  A19376
        JP    A18856
;*****
A18891:
        CALL  A19423
        LD    HL,8
        ADD   HL,DE
        EX    DE,HL
        CALL  A19376
        JR    A18856
;*****
A18904:
        LD    HL,8
        ADD   HL,DE
        EX    DE,HL
        CALL  A19423
        LD    HL,8
        ADD   HL,DE
        EX    DE,HL
        CALL  A19376
        JR    A18856
;*****
A18922:
        LD    HL,16
        ADD   HL,DE
        EX    DE,HL
        CALL  A19423

```

```

        JR      A18856
;*****
;PLOT/VLIN vector table.
;
A18932:
        DW      A18993
        DW      A19006
        DW      A19029
        DW      A19057
;*****
A18940:
        LD      A, (A17008)
        DEC     A
        RET     NZ
        PUSH    IX
        PUSH    IY
        LD      A, C
        LD      C, B
        LD      B, A
        LD      D, E
        LD      A, C
        SUB     B
        JP      P, A18963
        LD      E, C
        LD      C, B
        LD      B, E
        NEG
A18963:
        LD      E, B
        LD      C, B
        LD      B, A
        INC     B
        PUSH    BC
        LD      B, D
        LD      C, E
        CALL    A19348
        LD      HL, A18932
        RLCA

```

```
LD      C,A
LD      B,0
ADD     HL,BC
PUSH   DE
LD      E,(HL)
INC     HL
LD      D,(HL)
EX      DE,HL
POP     DE
JP      (HL)
```

```
;*****
```

```
A18988:
```

```
POP     IY
POP     IX
RET
```

```
;*****
```

```
A18993:
```

```
CALL   A19376
POP     BC
DEC     B
JR      Z,A18988
CALL   A19075
PUSH   BC
JR      A18993
```

```
;*****
```

```
A19006:
```

```
CALL   A19423
PUSH   DE
LD      HL,8
ADD     HL,DE
EX      DE,HL
CALL   A19376
POP     DE
POP     BC
DEC     B
JR      Z,A18988
CALL   A19075
PUSH   BC
```



```

        JR      A19006
;*****
A19029:
        LD      HL,8
        ADD     HL,DE
        EX      DE,HL
A19034:
        CALL    A19423
        PUSH    DE
        LD      HL,8
        ADD     HL,DE
        EX      DE,HL
        CALL    A19376
        POP     DE
        POP     BC
        DEC     B
        JR      Z,A18988
        CALL    A19075
        PUSH    BC
        JR      A19034
;*****
A19057:
        LD      HL,16
        ADD     HL,DE
        EX      DE,HL
A19062:
        CALL    A19423
        POP     BC
        DEC     B
        JR      Z,A18988
        CALL    A19075
        PUSH    BC
        JR      A19062
;*****
A19075:
        BIT     0,C
        JR      NZ,A19086
        SET     0,C

```

```

        LD     HL,4
        JR     A19091
;*****
A19086:
        LD     HL,252
        RES   0,C
A19091:
        ADD   HL,DE
        EX    DE,HL
        RET
;*****
;PLOT vector table.
;
A19094:
        DW    A19137
        DW    A19143
        DW    A19157
        DW    A19176
;*****
A19102:
        CALL  A29186
        NOP
        NOP
        PUSH  IX
        PUSH  IY
        LD   A,B
        LD   B,C
        LD   C,A
        CALL A19348
        LD   HL,A19094
        RLCA
        LD   C,A
        LD   B,0
        ADD  HL,BC
        PUSH DE
        LD   E,(HL)
        INC  HL
        LD   D,(HL)

```

```

        EX      DE,HL
        POP     DE
        JP      (HL)
;*****
A19132:
        POP     IY
        POP     IX
        RET
;*****
A19137:
        CALL    A19376
        JP      A19132
;*****
A19143:
        CALL    A19423
        LD      HL,8
        ADD     HL,DE
        EX     DE,HL
        CALL    A19376
        JP      A19132
;*****
A19157:
        LD      HL,8
        ADD     HL,DE
        EX     DE,HL
        CALL    A19423
        LD      HL,8
        ADD     HL,DE
        EX     DE,HL
        CALL    A19376
        JP      A19132
;*****
A19176:
        LD      HL,16
        ADD     HL,DE
        EX     DE,HL
        CALL    A19423
        JP      A19132

```

```

;*****
;SCRN(x,y) function vector table.
;
A19187:
    DW    A19225        ;read foreground color and
exit
    DW    A19231        ;read background color and
exit
    DW    A19237        ;
    DW    A19248        ;
;*****
;SCRN(x,y) function.
;On entry, (B,C)=(x,y) coordinates of a GR PLOT
block.
;On exit, A=block color.
;
A19195:
    CALL  A29186        ;check for GR mode only
    NOP
    NOP
    PUSH  IX
    PUSH  IY
    LD    A,B          ;A=x
    LD    B,C          ;B=y
    LD    C,A          ;C=x
    CALL  A19348        ;calculate GR offsets
A19210:
    LD    HL,A19187    ;SCRN(x,y) vector table
    RLCA
    LD    C,A
    LD    B,0
    ADD   HL,BC
    PUSH DE
    LD    E,(HL)
    INC  HL
    LD    D,(HL)
    EX   DE,HL
    POP  DE

```

```

        JP      (HL)          ;go to vector
;*****
;Read foreground color and exit SCRNX,Y function.
;
A19225:
        CALL   A19264        ;read foreground color
        JP     A19256
;*****
;Read background color and exit SCRNX,Y function.
;
A19231:
        CALL   A19287        ;read background color
        JP     A19256
;*****
;
;
A19237:
        LD     HL,16
        ADD    HL,DE
        EX     DE,HL
        CALL   A19264        ;read foreground color
        JP     A19256
;*****
;
;
A19248:
        LD     HL,16
        ADD    HL,DE
        EX     DE,HL
        CALL   A19287        ;read background color
A19256:
        NOP
        NOP
        NOP
        POP    IY
        POP    IX
        RET
;*****

```

;Read foreground color.
;On entry, DE=VRAM color table address.
;On exit, A=hi nibble of VRAM data (foreground color).

;

A19264:

```
PUSH  DE           ;save VRAM address
LD    HL,A16759    ;address to store data
LD    BC,1         ;1 byte to read
CALL  READ_VRAM    ;EOS read VRAM
LD    A,(A16759)   ;get it
SRL   A
SRL   A
SRL   A
SRL   A           ;get hi nibble into A
POP   DE          ;restore VRAM address
RET
```

;*****

;Read background color.
;On entry, DE=VRAM color table address.
;On exit, A=lo nibble of VRAM data (background color).

;

A19287:

```
PUSH  DE           ;save VRAM address
LD    HL,A16759    ;address to store data
LD    BC,1         ;1 byte to read
CALL  READ_VRAM    ;EOS read VRAM
LD    A,(A16759)   ;get it
AND   15           ;wipe out hi nibble
POP   DE          ;restore VRAM address
RET
```

;*****

;HOME and CLS commands.
;Actual entry is at A11090.

;

A19304:

```
LD    A,(A17008)   ;get screen mode
```

```

        CP      3           ;is it HGR2?
        RET     Z           ;YES, so don't do it (no
TEXT window to clear)
        LD      A,12       ;^L
        JP      A17242     ;print character in A on
screen
;*****
;Load VDP registers with VRAM table adresses.
;Requires table in format:  table code, address,
[...], 255 (end of table).
;
A19315:
        LD      A, (HL)
        CP      255
        RET     Z
        INC     HL
        LD      E, (HL)
        INC     HL
        LD      D, (HL)
        INC     HL
        EX      DE,HL
        PUSH   DE
        CALL   INIT_TABLE
        POP    DE
        EX      DE,HL
        JP      A19315
;*****
;Load VDP registers with data.
;Requires table in format:  register number, data,
[...], 255 (end of table).
;
A19334:
        LD      A, (HL)
        CP      255
        RET     Z
        LD      B,A
        INC     HL
        LD      C, (HL)

```

```

        INC     HL
        CALL   WRITE_REGISTER
        JP     A19334
;*****
;Calculate GR offsets.
;On entry, (B,C)=(x,y).   @@@@ ?? @@@@
;On exit, E=(24x+8), D=(y/2), and A=block type 0-3.
  @@@@ ?? @@@@
;
A19348:
        LD     D,B
        SRA   B           ;/2
        SRA   B           ;/4
        LD     A,B       ;effectively INT(B/4)
        ADD   A,A       ;*2
        ADD   A,A       ;*4
        ADD   A,A       ;*8
        LD     B,A       ;effectively 8*INT(B/4)
        ADD   A,A       ;*16
        ADD   A,B       ;+8
        ADD   A,8       ;effectively
(16*INT(B/4))+(8*INT(B/4))+8
;                               =
(24*INT(B/4))+8
        LD     B,C
        SRA   C           ;/2
        LD     H,C       ;save it; was it odd? (bit 0
shifted into CF)
        JR     NC,A19370 ;NO,
        ADD   A,4       ;YES, so add 4
A19370:
        LD     L,A       ;into L
        LD     A,D       ;original B=x
        AND   3         ;make into 0-3
        EX    DE,HL     ;
        RET
;*****
A19376:

```



```
PUSH DE
LD HL,A16759
LD BC,1
CALL READ_VRAM
LD A,(A16776)
RLCA
RLCA
RLCA
RLCA
LD C,A
LD A,15
LD HL,A16759
AND (HL)
OR C
LD (HL),A
```

A19402:

```
LD DE,A16760
LD BC,3
LDIR
POP DE
PUSH DE
LD HL,A16759
LD BC,4
CALL WRITE_VRAM
POP DE
RET
```

;*****

A19423:

```
PUSH DE
LD HL,A16759
LD BC,1
CALL READ_VRAM
LD A,(A16776)
LD C,A
LD A,240
LD HL,A16759
AND (HL)
OR C
```

```

        LD      (HL),A
        JP      A19402
;*****
A19448:
        LD      HL,(A16201)
        PUSH   HL
        LD      HL,A17234
        LD      (A16201),HL
        LD      HL,A17016
        CALL   A12110
        POP    HL
        LD      (A16201),HL
        JP      A12128
;*****
A19471:
        LD      HL,A17014
        BIT    2,(HL)
        JR     NZ,A19553
        CALL   A28487
        NOP
A19482:
        LD      C,A
        LD      A,(A17008)
        CP     3
        RET    Z
        LD      A,(A16136)
        OR     A
        JR     Z,A19510
        CALL   A17310
        INC    A
        JR     NZ,A19534
        LD      A,(A17003)
        LD      HL,A16135
        XOR    (HL)
        JR     NZ,A19520
A19510:
        DEC    A
        LD      (A16136),A

```

```

A19514:
    CALL    A17310
    INC     A
    JR      NZ,A19514
A19520:
    LD      A,(A17003)
    LD      HL,A16134
    XOR     (HL)
    JR      NZ,A19534
    LD      A,19
    JP      A7950
;*****
A19534:
    LD      A,C
    JP      A17242
;*****
A19538:
    PUSH   DE
    CALL   A18360
    LD     A,D
    POP   DE
    OR    A
    LD    A,4
    JR    NZ,A19482
    LD    (A17014),A
    RET
;*****
A19553:
    CP    13
    JR    Z,A19575
    LD    HL,(A17040)
    LD    (HL),A
    INC  HL
    LD    (A17040),HL
    LD    BC,A17039
    OR   A
    SBC  HL,BC
    RET  NZ

```

```

        JP      A24327
;*****
A19575:
        LD      HL, (A17040)
        XOR     A
        LD      (A17014), A
        LD      (HL), A
        PUSH   DE
        LD      DE, A17017
        SBC    HL, DE
        LD      A, L
        DEC    DE
        LD      (DE), A
        INC    DE
        LD      (A17040), DE
        LD      HL, (A16829)
        CALL   A19692
        CALL   A11777
        JP     Z, A19653
        LD     A, (DE)
        AND    223
        LD     HL, A20123
        LD     BC, 15
        CPIR
        JP     NZ, A19708
        CALL  A13659
        JP     NC, A19708
        LD     HL, A20138
A19632:
        LD     B, (HL)
        LD     A, B
        OR    A
        JR    Z, A19708
        INC  HL
        CALL  A11851
        JR    Z, A19667
        INC  HL
        LD     A, B

```

```

        ADD     A,L
        LD      L,A
        JR      NC,A19632
        INC     H
        JP      A19632
;*****
A19653:
        POP     DE
        LD      HL,(A16195)
        LD      (A16197),HL
        LD      HL,(A16199)
        LD      (A16201),HL
        RET
;*****
A19667:
        LD      A,B
        ADD     A,L
        LD      L,A
        JR      NC,A19673
        INC     H
A19673:
        CALL   A14549
        CALL   A11777
        PUSH   AF
        LD      C,(HL)
        LD      B,0
        LD      HL,A20339
        ADD    HL,BC
        LD      A,(HL)
        INC     HL
        LD      H,(HL)
        LD      L,A
        POP     AF
A19692:
        JP      (HL)
;*****
;^D CATALOG.

```

A19693:
CALL A21144
JP NC,A19708

A19699:
POP DE
RET

;*****
;^D DELETE.

A19701:
JR Z,A19708
CALL A20426

A19706:
JR C,A19699

A19708:
POP DE
JP A24351

;*****
;^D RENAME.

A19712:
JR Z,A19708
CALL A20469
JR A19706

;*****
;^D LOCK.

A19719:
JR Z,A19708
CALL A20641
JR A19706

;*****
;^D UNLOCK.

A19726:
JR Z,A19708
CALL A20640
JR A19706

```
;*****  
;^D BSAVE.
```

A19733:

```
    JR      Z,A19708  
    CALL   A20849  
    JR      A19706
```

```
;*****  
;^D BLOAD.
```

A19740:

```
    JR      Z,A19708  
    CALL   A20993  
    JR      A19706
```

```
;*****  
;^D BRUN.
```

A19747:

```
    JR      Z,A19708  
    CALL   A21140  
    JR      A19706
```

```
;*****  
;^D CLOSE.
```

A19754:

```
    JR      Z,A19708  
    CALL   A24612  
    JR      A19706
```

```
;*****  
;^D MON.
```

A19761:

```
    JR      Z,A19708  
    CALL   A23047  
    JR      A19706
```

```
;*****  
;^D NOMON.
```

A19768:
JR Z,A19708
CALL A23042
JR A19706
;*****
;^D LOAD.

A19775:
JR Z,A19708
CALL A23976
JR A19706
;*****
;^D SAVE.

A19782:
JR Z,A19708
CALL A23813
JR A19706
;*****
;^D OPEN.

A19789:
JR Z,A19708
CALL A24497
JR A19706
;*****
;^D APPEND.

A19796:
JR Z,A19708
CALL A21477
JR A19706
;*****
;^D WRITE.

A19803:
JR Z,A19708
CALL A22455

JR A19706
;*****
;^D READ.

A19810:

JR Z,A19708
CALL A22049
JR A19706
;*****
;^D POSITION.

A19817:

JR Z,A19708
CALL A21715
JR A19706
;*****
;^D PR.

A19824:

JR Z,A19708
CALL A24934
JR A19706
;*****
;^D IN.

A19831:

JR Z,A19708
CALL A24943
JP A19706
;*****
;^D FP.

A19839:

JP NZ,A19708
CALL A20419
JP A19706
;*****
;^D INT.

A19848:

JP NZ,A19708
CALL A20416
JP A19706

;*****

;^D INIT.

A19857:

JP Z,A19708
CALL A25267
JP A19706

;*****

;^D RUN.

A19866:

JP Z,A19708
CALL A24012
JP A19706

;*****

;^D RECOVER.

A19875:

JP Z,A19708
CALL A20532
JP A19706

;*****

A19884:

PUSH DE
LD DE,A16247
CALL A11777
JP Z,A19969
LD A,(DE)
AND 95
LD HL,A20127
LD BC,11
CPIR
JP NZ,A19969

```
CALL A13659
JP NC,A19969
LD HL,A20183
```

A19917:

```
LD B,(HL)
LD A,B
OR A
JR Z,A19969
INC HL
CALL A11851
JR Z,A19939
INC HL
LD A,B
ADD A,L
LD L,A
JP NC,A19917
INC H
JP A19917
```

;*****

A19939:

```
LD A,B
ADD A,L
LD L,A
JR NC,A19945
INC H
```

A19945:

```
CALL A14549
CALL A11777
PUSH AF
LD C,(HL)
LD B,0
LD HL,A20289
ADD HL,BC
LD A,(HL)
INC HL
LD H,(HL)
LD L,A
POP AF
```

```

        EXX
        RES    0,B
        EXX
        JP     (HL)
;*****
A19969:
        POP    DE
        RET
;*****
;Immediate mode CATALOG.

A19971:
        CALL   A21144
        JR     NC,A19990
A19976:
        POP    DE
        POP    HL
        POP    AF
        POP    HL
        JP     A16035
;*****
;Immediate mode DELETE.

A19983:
        JR     Z,A19990
        CALL   A20426
A19988:
        JR     C,A19976
A19990:
        POP    HL
        POP    HL
        POP    AF
        EX     (SP),HL
        LD     HL,A20389
        EX     (SP),HL
        OR     A
        PUSH   AF
        JP     (HL)

```

```
;*****  
;Immediate mode RENAME.
```

A20001:

```
    JR     Z,A19990  
    CALL  A20469  
    JR     A19988
```

```
;*****  
;Immediate mode LOCK.
```

A20008:

```
    JR     Z,A19990  
    CALL  A20641  
    JR     A19988
```

```
;*****  
;Immediate mode UNLOCK.
```

A20015:

```
    JR     Z,A19990  
    CALL  A20640  
    JR     A19988
```

```
;*****  
;Immediate mode BSAVE.
```

A20022:

```
    JR     Z,A19990  
    CALL  A20849  
    JR     A19988
```

```
;*****  
;Immediate mode BLOAD.
```

A20029:

```
    JR     Z,A19990  
    CALL  A20993  
    JR     A19988
```

```
;*****  
;Immediate mode BRUN.
```

A20036:
JR Z,A19990
CALL A21140
JR A19988
;*****
;Immediate mode CLOSE.

A20043:
JR Z,A19990
CALL A24612
JR A19988
;*****
;Immediate mode MON.

A20050:
JR Z,A19990
CALL A23047
JR A19988
;*****
;Immediate mode NOMON.

A20057:
JR Z,A19990
CALL A23042
JR A19988
;*****
;Immediate mode LOAD.

A20064:
JR Z,A19990
CALL A23976
JR A19988
;*****
;Immediate mode SAVE.

A20071:
JR Z,A19990
CALL A23813

```
        JR      A19988
;*****
;Immediate mode FP.
```

A20078:

```
        JP      NZ,A19990
        CALL    A20419
        JP      A19988
;*****
;Immediate mode INT.
```

A20087:

```
        JP      NZ,A19990
        CALL    A20416
        JP      A19988
;*****
;Immediate mode INIT.
```

A20096:

```
        JP      Z,A19990
        CALL    A25267
        JP      A19988
;*****
;Immediate mode RUN.
```

A20105:

```
        JP      Z,A19990
        CALL    A24012
        JP      A19988
;*****
;Immediate mode RECOVER.
```

A20114:

```
        JP      Z,A19990
        CALL    A20532
        JP      A19988
;*****
;First letters of I/O commands.
```

```

;
A20123:
    DB      "OAWP "
A20127:
    DB      "CDRLUBNMSIF "
;*****
;^D-only I/O commands:  length, word, offset into
vector table.
;
A20138:
    DB      4
    DB      "OPEN"
    DB      0
    DB      6
    DB      "APPEND"
    DB      2
    DB      4
    DB      "READ"
    DB      4
    DB      5
    DB      "WRITE"
    DB      6
    DB      8
    DB      "POSITION"
    DB      8
    DB      2
    DB      "PR"
    DB      10
    DB      2
    DB      "IN"
    DB      12
;*****
;^D or immediate I/O commands:  length, word, offset
into vector table.
;
A20183:
    DB      7
    DB      "CATALOG"

```


DB	14
DB	6
DB	"DELETE "
DB	16
DB	6
DB	"RENAME "
DB	18
DB	4
DB	"LOCK "
DB	20
DB	6
DB	"UNLOCK "
DB	22
DB	5
DB	"BSAVE "
DB	24
DB	5
DB	"BLOAD "
DB	26
DB	4
DB	"BRUN "
DB	28
DB	5
DB	"CLOSE "
DB	30
DB	3
DB	"MON "
DB	32
DB	5
DB	"NOMON "
DB	34
DB	4
DB	"LOAD "
DB	36
DB	4
DB	"SAVE "
DB	38
DB	2

```
DB      "FP"  
DB      40  
DB      3  
DB      "INT"  
DB      42  
DB      4  
DB      "INIT"  
DB      44  
DB      3
```

A20289:

```
DB      "RUN"  
DB      46  
DB      7  
DB      "RECOVER"  
DB      48  
DB      0          ;end of table
```

;*****

;Immediate mode I/O command vector table.

;

A20303:

```
DW      A19971      ;CATALOG  
DW      A19983      ;DELETE  
DW      A20001      ;RENAME  
DW      A20008      ;LOCK  
DW      A20015      ;UNLOCK  
DW      A20022      ;BSAVE  
DW      A20029      ;BLOAD  
DW      A20036      ;BRUN  
DW      A20043      ;CLOSE  
DW      A20050      ;MON  
DW      A20057      ;NOMON  
DW      A20064      ;LOAD  
DW      A20071      ;SAVE  
DW      A20078      ;FP  
DW      A20087      ;INT  
DW      A20096      ;INIT  
DW      A20105      ;RUN  
DW      A20114      ;RECOVER
```

```
;*****  
;^D I/O command vector table.  
;
```

```
A20339:
```

```
    DW    A19789    ;OPEN  
    DW    A19796    ;APPEND  
    DW    A19810    ;READ  
    DW    A19803    ;WRITE  
    DW    A19817    ;POSITION  
    DW    A19824    ;PR  
    DW    A19831    ;IN  
    DW    A19693    ;CATALOG  
    DW    A19701    ;DELETE  
    DW    A19712    ;RENAME  
    DW    A19719    ;LOCK  
    DW    A19726    ;UNLOCK  
    DW    A19733    ;BSAVE  
    DW    A19740    ;BLOAD  
    DW    A19747    ;BRUN  
    DW    A19754    ;CLOSE  
    DW    A19761    ;MON  
    DW    A19768    ;NOMON  
    DW    A19775    ;LOAD  
    DW    A19782    ;SAVE  
    DW    A19839    ;FP  
    DW    A19848    ;INT  
    DW    A19857    ;INIT  
    DW    A19866    ;RUN  
    DW    A19875    ;RECOVER
```

```
;*****
```

```
A20389:
```

```
    DB    26  
    DB    "Illegal Form Of OS Command"
```

```
;*****
```

```
;INT (20416) and FP (20419). Instead of invoking  
integer or floating-point  
;BASICS, these routines merely change the prompt to  
either > (INT) or ] (FP).
```

```

;
A20416:
    LD     A,62           ;INT prompt >
    DB     33             ;LD HL,23870 ;garbage
                                ;I hate overlapping code!

A20419:
    LD     A,93           ;FP prompt ] in this frame
    LD     (A1146),A
    SCF
    RET

;*****
A20426:
    CALL  A23324
    RET   NC
    CALL  A23364
    RET   NC
    LD    A,65
    CALL  A23494
    LD    A,(A16821)
    LD    HL,A16797
    CALL  _DELETE_FILE
    SCF
    RET   Z
    LD    A,72
    CALL  A23494
    LD    A,(A16821)
    LD    HL,A16797
    CALL  _DELETE_FILE
    SCF
    RET   Z
    JP    A24297

;*****
A20469:
    CALL  A23324
    RET   NC
    CALL  A23315
    RET   NC
    CALL  A23364

```

```
RET    NC
LD     A,65
CALL   A23494
LD     A,65
CALL   A23489
LD     A,(A16821)
LD     DE,A16797
LD     HL,A16809
CALL   _RENAME_FILE
SCF
RET    Z
LD     A,72
CALL   A23494
LD     A,72
CALL   A23489
LD     A,(A16821)
LD     DE,A16797
LD     HL,A16809
CALL   _RENAME_FILE
SCF
RET    Z
JP     A24297
```

;*****

A20532:

```
CALL   A23324
RET    NC
CALL   A23364
RET    NC
LD     A,65
CALL   A23494
LD     A,(A16821)
LD     DE,A16797
LD     HL,A16885
CALL   _QUERY_FILE
SCF
RET    Z
LD     A,72
CALL   A23494
```

```

LD      A, (A16821)
LD      DE, A16797
LD      HL, A16885
CALL    _QUERY_FILE
SCF
RET     Z
LD      BC, 12
LD      DE, A16809
LD      HL, A16797
LDIR
LD      A, 97
CALL    A23494
LD      A, 65
CALL    A23489
LD      A, (A16821)
LD      DE, A16797
LD      HL, A16809
CALL    _RENAME_FILE
SCF
RET     Z
LD      A, 104
CALL    A23494
LD      A, 104
CALL    A23489
LD      A, (A16821)
LD      DE, A16797
LD      HL, A16809
CALL    _RENAME_FILE
SCF
RET     Z
JP      A24297

```

```
;*****
```

```
A20640:
```

```

DB      230          ;AND 55 ;clear CF
                        ;I hate overlapping code!

```

```
A20641:
```

```

SCF          ;set CF in this frame
PUSH    AF

```

```
CALL A23324
POP BC
RET NC
PUSH BC
CALL A23364
POP BC
RET NC
PUSH BC
LD A,65
CALL A23494
LD HL,A16885
LD DE,A16797
LD A,(A16821)
CALL _QUERY_FILE
JP Z,A20695
LD A,72
CALL A23494
LD HL,A16885
LD DE,A16797
LD A,(A16821)
CALL _QUERY_FILE
JP NZ,A24297
```

A20695:

```
LD A,(A16897)
RLCA
RLCA
LD B,A
POP AF
BIT 3,B
JR NZ,A20725
LD A,B
RRA
SRA A
LD (A16897),A
LD HL,A16885
LD DE,A16797
LD A,(A16821)
CALL _SET_FILE
```

```
A20725:
    SCF
    RET    Z
    JP     A24297
```

```
;*****
```

```
A20730:
    LD     B,76
    DB     33           ;LD HL,16646  ;garbage code
for B-cascade
```

```
;I hate overlapping code!
```

```
A20733:
    LD     B,65
```

```
A20735:
    CALL  A11777
    RET   Z
    CP   44
    JP   NZ,A20991
    INC  DE
    CALL A11777
    RET  Z
    AND  223
    XOR  B
    RET  NZ
    INC  DE
```

```
A20754:
    CALL  A11777
    RET   Z
    CP   36
    JR   NZ,A20797
    INC  DE
    CALL A11777
    RET  Z
    CALL A20825
    RET  NC
    LD   HL,0
```

```
A20774:
    ADD  A,L
    LD   L,A
```



```
INC    DE
CALL  A20825
JR    NC,A20806
ADD   HL,HL
JR    C,A20795
ADD   HL,HL
JR    C,A20795
ADD   HL,HL
JR    C,A20795
ADD   HL,HL
JP    NC,A20774
```

A20795:

```
CCF
RET
```

;*****

A20797:

```
CALL  A11885
RET   NC
PUSH  BC
CALL  A12451
POP   BC
```

A20806:

```
LD    A,B
CP    65
PUSH  DE
LD    DE,A16791
JR    Z,A20818
LD    DE,A16793
```

A20818:

```
EX    DE,HL
LD    (HL),E
INC   HL
LD    (HL),D
POP   DE
SCF
RET
```

;*****

A20825:

```
LD      A, (DE)
OR      A
RET     Z
CALL   A11885
RLA
AND     191
SUB     32
RRA
RET     C
SUB     39
CCF
RET     NC
CP      10
CCF
RET     NC
CP      16
RET
```

```
;*****
```

```
A20849:
```

```
CALL   A23324
RET     NC
CALL   A20733
RET     NC
CALL   A20730
RET     NC
CALL   A23364
RET     NC
CALL   A23504
LD      (A17044), HL
LD      A, (A16821)
CALL   _DELETE_FILE
LD      HL, (A16793)
LD      BC, 5
ADD     HL, BC
EX      DE, HL
LD      BC, 0
LD      A, (A16821)
LD      HL, (A17044)
```

```
CALL  _MAKE_FILE
JP    NZ,A24297
LD    A,(A16821)
LD    B,2
LD    HL,(A17044)
CALL  _OPEN_FILE
JP    NZ,A24297
LD    (A16823),A
LD    B,255
LD    HL,(A17044)
CALL  A24980
LD    HL,1
LD    (A16788),HL
LD    A,2
LD    (A16790),A
LD    A,(A16823)
LD    BC,5
LD    HL,A16788
CALL  _WRITE_FILE
JP    NZ,A24297
LD    A,(A16823)
LD    BC,(A16793)
LD    HL,(A16791)
CALL  _WRITE_FILE
JP    NZ,A24297
LD    A,(A16823)
CALL  _CLOSE_FILE
JP    NZ,A24297
LD    A,(A16823)
LD    B,A
CALL  A25087
LD    A,72
CALL  A23635
SCF
RET
```

```
;*****
```

```
A20990:
```

```
POP  AF
```

```

A20991:
    OR      A
    RET
;*****
A20993:
    CALL   A23324
    RET    NC
    CALL   A11777
    JR     Z,A21031
    CP     44
    JP     NZ,A20991
    PUSH   DE
    INC    DE
    CALL   A11777
    POP    DE
    RET    Z
    AND    223
    CP     65
    JR     NZ,A20991
    CALL   A20733
    RET    NC
    LD     HL,(A16791)
    LD     (A16793),HL
    DB     230           ;AND 55 ;clear CF in this
frame
                                ;I hate overlapping code!
A21031:
    SCF                                ;set CF in this frame
    PUSH   AF
A21033:
    CALL   A23364
    JR     NC,A20990
    LD     A,72
    CALL   A23494
    LD     A,(A16821)
    LD     B,1
    LD     HL,A16797
    CALL   _OPEN_FILE

```

```
JP      NZ,A24297
LD      (A16822),A
LD      B,255
LD      HL,A16797
CALL    A24980
LD      A,(A16822)
LD      BC,5
LD      HL,A16788
CALL    _READ_FILE
JP      NZ,A24297
LD      A,(A16790)
CP      2
JP      NZ,A24357
POP     AF
JR      C,A21100
LD      HL,(A16793)
LD      (A16791),HL
```

A21100:

```
LD      A,(A16822)
LD      BC,65535
LD      HL,(A16791)
CALL    _READ_FILE
JR      Z,A21119
CP      10
JP      NZ,A24297
```

A21119:

```
LD      A,(A16822)
CALL    _CLOSE_FILE
LD      A,(A16822)
LD      B,A
CALL    A25087
LD      HL,(A16791)
SCF
RET     Z
JP      A24297
```

;*****

A21140:

```
CALL    A20993
```

```

        JP      (HL)
;*****
A21144:
        LD      HL,0
        LD      (A16793),HL
        CALL   A23364
        RET    NC
        LD      A,(A16821)
        LD      HL,A21444
        LD      B,1
        CALL   _OPEN_FILE
        JP      NZ,A24297
        LD      B,255
        LD      HL,A21444
        CALL   A24980
        LD      (A16822),A
        LD      HL,A16885
        LD      BC,26
        CALL   _READ_FILE
        JP      NZ,A24297
        CALL   A21331
        LD      A,(A16822)
        LD      HL,A16885
        LD      BC,26
        CALL   _READ_FILE
        JP      NZ,A24297
        LD      B,37
A21211:
        PUSH   BC
        LD      A,(A16822)
        LD      HL,A16885
        LD      BC,26
        CALL   _READ_FILE
        JR      NZ,A21280
        LD      HL,A16897
        BIT    0,(HL)
        JR      NZ,A21287
        CP      16

```

```
JR      Z,A21325
BIT     3,(HL)
JR      NZ,A21261
BIT     2,(HL)
JR      Z,A21258
LD      DE,(A16902)
LD      HL,(A16793)
ADD     HL,DE
LD      (A16793),HL
JR      A21261
```

```
;*****
```

```
A21258:
```

```
CALL   A21352
```

```
A21261:
```

```
POP    BC
DJNZ   A21211
LD     A,(A16822)
LD     HL,A16885
LD     BC,10
CALL   _READ_FILE
LD     B,39
JR     A21211
```

```
;*****
```

```
A21280:
```

```
POP    BC
CALL   A21312
JP     A24297
```

```
;*****
```

```
A21287:
```

```
CALL   A12128
LD     DE,(A16902)
LD     HL,(A16793)
ADD    HL,DE
EX     DE,HL
CALL   A12967
LD     HL,A21463
CALL   A12110
CALL   A12128
```

```

A21311:
    POP    BC
A21312:
    LD     A, (A16822)
    PUSH  AF
    CALL  _CLOSE_FILE
    POP   BC
    CALL  A25087
    SCF
    RET
;*****
A21325:
    CALL  A21311
    JP    A24336
;*****
A21331:
    LD     HL, A16885
    CALL  A21413
    LD     HL, A21454
    CALL  A12110
    LD     HL, A16912
    CALL  A21388
    JP    A12128
;*****
A21352:
    CALL  A21408
    CALL  A21400
    LD     HL, A16885
    CALL  A21413
    PUSH  DE
    CALL  A11994
    LD     DE, (A16902)
    LD     A, 32
    CALL  A12968
    CALL  A21408
    POP   DE
    EX    DE, HL
    DEC  HL

```



```

        LD     (HL), 32
        LD     HL, A16912
A21388:
        LD     B, 12
A21390:
        LD     A, (HL)
        INC   HL
        CALL  A11994
        DJNZ  A21390
        JP    A12128
;*****
A21400:
        LD     A, (A16897)
        RLCA
        LD     A, 42
        JR    C, A21410
A21408:
        LD     A, 32
A21410:
        JP    A11994
;*****
A21413:
        PUSH  HL
        LD     DE, A16912
        LD     B, 12
A21419:
        LD     A, 32
        LD     (DE), A
        INC   DE
        DJNZ  A21419
        LD     DE, A16912
        LD     B, 12
A21430:
        LD     A, (HL)
        CP    3
        JR    Z, A21440
        LD     (DE), A
        INC   DE

```

```

        INC     HL
        DJNZ   A21430
A21440:
        DEC     HL
        LD      A, (HL)
        POP    HL
        RET
;*****
A21444:
        DB     "DIRECTORY"
        DB     3
A21454:
        DB     8
        DB     "Volume: "
A21463:
        DB     13
        DB     " Blocks Free"
        DB     13
;*****
A21477:
        CALL   A23324
        RET    NC
        CALL   A23364
        RET    NC
        LD     A, 65
        CALL   A23494
        LD     DE, A16939
        LD     HL, A16797
        LD     BC, 12
        LDIR
        EX     DE, HL
        LD     (HL), C
        INC   HL
        LD     (HL), C
        LD     HL, A16797
        CALL   A25116
        JR     NC, A21560
        PUSH  HL

```

```

    PUSH    DE
    LD      A, (DE)
    PUSH    AF
    LD      HL, A16797
    LD      A, (A16821)
    CALL    _CHECK_FCB
    POP     BC
    POP     DE
    POP     HL
    JR      NZ, A21556
    PUSH    DE
    PUSH    BC
    LD      DE, A16951
    LDI
    LDI
    CALL    A24702
    POP     AF
    POP     DE
    LD      (DE), A
    LD      HL, A16951
    LD      A, (HL)
    INC     HL
    OR      (HL)
    LD      B, 7
    JR      NZ, A21571
A21556:
    LD      B, 5
    JR      A21571
;*****
A21560:
    LD      HL, 0
    LD      (A16793), HL
    CALL    A24576
    LD      B, 5
A21571:
    PUSH    BC
    LD      A, (A16821)
    LD      B, 1

```

```
LD     HL,A16797
CALL  _OPEN_FILE
POP   BC
JP    NZ,A24297
LD    (A16822),A
LD    HL,A16797
CALL  A24980
LD    A,B
LD    B,80
AND   2
JR    Z,A21605
LD    B,112
```

A21605:

```
PUSH  BC
LD    HL,A16797
CALL  A25116
LD    D,(HL)
INC   HL
LD    E,(HL)
LD    (A16793),DE
CALL  A23504
LD    (A17042),HL
LD    A,(A16821)
CALL  _DELETE_FILE
LD    HL,(A17042)
LD    A,(A16821)
LD    BC,0
LD    D,B
LD    E,C
CALL  _MAKE_FILE
JP    NZ,A24297
LD    HL,(A17042)
LD    A,(A16821)
LD    B,2
CALL  _OPEN_FILE
POP   BC
JP    NZ,A24297
LD    (A16823),A
```

```

LD      (A16828),A
LD      HL,(A17042)
SET     3,B
CALL    A24980
LD      BC,A16939
CALL    A25182
JR      A21688
;*****
A21685:
CALL    A23796
A21688:
CALL    A23769
JR      NZ,A21685
LD      A,(A16822)
CALL    _CLOSE_FILE
JP      NZ,A24297
LD      A,(A16822)
LD      B,A
CALL    A25087
CALL    A24853
JP      A22538
;*****
A21715:
CALL    A23324
RET     NC
LD      HL,0
LD      (A16793),HL
CALL    A11777
JR      NZ,A21732
SCF
RET
;*****
A21732:
LD      B,82
CALL    A20735
RET     NC
LD      A,65
CALL    A23494

```

```
LD     HL,A16797
CALL  A25116
JP    NC,A24354
LD     HL,A16797
CALL  _CHECK_FCB
JR    Z,A21788
LD     A,(A16821)
LD     B,1
LD     HL,A16797
CALL  _OPEN_FILE
JP    NZ,A24297
LD     (A16822),A
LD     HL,A16797
LD     A,(A16822)
LD     B,1
CALL  A24980
```

A21788:

```
LD     HL,(A16793)
LD     A,H
OR     L
SCF
RET    Z
LD     C,L
LD     B,H
PUSH  BC
LD     A,(A16822)
CALL  A25151
LD     L,A
PUSH  HL
JR    A21820
```

;*****

A21808:

```
PUSH  BC
LD     H,A
LD     A,L
CP    16
PUSH  HL
LD     A,H
```

```

        JR      NZ ,A21820
        CALL   A23796
A21820:
        CALL   A23769
        POP    HL
        JP     Z ,A24336
        CP     13
        POP    BC
        JR     NZ ,A21808
        DEC    BC
        LD     A ,B
        OR     C
        JR     NZ ,A21808
        SCF
        RET
;*****
A21839:
        LD     HL , (A16833)
        JP     (HL)
;*****
A21843:
        LD     HL , (A16831)
        JP     (HL)
;*****
A21847:
        CALL   A21981
        PUSH   AF
        CALL   A17234
        POP    AF
        RET
;*****
A21856:
        LD     HL ,A12043
        LD     (A16201) ,HL
        LD     A ,4
        LD     (A17014) ,A
        POP    AF
        RET

```

;*****

A21869:

PUSH AF
CALL A17234
POP AF

A21874:

PUSH AF
CALL A28510
NOP
LD A, (A17015)
CP 13
JR Z, A21856
LD A, 4

A21888:

LD (A17015), A
LD A, (A16828)
LD (A16823), A
CALL A25151
OR A
PUSH AF
PUSH HL
CALL M, A23769
POP HL
POP AF
LD B, A
AND 34
JR Z, A21976
INC HL
INC HL
INC HL
INC HL
LD E, (HL)
INC HL
LD D, (HL)
DEC DE
LD (HL), D
DEC HL
LD (HL), E


```
LD    A,D
OR    E
JR    Z,A21973
POP   AF
PUSH  AF
CP    13
JR    NZ,A21976
POP   AF
PUSH  HL
PUSH  DE
PUSH  BC
CALL  A23796
POP   BC
POP   DE
```

A21943:

```
PUSH  DE
PUSH  BC
BIT   7,B
CALL  NZ,A23769
XOR   A
CALL  A23796
POP   BC
POP   DE
DEC   DE
LD    A,D
OR    E
JR    NZ,A21943
POP   HL
```

A21962:

```
INC   HL
INC   HL
LD    E,(HL)
INC   HL
LD    D,(HL)
DEC   HL
DEC   HL
LD    (HL),D
DEC   HL
```

```

        LD      (HL) , E
        RET
;*****
A21973:
        CALL   A21962
A21976:
        POP    AF
        CALL   A23796
        RET
;*****
A21981:
        LD      A, (A16827)
        LD      (A16822) , A
        CALL   A23769
        JP      Z, A24336
        PUSH   AF
        LD      A, (A16827)
        CALL   A25151
        AND    34
        JR      Z, A22047
        INC    HL
        INC    HL
        INC    HL
        INC    HL
        LD      E, (HL)
        INC    HL
        LD      D, (HL)
        DEC    DE
        LD      (HL) , D
        DEC    HL
        LD      (HL) , E
        LD      A, D
        OR     E
        JR      Z, A22037
        POP    AF
        PUSH   AF
        CP     13
        JR      NZ, A22047

```

```
        PUSH    HL
A22026:
        PUSH    DE
        CALL    A23769
        POP     DE
        DEC     DE
        LD      A,D
        OR      E
        JR      NZ,A22026
        POP     HL
```

```
A22037:
        INC     HL
        INC     HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        DEC     HL
        DEC     HL
        LD      (HL),D
        DEC     HL
        LD      (HL),E
```

```
A22047:
        POP     AF
        RET
```

```
;*****
```

```
A22049:
        CALL    A23324
        RET     NC
        LD      A,65
        CALL    A23494
        CALL    A11777
        JR      Z,A22069
        CALL    A22923
        JR      C,A22181
        RET
```

```
;*****
```

```
A22069:
        LD      HL,A16797
```

```
CALL A25116
JP NC,A24354
PUSH HL
LD HL,A16797
CALL _CHECK_FCB
JR Z,A22101
LD A,(A16821)
LD B,1
LD HL,A16797
CALL _OPEN_FILE
JP NZ,A24297
```

A22101:

```
POP HL
PUSH AF
LD A,(HL)
INC HL
OR (HL)
LD B,1
JR Z,A22112
LD B,3
```

A22112:

```
POP AF
LD (A16827),A
LD HL,A16797
CALL A24980
```

A22122:

```
LD HL,(A16197)
LD A,83
CP L
JR NZ,A22135
LD A,85
CP H
SCF
RET Z
```

A22135:

```
LD (A16195),HL
LD HL,A21843
LD (A16197),HL
```

```

        LD     HL, (A16203)
        LD     (A16795), HL
        LD     HL, A22157
        LD     (A16203), HL
        SCF
A22157:
        RET
;*****
A22158:
        LD     C, A
        LD     A, 1
        CP     B
        LD     A, C
        JR     Z, A22170
        CALL  A24869
        JR     A22203
;*****
A22170:
        LD     (A16822), A
        LD     (A16827), A
        CALL  _RESET_FILE
        JR     A22228
;*****
A22181:
        PUSH  HL
        PUSH  DE
        OR    A
        PUSH  AF
        CALL  A22911
        LD    A, (HL)
        INC  HL
        OR   (HL)
        JR   Z, A22296
        POP  AF
        PUSH HL
        LD   HL, A16797
        CALL _CHECK_FCB
        JR   Z, A22158

```

A22203:
LD A, (A16821)
LD HL, A16797
LD B, 1
CALL _OPEN_FILE
JP NZ, A24297
LD (A16827), A
LD HL, A16797
LD B, 3
CALL A24980

A22228:
POP HL
LD B, (HL)
DEC HL
LD C, (HL)
PUSH BC
JR A22249

;*****

A22235:
PUSH DE
PUSH BC
JR A22244

;*****

A22239:
PUSH BC
CALL A23769
POP BC

A22244:
LD A, B
OR C
DEC BC
JR NZ, A22239

A22249:
POP BC
POP DE
LD A, D
OR E
DEC DE

```

A22254:
    JR     NZ ,A22235
    POP   DE
    PUSH  BC
A22258:
    JR     A22274
;*****
A22260:
    DEC   BC
    LD    A , B
    OR    C
    JR    NZ ,A22267
    POP   BC
    PUSH  BC
A22267:
    PUSH  BC
    PUSH  DE
    CALL  A23769
    POP   DE
    POP   BC
A22274:
    LD    A , D
    OR    E
    DEC   DE
    JR    NZ ,A22260
    POP   DE
    LD    A , (A16827)
    CALL  A25151
    LD    DE , 4
    ADD   HL , DE
    LD    (HL) , C
    INC   HL
    LD    (HL) , B
    JP    A22122
;*****
A22296:
    POP   AF
    JP    NZ ,A24357

```

```
LD HL,A16797
CALL _CHECK_FCB
JR Z,A22336
```

A22308:

```
LD HL,A16797
LD A,(A16821)
LD B,1
CALL _OPEN_FILE
JP NZ,A24297
LD (A16827),A
LD B,1
LD HL,A16797
CALL A24980
POP HL
JR A22436
```

;*****

A22336:

```
LD C,A
LD A,1
CP B
JR NZ,A22448
LD A,C
LD (A16827),A
CALL A25151
INC HL
INC HL
LD A,(HL)
INC HL
LD H,(HL)
LD L,A
LD DE,13
ADD HL,DE
LD C,(HL)
INC HL
LD B,(HL)
LD DE,11
ADD HL,DE
LD E,(HL)
```



```
INC    HL
LD     D, (HL)
PUSH  BC
LD     BC, 7
ADD   HL, BC
POP   BC
LD     A, (HL)
INC   HL
LD     H, (HL)
LD     L, A
LD     A, H
AND   3
LD     H, A
EX    DE, HL
SBC   HL, BC
JR    Z, A22404
LD     A, H
OR    A
JP    NZ, A24330
LD     A, L
LD     L, H
RLA
JP    C, A24330
RLA
JP    C, A24330
LD     H, A
```

A22404:

```
ADD   HL, DE
JP    C, A24330
POP   BC
POP   BC
EX    DE, HL
LD     H, B
LD     L, C
SBC   HL, DE
JP    Z, A22122
JR    NC, A22445
PUSH  BC
```

```

        LD     A, (A16827)
        LD     (A16822), A
        CALL  _RESET_FILE
        JR     A22436
;*****
A22432:
        PUSH  BC
        CALL  A23769
A22436:
        POP   BC
        LD    A, B
        OR    C
        DEC   BC
        JR    NZ, A22432
        JP    A22122
;*****
A22445:
        PUSH  HL
        JR    A22436
;*****
A22448:
        LD    A, C
        CALL  A24869
        JP    A22308
;*****
A22455:
        CALL  A23324
        RET   NC
        LD    A, 65
        CALL  A23494
        CALL  A11777
        JR    Z, A22475
        CALL  A22923
        JR    C, A22562
        RET
;*****
A22475:
        LD    HL, A16797

```

```
CALL    A25116
JP      NC,A24354
PUSH   HL
LD      HL,A16797
CALL   _CHECK_FCB
CALL   Z,A24869
LD      A,(A16821)
LD      B,2
LD      HL,A16797
CALL   _OPEN_FILE
JP      NZ,A24297
PUSH   AF
CALL   _RESET_FILE
JP      NZ,A24297
POP    AF
POP    HL
PUSH   AF
LD      A,(HL)
INC    HL
OR     (HL)
POP    BC
LD      A,B
LD      B,48
JR     NZ,A22529
LD      B,16
```

A22529:

```
LD      (A16828),A
LD      HL,A16797
CALL   A24980
```

A22538:

;Here is some tricky code:

;The intent is to determine whether or not the
vector stored at
;(A16201) is A21839. Since the Z80 has no direct
16-bit compare
;instructions, the choices are to do a 16-bit

subtraction from
;A21839 (ZF=1 if identical, ZF=0 if not), or else do
8-bit CPs
;with the lobyte and hobyte, one at a time (if
both match the
;corresponding bytes of A21839, it's the same,
otherwise it's not).

;The code here does the latter. The only problem is
that it
;has to hard-code the lobyte and hobyte value of
A21839 in-line.
;A disassembler has no way to know that these
immediate values are
;linked to the 16-bit value of the label A21839. If
the assembler
;used by Lazer Microsystems was full-featured
enough, however, it
;would have been easy enough to establish the
linkage in the source
;code. If not, they'd have had to assemble it once
to find out the
;value of the label A21839, then hard-code the
correct values, and
;reassemble. This would be a pain because you'd
have to do this
;every time you edited the listing and made the code
move around.
;I've done it here using the LOW and HIGH operators
of Z80ASM+.

;Note that there is a bug in the original code to do
the comparison!
;Instead of the expected CP H for the hobyte, it's
CP L! The bug is
;present in the SmartBASIC 2.0 source code as well.
I've taken the
;liberty of fixing it here in this regenerated SB1.x

code.

;NOTE: the current distributed versions of SB1.x
(20X, 20Y) are
;*NOT* corrected for this bug!

A21839_LO EQU LOW A21839 ;Z80ASM+ operator to
take lobyte
A21839_HI EQU HIGH A21839 ;Z80ASM+ operator to
take hobyte

LD HL, (A16201) ;get current vector to
transmit data to device

; LD A, 79 ;lobyte of A21839 (orig. 79)
LD A, A21839_LO ;lobyte of A21839

CP L ;is it A21839?
JR NZ, A22551 ;no, so make it A16201

; LD A, 85 ;hobyte of A21839 (orig. 85)
LD A, A21839_HI ;hobyte of A21839

; CP L ;original SB1.0 bug!
CP H ;bugfix by RFD

SCF ;CF=1
RET Z ;yes, it's A21839, so exit
;without changing the vector

A22551:

LD (A16199), HL ;save as current PR vector
for LOAD and WRITE

LD HL, A21839

LD (A16201), HL ;and set new current vector
for

;transmitting data to device

to A21839

SCF ;CF=1

```
RET
;*****
A22562:
PUSH HL
PUSH DE
OR A
PUSH AF
CALL A22911
LD A, (HL)
INC HL
OR (HL)
JP Z, A22764
POP AF
PUSH HL
LD HL, A16797
CALL _CHECK_FCB
CALL Z, A24869
LD DE, A16797
LD A, (A16821)
CALL _TRIM_FILE
JP NZ, A24297
LD HL, A16797
LD B, 1
LD A, (A16821)
CALL _OPEN_FILE
JP NZ, A24297
LD (A16827), A
LD HL, A16797
LD B, 3
CALL A24980
CALL A23504
LD (A17042), HL
LD A, (A16821)
CALL _DELETE_FILE
LD HL, (A17042)
LD A, (A16821)
LD BC, 0
LD D, B
```

```

LD      E,C
CALL   _MAKE_FILE
JP     NZ,A24297
LD     HL,(A17042)
LD     A,(A16821)
LD     B,2
CALL   _OPEN_FILE
JP     NZ,A24297
POP    HL
LD     B,(HL)
DEC    HL
LD     C,(HL)
PUSH   BC
LD     (A16793),BC
LD     HL,(A17042)
LD     (A16828),A
LD     (A16823),A
LD     B,184
CALL   A24980
LD     A,(A16827)
LD     (A16822),A
JR     A22714
;*****
A22697:
      PUSH DE
      PUSH BC
      JR   A22709
;*****
A22701:
      PUSH BC
      CALL A23769
      CALL A23796
      POP  BC
A22709:
      LD   A,B
      OR   C
      DEC  BC
      JR   NZ,A22701

```

A22714:

```
POP    BC
POP    DE
LD     A,D
OR     E
DEC    DE
JR     NZ,A22697
POP    DE
PUSH   BC
JR     A22742
```

;*****

A22725:

```
DEC    BC
LD     A,B
OR     C
JR     NZ,A22732
POP    BC
PUSH   BC
```

A22732:

```
PUSH   BC
PUSH   DE
CALL   A23769
CALL   A23796
POP    DE
POP    BC
```

A22742:

```
LD     A,D
OR     E
DEC    DE
JR     NZ,A22725
POP    DE
LD     A,(A16828)
CALL   A25151
LD     DE,4
ADD    HL,DE
LD     (HL),C
INC    HL
LD     (HL),B
```



```
        JP      A22538
;*****
A22764:
        POP     AF
        JP      NZ,A24357
        LD      HL,A16797
        CALL    _CHECK_FCB
        CALL    Z,A24869
        LD      HL,A16797
        LD      A,(A16821)
        LD      B,1
        CALL    _OPEN_FILE
        JP      NZ,A24297
        LD      (A16827),A
        LD      B,1
        LD      HL,A16797
        CALL    A24980
        CALL    A23504
        LD      (A17042),HL
        LD      A,(A16821)
        CALL    _DELETE_FILE
        LD      HL,(A17042)
        LD      BC,0
        LD      D,B
        LD      E,C
        LD      A,(A16821)
        CALL    _MAKE_FILE
        JP      NZ,A24297
        LD      A,(A16821)
        LD      B,2
        LD      HL,(A17042)
        CALL    _OPEN_FILE
        JP      NZ,A24297
        LD      (A16828),A
        LD      B,152
        LD      HL,0
        LD      (A16793),HL
        LD      HL,(A17042)
```

```
CALL A24980
LD A, (A16827)
CALL A25151
INC HL
INC HL
LD C, (HL)
INC HL
LD B, (HL)
LD A, (A16828)
CALL A25182
POP BC
POP BC
LD A, (A16827)
LD (A16822), A
LD A, (A16828)
LD (A16823), A
JR A22903
```

```
;*****
```

```
A22895:
```

```
PUSH BC
CALL A23769
CALL A23796
POP BC
```

```
A22903:
```

```
LD A, B
OR C
DEC BC
JR NZ, A22895
```

```
A22908:
```

```
JP A22538
```

```
;*****
```

```
A22911:
```

```
LD HL, A16797
CALL A25116
RET C
JP A24354
```

```
;*****
```

```
A22921:
```

```

        OR      A
        RET
;*****
A22923:
        LD      A, (DE)
        CP      44
        JP      NZ, A22921
        INC     DE
        CALL    A11777
        RET     Z
        LD      B, 66
        AND     95
        CP      B
        JR      Z, A22947
        LD      B, 82
        CP      B
        JP      NZ, A22921
A22947:
        LD      C, 1
        PUSH    BC
        CALL    A23034
        POP     HL
        RET     NC
        LD      C, L
        LD      HL, (A16793)
        LD      (A16791), HL
        CALL    A11777
        JR      Z, A22978
        PUSH    BC
        CALL    A23021
        POP     BC
        RET     NC
        CALL    A11777
        RET     NZ
        INC     C
A22978:
        DEC     C
        LD      A, 66

```

```

        JR      Z,A22998
        CP      B
        LD      DE,(A16791)
        LD      HL,(A16793)
        JR      Z,A22994
A22993:
        EX      DE,HL
A22994:
        LD      A,1
        SCF
        RET
;*****
A22998:
        CP      B
        JR      Z,A23010
        LD      HL,(A16791)
        LD      DE,0
        XOR     A
        SCF
        RET
;*****
A23010:
        LD      DE,(A16791)
        LD      HL,0
        LD      A,1
        SCF
        RET
;*****
A23021:
        CALL   A23304
        RET    NC
        CALL   A11777
        RET    Z
        AND    95
        CP     B
        JR     NZ,A22921
A23034:
        XOR    16

```

```

        LD     B,A
        INC   DE
        CALL  A20754
        RET
;*****
A23042:
        LD     HL,A23140
        JR     A23050
;*****
A23047:
        LD     HL,A23148
A23050:
        PUSH  HL
        LD     BC,5
A23054:
        LD     A,(DE)
        AND   95
        LD     HL,A23156
        PUSH  BC
        LD     BC,4
        CPIR
        JR     NZ,A23136
        LD     H,B
        LD     L,C
        ADD   HL,BC
        POP   BC
        DEC   C
        JP    Z,A23137
        RRA
        AND   3
        CP    3
        JR     NZ,A23084
        INC   A
A23084:
        ADD   A,A
        INC   A
        INC   A
        INC   A

```

```
ADD    A,B
LD     B,A
EX     DE,HL
EX     (SP),HL
EX     DE,HL
ADD    HL,DE
EX     DE,HL
EX     (SP),HL
EX     DE,HL
LD     A,(HL)
INC    HL
LD     H,(HL)
LD     L,A
CALL   A5938
INC    DE
CALL   A11777
JR     Z,A23122
CP     44
JP     NZ,A23137
INC    DE
CALL   A11777
JP     A23054
```

```
;*****
```

```
A23122:
```

```
LD     A,B
LD     HL,A23160
LD     BC,15
CPIR
POP    HL
SCF
RET    Z
OR     A
RET
```

```
;*****
```

```
A23136:
```

```
POP    BC
```

```
A23137:
```

```
POP    HL
```

```

        OR      A
        RET
;*****
;MON/NOMON vector table.
;
A23140:
        DW      A23224      ;NOMON O
        DW      A23203      ;NOMON I
        DW      A23217      ;NOMON C
        DW      A23210      ;NOMON L
A23148:
        DW      A23196      ;MON O
        DW      A23175      ;MON I
        DW      A23189      ;MON C
        DW      A23182      ;MON L
;*****
;Arguments for MON/NOMON.
;
A23156:
        DB      "OICL"
;*****
;Table of characters not to print with MON.  Not
sure why not for some...
;
A23160:
        DB      19      ;^S (pause)
        DB      26      ;^Z
        DB      15      ;^O (old DELETE)
        DB      8       ;^H (BS)
        DB      3       ;^C (break)
        DB      10      ;^J (LF)
        DB      7       ;^G (bell)
        DB      11      ;^K
        DB      16      ;^P (screen dump)
        DB      5       ;^E
        DB      14      ;^N (old INSERT)
        DB      12      ;^L (FF)
        DB      23      ;^X (cancel line)

```

```

        DB      18      ; ^R
        DB      21      ; ^U
;*****
A23175:
        LD      HL,A19448
        LD      (A16829),HL
        RET
;*****
A23182:
        LD      HL,A21869
        LD      (A16833),HL
        RET
;*****
A23189:
        LD      HL,A21847
        LD      (A16831),HL
        RET
;*****
A23196:
        LD      HL,A17234
        LD      (A16835),HL
        RET
;*****
A23203:
        LD      HL,A23209
        LD      (A16829),HL
A23209:
        RET
;*****
A23210:
        LD      HL,A21874
        LD      (A16833),HL
        RET
;*****
A23217:
        LD      HL,A21981
        LD      (A16831),HL
        RET

```



```

;*****
A23224:
    LD    HL,A23209
    LD    (A16835),HL
    RET
;*****
;Table of acceptable non-letter, non-numeral
filename characters.
A23231:
    DB    "!#$%&'+-.@\"
    DB    92                                ; \ backslash
    DB    "^`{|}~|_\"
;*****
;Default filenames.
;
A23249:
    DB    "$$$$1A\"
    DB    3
A23256:
    DB    "$$$$2A\"
    DB    3
;*****
;Slot, volume and drive.
;
A23263:
    DB    "SVD\"
;*****
;Verification table for valid SVD combinations.
;
A23266:
    DB    001H,010H,013H,011H,023H,014H,024H
;*****
;EOS device numbers (in SmartBASIC Drive order)
;note that the end of the table overlaps the first
byte
;of code at A23279. It is probable that D7 was
never
;supposed to be allowed, and is the result of a bad

```

range
;check for the Dn parser. Either that, or someone
at
;MMSG (the source of the SmartBASIC 1.0 binary which
was
;used as the basis for SmartBASIC 1.x) had patched
the
;range check to allow for D7. I don't know.

;
A23273:
DB 8 ;tape 1

A23272 EQU A23273-1

DB 24 ;tape 2
DB 9 ;tape 3 (never produced by
Coleco)
DB 25 ;tape 4 (never produced by
Coleco)
DB 4 ;disk 1
DB 5 ;disk 2

;*****

A23279:
LD A, (DE) ;DB 26 ;RAM disk
CALL A11865
RET C
CALL A11885
RET C
PUSH BC
PUSH HL
LD BC, 18
LD HL, A23231
CPIR
POP HL
POP BC
SCF
RET Z
CCF

```

        RET
;*****
A23304:
        CALL    A11777
        RET     Z
        SUB     44
        INC     DE
        SCF
        RET     Z
A23313:
        CCF
        RET
;*****
A23315:
        CALL    A23304
        RET     NC
        LD      HL,A16809
        JR      A23327
;*****
A23324:
        LD      HL,A16797
A23327:
        LD      BC,10
        CALL    A11777
        RET     Z
        CALL    A23279
        RET     NC
A23338:
        EX      DE,HL
        LDI
        EX      DE,HL
        JP      PO,A23352
        CALL    A23279
        JR      C,A23338
        DB      254                ;CP 19 ;bypasses the INC DE
                                        ;I hate overlapping code!
A23351:
        INC     DE

```

A23352:

CALL A23279
JR C,A23351
LD (HL),65
INC HL
LD (HL),3
SCF
RET

;*****

A23364:

LD BC,0
CALL A11777
JR Z,A23313
CP 44
JR NZ,A23381
INC DE
CALL A11777
RET Z

A23381:

CALL A23433
RET NC
LD A,C
OR C
RET Z
CP 4
RET NC
LD A,B
LD BC,7
LD HL,A23266
CPIR
RET Z

A23401:

OR A
RET

;*****

A23403:

INC C
SUB 67

```
ADD    A,B
LD     B,A
JR     C,A23313
CALL  A11777
RET    Z
CALL  A11885
RET    NC
INC    DE
```

A23419:

```
CALL  A11777
JR     Z,A23313
CP     44
JR     NZ,A23401
INC    DE
CALL  A11777
RET    Z
```

A23433:

```
AND    223
PUSH   BC
LD     BC,3
LD     HL,A23263
CPIR
POP    BC
RET    NZ
INC    DE
JP     PE,A23403
INC    C
INC    B
RET    Z
CALL  A11777
RET    Z
SUB    49
CP     7
RET    NC
LD     HL,A23273
ADD    A,L
LD     L,A
JR     NC,A23470
```

```

        INC     H
A23470:
        LD     A, (HL)
        LD     (A16821), A
        INC     DE
        JR     A23419
;*****
A23477:
        PUSH   BC
        LD     A, 3
        LD     BC, 12
        CPIR
        POP    BC
        DEC    HL
        DEC    HL
        RET
;*****
A23489:
        LD     HL, A16809
        JR     A23497
;*****
A23494:
        LD     HL, A16797
A23497:
        PUSH   AF
        CALL  A23477
        POP    AF
        LD     (HL), A
        RET
;*****
A23504:
        LD     HL, A23249
        CALL  _CHECK_FCB
        RET    NZ
        LD     HL, A23256
        CALL  _CHECK_FCB
        RET    NZ
        JP    A24354

```

;*****

A23521:

LD A,65 ;"A"
CALL A23494 ;add the "A" to the filename

string

LD A,(A16821) ;get EOS device #
LD B,1 ;code for open for read
LD HL,A16797 ;filename string address
CALL _OPEN_FILE ;open file
LD (A16822),A ;save EOS file #
RET Z ;open okay, so exit
LD A,72 ;failed, so let's try an "H"

file

CALL A23494 ;add the "H" to the filename

string

LD A,(A16821) ;get EOS device #
LD B,1 ;code for open for read
LD HL,A16797 ;filename string address
CALL _OPEN_FILE ;open file
JP NZ,A24297 ;failed again, so error exit
LD (A16822),A ;save EOS file #
LD BC,3 ;3 bytes to read
LD HL,A16824 ;buffer address
CALL _READ_FILE ;read the data
JR NZ,A23624 ;failed, so error exit
LD A,(A16826) ;okay, so get some number
DEC A ;one less
;(could be check for zero

vs. non-zero as a flag;

;default value of (A16826)

is 255)

JR NZ,A23608 ;not done yet (or not zero),

so check something

LD BC,(A16824) ;all done, so get count

JR A23603 ;and bypass the H-file

header (throw it away)

;*****

;Read and discard H-file header.

;On entry, file is open for read, BC has number of bytes in header.
;On exit, BC bytes are read (and discarded) from the file.

A23586:

```
PUSH BC ;save count
LD A, (A16822) ;get EOS file #
LD BC,1 ;one byte to read
LD HL,A16824 ;buffer address
CALL _READ_FILE ;read the byte
JR NZ,A23624 ;failed, so error exit
POP BC ;ok, so restore count
DEC BC ;one less byte to read
```

A23603:

```
LD A,B
OR C ;is BC=0?
JR NZ,A23586 ;not yet
RET
```

;*****

A23608:

```
LD A, (A16822) ;get EOS file #
CALL _CV_A ;it's a RET (unimplemented
function)
;in all known EOS versions!
;What does "_CV_A" mean?
```

The EOS-6 listing doesn't say.

;I wonder if it means

"ConVert to ASCII"...

```
RET Z ;passes the test (whatever
it was supposed to be)
LD A, (A16822) ;failed! so get EOS file #
CALL _CLOSE_FILE ;close the file
JP A24357 ;and exit
```

;*****

A23624:

```
PUSH AF ;save error code and flags
LD A, (A16822) ;get EOS file #
```



```

        CALL  _CLOSE_FILE ;close file
        POP   AF          ;restore error code and
flags
        JP    A24297      ;and exit
;*****
A23635:
        PUSH  AF
        LD    BC,12
        LD    DE,A16809
        LD    HL,A16797
        LDIR
        LD    A,65
        CALL  A23494
        LD    A,(A16821)
        LD    DE,A16797
        LD    HL,A16885
        CALL  _QUERY_FILE
        JR    Z,A23685
        LD    A,72
        CALL  A23494
        LD    A,(A16821)
        LD    DE,A16797
        LD    HL,A16885
        CALL  _QUERY_FILE
        JR    NZ,A23748
A23685:
        LD    A,97
        CALL  A23489
        LD    A,(A16821)
        LD    HL,A16809
        CALL  _DELETE_FILE
        JR    Z,A23715
        LD    A,104
        CALL  A23489
        LD    A,(A16821)
        LD    HL,A16809
        CALL  _DELETE_FILE
A23715:

```

```
LD     HL,A16797
CALL  A23477
LD     A,(HL)
PUSH  AF
LD     HL,A16809
CALL  A23477
POP   AF
OR    32
LD    (HL),A
LD    A,(A16821)
LD    DE,A16797
LD    HL,A16809
CALL  _RENAME_FILE
JP    NZ,A24297
```

A23748:

```
POP   AF
CALL  A23489
LD    A,(A16821)
LD    DE,(A17044)
LD    HL,A16809
CALL  _RENAME_FILE
RET   Z
JP    A24297
```

;*****

A23769:

```
LD    A,(A16822)
LD    BC,1
LD    HL,A16824
CALL  _READ_FILE
JR    NZ,A23790
LD    A,B
OR    C
LD    HL,A16824
LD    A,(HL)
RET
```

;*****

A23790:

```
CP    10
```

```

        RET     Z
        JP     A24297
;*****
A23796:
        LD     HL,A16824
        LD     (HL),A
        LD     BC,1
        LD     A,(A16823)
        CALL  _WRITE_FILE
        RET     Z
        JP     A24297
;*****
A23813:
        LD     HL,(A16201)
        PUSH  HL
        LD     HL,A23796
        LD     (A16201),HL
        CALL  A23324
        JP     NC,A23930
        CALL  A23364
        JP     NC,A23930
        CALL  A23504
        LD     (A17044),HL
        LD     A,(A16821)
        CALL  _DELETE_FILE
        CALL  A25207
        LD     A,(A16821)
        LD     HL,(A17044)
        LD     BC,(A16793)
        LD     DE,(A16791)
        CALL  _MAKE_FILE
        JP     NZ,A24297
        LD     A,(A16821)
        LD     B,2
        CALL  _OPEN_FILE
        JP     NZ,A24297
        LD     (A16823),A
        LD     B,255

```

```
LD HL,(A17044)
CALL A24980
CALL A24099
LD A,(A16823)
CALL _CLOSE_FILE
JP NZ,A24297
LD A,(A16823)
LD B,A
CALL A25087
LD A,(A16821)
LD DE,(A17044)
CALL _TRIM_FILE
JP NZ,A24297
LD A,65
CALL A23635
SCF
```

A23930:

```
POP HL
LD (A16201),HL
RET
```

;*****

A23935:

```
CALL A23769
RET NZ
LD HL,(A16791)
LD (A16197),HL
LD HL,(A16793)
LD (A16201),HL
LD HL,(A16795)
LD (A16203),HL
LD A,(A16822)
CALL _CLOSE_FILE
JP NZ,A24297
LD A,(A16822)
LD B,A
CALL A25087
LD A,13
```

A23975:

```
RET
;*****
;Main part of LOAD.
```

A23976:

```
CALL A24025
LD HL,A23935
```

A23982:

```
LD (A16197),HL
LD HL,(A16203)
LD (A16795),HL
LD HL,(A16835)
LD (A16203),HL
LD HL,(A16201)
LD (A16793),HL
LD HL,A23975
LD (A16201),HL
JP A6356
```

```
A24010 EQU $-2 ;vector address
```

```
;*****
;Main part of RUN.
```

A24012:

```
CALL A24025
```

A24015:

```
LD HL,A24054
JR A23982
```

```
;*****
```

A24020:

```
CALL A24033
JR A24015
```

```
;*****
```

A24025:

```
CALL A23324
RET NC
CALL A23364
```

```
RET NC
A24033:
CALL A23521
LD A, (A16822)
LD B, 255
LD HL, A16797
CALL A24980
LD HL, (A16197)
LD (A16791), HL
RET
```

```
;*****
```

```
A24054:
CALL A23769
RET NZ
LD HL, (A16791)
LD (A16197), HL
LD HL, (A16793)
LD (A16201), HL
LD HL, (A16795)
LD (A16203), HL
LD A, (A16822)
CALL _CLOSE_FILE
JP NZ, A24297
LD A, (A16822)
LD B, A
CALL A25087
EXX
LD C, 0
EXX
JP A29995
```

```
;*****
```

```
A24099:
XOR A
LD (A16148), A
LD HL, (A16089)
LD BC, (A16091)
JR A24116
```

```
;*****
```

```
A24112:
    CALL  A13459
    DEC   BC
```

```
A24116:
    LD    A,B
    OR    C
    JR    NZ,A24112
    LD    A,32
    LD    (A16148),A
    RET
```

```
;*****
```

```
;File I/O error messages:  length, message.  In
ERRNUM order (1st is 22).
```

```
;Errors 22, 26, 34, and 35 are patched out (0
length); possibly these are
```

```
;valid errors under AppleSoft which are irrelevant
for the ADAM hardware.
```

```
;I don't have an AppleSoft reference manual to look
it up.
```

```
;
```

```
A24126:
    DB    0                                ;22
```

```
unknown error
```

```
    DB    11,"Range Error"                ;23
```

```
    DB    15,"Write Protected"            ;24
```

```
    DB    11,"End of Data"                ;25
```

```
    DB    0                                ;26
```

```
unknown error
```

```
    DB    14,"File Not Found"              ;27
```

```
    DB    9,"I/O Error"                    ;28
```

```
    DB    12,"No More Room"                ;29
```

```
    DB    11,"File Locked"                 ;30
```

```
    DB    12,"Syntax Error"                ;31
```

```
    DB    20,"No Buffers Available"         ;32
```

```
    DB    18,"File Type Mismatch"         ;33
```

```
    DB    0                                ;34
```

```
unknown error
```

```
    DB    0                                ;35
```

unknown error

DB 23, "Control Buffer Overflow" ;36

;*****

;EOS to SmartBASIC error converter.

;

A24297:

CP FULL_DIR_ERR

JR Z, A24345 ;No More Room

CP FULL_TAPE_ERR

JR Z, A24345 ;No More Room

CP NO_FILE_ERR

JR Z, A24339 ;File Not Found

CP RANGE_ERR

JR Z, A24330 ;Range Error

CP NO_FCB_ERR

JR Z, A24354 ;No Buffers

Available

CP DELETE_ERR

JR Z, A24348 ;File Locked

;*****

;Multiple file error handler. Entry at 24359 prints error code in A.

;Another wretched cascade of garbage instructions (LD BC,nnnn). I'm

;sure I've mentioned this, but I hate overlapping code!

;

A24321:

LD A, 28 ;I/O Error

DB 1

LD A, 22 ;unused; orphaned

DB 1

A24327:

LD A, 36 ;Control Buffer Overflow

DB 1

A24330:

LD A, 23 ;Range Error

DB 1


```

A24333:
    LD    A,24        ;Write Protected
    DB    1
A24336:
    LD    A,25        ;End of Data
    DB    1
A24339:
    LD    A,27        ;File Not Found
    DB    1
    LD    A,26        ;unused; orphaned
    DB    1
A24345:
    LD    A,29        ;No More Room
    DB    1
A24348:
    LD    A,30        ;File Locked
    DB    1
A24351:
    LD    A,31        ;Syntax Error
    DB    1
A24354:
    LD    A,32        ;No Buffers Available
    DB    1
A24357:
    LD    A,33        ;File Type Mismatch
A24359:
    LD    (A16128),A
;*****
    EXX
    SET   2,B
    BIT   6,B
    EXX
    JR    Z,A24384
    EXX
    LD    (A16122),DE
    LD    (A16124),HL
    BIT   0,B
    EXX

```

```

        JP      NZ ,A8082
A24384:
        CALL   A24419
        CALL   A12128
        LD     A , (A16128)
        SUB    22
        LD     B ,A
A24396:
        LD     HL ,A24126
        JR     Z ,A24410
A24401:
        LD     A , (HL)
        INC    HL
        ADD    A ,L
        LD     L ,A
        JR     NC ,A24408
        INC    H
A24408:
        DJNZ   A24401
A24410:
        CALL   A12110
        CALL   A12128
        JP     A16035
;*****
A24419:
        LD     HL ,A12043
        LD     (A16201) ,HL
        LD     HL ,A17289
        LD     (A16197) ,HL
        LD     HL ,A17234
        LD     (A16203) ,HL
        LD     B ,2
        LD     HL ,A16836
A24442:
        PUSH   BC
        INC    HL
        LD     A , (HL)
        INC    HL

```

```
OR      A
JR      Z,A24476
XOR     A
DEC     HL
LD      (HL),A
INC     HL
LD      A,(HL)
PUSH    HL
CALL    _CLOSE_FILE
POP     HL
PUSH    HL
INC     HL
LD      A,(HL)
INC     HL
LD      H,(HL)
LD      L,A
LD      E,L
LD      D,H
LD      BC,23
ADD     HL,BC
LD      A,(HL)
CALL    _TRIM_FILE
POP     HL
```

A24476:

```
LD      DE,8
ADD     HL,DE
POP     BC
DJNZ   A24442
LD      HL,A16857
LD      B,2
LD      DE,14
XOR     A
```

A24492:

```
LD      (HL),A
ADD     HL,DE
DJNZ   A24492
RET
```

;*****

A24497:
CALL A23324
RET NC
PUSH DE
CALL A20730
POP HL
JR C,A24515
EX DE,HL
LD HL,0
LD (A16793),HL

A24515:
CALL A23364
RET NC
LD A,65
CALL A23494
LD DE,A16797
LD HL,A16885
LD A,(A16821)
CALL _QUERY_FILE
JR Z,A24562
CP 5
JP NZ,A24297
LD HL,A16797
LD BC,0
LD D,B
LD E,C
LD A,(A16821)
CALL _MAKE_FILE
JR Z,A24576
JP A24297

;*****

A24562:
LD HL,A16797
CALL _CHECK_FCB
JR NZ,A24576
CALL _RESET_FILE
JP NZ,A24297

A24576:

```

        LD     HL,A16857
        LD     DE,14
        LD     B,2
A24584:
        LD     A,(HL)
        OR     A
        JR     Z,A24594
        ADD    HL,DE
        DJNZ   A24584
        JP     A24354
;*****
A24594:
        EX     DE,HL
        LD     HL,A16797
        LD     BC,12
        LDIR
        LD     HL,(A16793)
        EX     DE,HL
        LD     (HL),E
        INC    HL
        LD     (HL),D
        SCF
        RET
;*****
A24612:
        CALL   A23324
        RET    NC
        CALL   A23364
        RET    NC
        LD     A,65
        CALL   A23494
        LD     HL,A16797
        CALL   _CHECK_FCB
        JR     Z,A24702
        LD     HL,A16797
        CALL   A25195
        DEC    C
        DEC    C

```

```

        PUSH    BC
        LD      HL,A16837
        LD      B,2
A24647:
        LD      DE,7
        LD      A,(HL)
        INC     HL
        LD      C,(HL)
        ADD     HL,DE
        OR      A
        JR      Z,A24683
        RRCA
        JR      C,A24683
        AND     96
        JR      Z,A24683
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        DEC     HL
        EX      (SP),HL
        PUSH   BC
        LD      B,L
        LD      C,L
        LD      HL,A16797
        CALL   A11852
        LD      L,C
        POP    BC
        EX      (SP),HL
        JR      Z,A24700
A24683:
        INC     HL
        INC     HL
        DJNZ   A24647
        POP    HL
        LD      A,(A16821)
        LD      DE,A16797
        CALL   _TRIM_FILE
        JP     A24853

```

;*****

A24700:

POP HL
LD A,C

A24702:

PUSH AF
CALL A25151
RRCA
RLA
JR NC,A24726
POP AF
PUSH AF
CALL A24911
JP Z,A24809
JP P,A24809
POP AF
XOR 3
JR A24702

;*****

A24726:

AND 192
JP Z,A24823
RLCA
JR NC,A24763
JR A24739

;*****

A24736:

CALL A23796

A24739:

CALL A23769
JR NZ,A24736
LD A,(A16822)
CALL _CLOSE_FILE
LD A,(A16822)
LD B,A
CALL A25087
LD HL,A17289
LD (A16197),HL

A24763:

```
POP    AF
PUSH   AF
CALL   _CLOSE_FILE
JP     NZ,A24297
LD     A,(A16821)
LD     DE,(A17042)
CALL   _TRIM_FILE
LD     HL,(A17042)
LD     (A17044),HL
LD     A,65
CALL   A23494
LD     A,65
CALL   A23635
POP    BC
CALL   A25087
LD     HL,A12043
LD     (A16201),HL
JR     A24853
```

;*****

A24809:

```
LD     HL,A17289
LD     (A16197),HL
LD     HL,A17234
LD     (A16203),HL
JR     A24829
```

;*****

A24823:

```
LD     HL,A12043
LD     (A16201),HL
```

A24829:

```
POP    AF
PUSH   AF
CALL   _CLOSE_FILE
JP     NZ,A24297
LD     A,(A16821)
LD     DE,A16797
CALL   _TRIM_FILE
```



```
POP    BC
JP     NZ,A24297
CALL   A25087
```

A24853:

```
LD     HL,A16797
CALL   A25116
CCF
RET    C
LD     BC,-12
ADD    HL,BC
LD     (HL),0
SCF
RET
```

;*****

A24869:

```
PUSH   AF
CALL   A25116
POP    AF
EX     DE,HL
LD     B,(HL)
PUSH   BC
PUSH   HL
PUSH   AF
CALL   A24911
JR     Z,A24897
JP     P,A24897
POP    AF
XOR    3
CALL   A24907
```

A24893:

```
POP    HL
POP    AF
LD     (HL),A
RET
```

;*****

A24897:

```
POP    AF
CALL   A24903
```

```

        JR      A24893
;*****
A24903:
        PUSH   AF
        JP     A24809
;*****
A24907:
        PUSH   AF
        JP     A24739
;*****
A24911:
        XOR    3
        LD     HL,A16838
        LD     B,2
        LD     DE,10
A24921:
        CP     (HL)
        JR     Z,A24930
        ADD   HL,DE
        DJNZ  A24921
        XOR   A
        JR     A24932
;*****
A24930:
        DEC   HL
        LD   A,(HL)
A24932:
        OR   A
        RET
;*****
A24934:
        CALL A24954
        RET  NC
        CALL A12071
        SCF
        RET
;*****
A24943:

```

```

        CALL    A24954
        RET     NC
        CALL    A12097
        SCF
        RET
;*****
A24952:
        OR     A
        RET
;*****
A24954:
        CALL    A11777
        RET     Z
        CP     35
        JR     NZ,A24952
        INC    DE
        CALL    A11777
        RET     Z
        CALL    A11885
        RET     NC
        SUB    48
        CP     8
        RET     NC
        LD     L,A
        LD     H,0
        RET
;*****
A24980:
        PUSH   DE
        PUSH   HL
        PUSH   BC
        PUSH   AF
        LD     B,2
        LD     HL,A16837
        LD     DE,10
A24992:
        LD     A,(HL)
        OR     A

```

```
JR      Z,A25002
ADD     HL,DE
DJNZ   A24992
JP      A24354
```

```
;*****
```

```
A25002:
```

```
POP     AF
POP     BC
LD      (HL),B
INC     HL
LD      (HL),A
EX      (SP),HL
PUSH   HL
PUSH   BC
PUSH   AF
CALL   _CHECK_FCB
JR     NZ,A25065
POP     AF
POP     BC
EX     DE,HL
POP     HL
EX     (SP),HL
INC     HL
LD     (HL),E
INC     HL
LD     (HL),D
PUSH   BC
INC     B
JR     Z,A25061
POP     BC
POP     DE
PUSH   DE
PUSH   BC
PUSH   AF
PUSH   HL
EX     DE,HL
BIT    3,B
JR     NZ,A25071
```

```

        CALL    A25116
        JP      NC,A24354
        EX      DE,HL
        POP     HL
A25048:
        INC     HL
        EX      DE,HL
        LDI
        LDI
        DEC     HL
        DEC     HL
        LDI
        LDI
        POP     AF
A25061:
        POP     BC
        POP     HL
        POP     DE
        RET
;*****
A25065:
        POP     AF
        POP     BC
        POP     HL
        POP     HL
        POP     DE
        RET
;*****
A25071:
        POP     HL
        LD      DE,3
        LD      A,B
        XOR     8
        SBC    HL,DE
        LD      (HL),A
        ADD    HL,DE
        LD      DE,A16793
        JR      A25048

```

;*****

A25087:

PUSH DE
PUSH AF
PUSH HL
PUSH BC
LD A,B
LD B,2
LD HL,A16837
LD DE,9

A25100:

INC HL
CP (HL)
JR NZ,A25108
DEC HL
LD (HL),0
INC HL

A25108:

ADD HL,DE
DJNZ A25100
POP BC
POP HL
POP AF
POP DE
RET

;*****

A25116:

CALL A25195
LD B,C
LD A,2
LD DE,A16857

A25125:

PUSH AF
CALL A11852
JR Z,A25144
PUSH HL
LD HL,14
ADD HL,DE

```

EX      DE ,HL
POP     HL
POP     AF
DEC     A
JR      NZ ,A25125
OR      A
RET
;*****
A25144:
POP     AF
LD      HL ,12
ADD     HL ,DE
SCF
RET
;*****
A25151:
PUSH   DE
PUSH   BC
LD      C ,A
LD      B ,2
LD      HL ,A16837
LD      DE ,10
A25162:
LD      A , (HL)
OR      A
JR      Z ,A25172
INC     HL
LD      A , (HL)
DEC     HL
CP      C
JR      Z ,A25178
A25172:
ADD     HL ,DE
DJNZ   A25162
JP      A24330
;*****
A25178:
POP     BC

```

```
POP DE
LD A, (HL)
RET
```

```
;*****
```

```
A25182:
```

```
PUSH BC
CALL A25151
LD BC, 8
ADD HL, BC
POP BC
LD (HL), C
INC HL
LD (HL), B
RET
```

```
;*****
```

```
A25195:
```

```
PUSH HL
LD C, 0
LD A, 3
```

```
A25200:
```

```
INC C
CP (HL)
INC HL
JR NZ, A25200
POP HL
RET
```

```
;*****
```

```
A25207:
```

```
PUSH DE
LD HL, (A16201)
PUSH HL
LD HL, A25236
LD (A16201), HL
LD HL, 0
LD (A16791), HL
LD (A16793), HL
CALL A24099
POP HL
```



```

        LD      (A16201),HL
        POP    DE
        RET
;*****
A25236:
        LD      HL,(A16791)
        LD      BC,1
        OR      A
        ADC    HL,BC
        LD      (A16791),HL
        RET    NC
        LD      HL,(A16793)
        INC    HL
        LD      (A16793),HL
        RET
;*****
;INIT protect check filename.
;
A25257:
        DB      "BASICPGM"
        DB      2,3
;*****
A25267:
        CALL   A23324
        RET    NC
        CALL   A23364
        RET    NC
        LD      HL,A16885
        LD      DE,A25257
        LD      A,(A16821)
        CALL   _QUERY_FILE
        SCF
        RET    Z
        LD      HL,A16797
        CALL   A25195
        LD      B,0
        DEC    BC
        DEC    BC

```

```
ADD HL,BC
LD (HL),A
LD HL,A16797
LD A,(A16821)
```

A25307:

```
CALL A31276
NOP
NOP
CALL _INIT_TAPE_DIR
LD A,(A16821)
LD HL,A25365
LD B,2
CALL _OPEN_FILE
JP NZ,A24297
LD HL,A25362
LD BC,3
PUSH AF
CALL _WRITE_FILE
JP NZ,A24297
POP AF
CALL _CLOSE_FILE
JP NZ,A24297
SCF
RET Z
JP A24297
```

;*****

A25354:

```
LD BC,4880 ;B=13H, C=01H: code for
January 13th
LD D,87 ;D=57H: code for 1957
;Lazer Microsystems
programmer Joel Lagerquist
;says that this date is
probably the birthday
;of fellow programmer Robert
Greenberg (son of
;Coleco president Alan
Greenberg).
```

```

; (JKL to RFD, spring 1997)
        JP      _SET_DATE
;*****
;INIT data.
;
A25362:
        JP      _GOTO_WP      ;block 0 boot routine--jump
to SmartWriter
A25365:
        DB      "BOOT"      ;directory entry for BOOT
        DB      3
;*****
A25370:
        LD      A,3
        LD      (A17008),A
        LD      (A17005),A
        LD      A,128
        LD      (A17004),A
        PUSH   IY
        PUSH   IX
        LD      B,1
        LD      C,128
        CALL   WRITE_REGISTER
        CALL   A25433
        CALL   A25433
        LD      B,1
        LD      C,192
        CALL   WRITE_REGISTER
        POP    IX
        POP    IY
        RET
;*****
;HGR2 VRAM table address data.
;
A25412:
        DB      0      ;sprite attribute table
        DW      8064
        DB      1      ;sprite generator table

```

```

        DW      14336
        DB      2          ;pattern name table
        DW      6144
        DB      3          ;pattern generator table
        DW      8192
        DB      4          ;color table
        DW      0
        DB      255       ;end of table
;*****
;HGR2 VDP register data.
;
A25428:
        DB      0,2       ;register 0, data 2
        DB      7,1       ;register 7, data 1

A25431      EQU      A25428+3

        DB      255       ;end of table
;*****
A25433:
        LD      HL,A16763
        LD      (HL),0
        INC     HL
        LD      (HL),0
        LD      HL,A25412
        CALL    A19315
        LD      HL,A25428
        CALL    A19334
        LD      HL,8192
        LD      DE,6144
        LD      A,0
        CALL    FILL_VRAM
        LD      HL,0
        LD      DE,6144
A25470:
        LD      A,17

A25471      EQU      A25470+1      ;color

```

```
CALL    FILL_VRAM
NOP
NOP
NOP
NOP
NOP
CALL    A26272
RET
```

```
;*****
```

```
A25484:
```

```
LD      A,2
LD      (A17008),A
XOR     A
LD      (A17005),A
LD      A,128
LD      (A17004),A
LD      HL,A16953
OR      (HL)
LD      (HL),A
LD      A,128
LD      HL,A16954
OR      (HL)
LD      (HL),A
PUSH    IX
PUSH    IY
LD      B,1
LD      C,128
CALL    WRITE_REGISTER
CALL    A25433
CALL    A25433
LD      HL,6784
LD      DE,128
LD      A,160
CALL    FILL_VRAM
LD      HL,13568      ;VRAM address
LD      DE,768
LD      A,255
```

```
CALL FILL_VRAM
LD HL,32
LD BC,96
LD DE,13568 ;VRAM address
CALL PUT_ASCII
LD HL,5376
LD DE,768
```

A25567:

```
LD A,241
```

A25568 EQU A25567+1

```
CALL FILL_VRAM
LD BC,7683
LD DE,276
LD HL,6144
LD A,48
EX AF,AF'
LD A,0
CALL A17334
LD B,1
LD C,192
CALL WRITE_REGISTER
POP IY
POP IX
RET
```

;*****

A25601:

```
LD A,(A17008)
CALL A10178
LD A,C
LD C,B
LD B,A
```

A25610:

```
PUSH IY
PUSH IX
PUSH BC
PUSH DE
```

```
CALL    A26072
PUSH    DE
LD      HL,8192
ADD     HL,DE
EX      DE,HL
LD      (A26139),DE
LD      HL,A26123
LD      BC,8
CALL    READ_VRAM
POP     DE
LD      HL,0
ADD     HL,DE
EX      DE,HL
LD      HL,A26131
LD      BC,8
CALL    READ_VRAM
POP     DE
POP     BC
LD      D,B
LD      E,C
EXX
PUSH    BC
PUSH    DE
PUSH    HL
EXX
PUSH    DE
EXX
POP     DE
EXX
CALL    A25923
LD      BC,(A26139)
CALL    A26095
EXX
POP     HL
POP     DE
POP     BC
EXX
POP     IX
```

```

        POP     IY
        RET
;*****
A25686:
        LD     A,C
        LD     C,B
        LD     B,A
        LD     A,E
        LD     E,D
        LD     D,A
A25692:
        XOR    A
        LD     A,B
        CP     D
        JR     NZ,A25706
        XOR    A
        LD     A,C
        CP     E
        JR     NZ,A25706
        CALL  A25610
        RET
;*****
A25706:
        LD     A,(A17008)
        CALL  A10178
        PUSH  IX
        PUSH  IY
        PUSH  BC
        PUSH  DE
        CALL  A26072
        PUSH  DE
        LD     HL,8192
        ADD   HL,DE
        EX    DE,HL
        LD     (A26139),DE
        LD     HL,A26123
        LD     BC,8
        CALL  READ_VRAM

```



```

    POP    DE
    LD     HL,0
    ADD   HL,DE
    EX    DE,HL
    LD     HL,A26131
    LD     BC,8
    CALL  READ_VRAM
    POP   DE
    POP   BC
    RES   0,L
    RES   1,L
    RES   2,L
    LD    A,D
    SUB   B
    JR    NC,A25771
    NEG
    SET   0,L
A25771:
    LD    D,A
    LD    A,E
    SUB   C
    JR    NC,A25780
    NEG
    SET   1,L
A25780:
    LD    E,A
    SUB   D
    JR    C,A25789
    LD    A,D
    LD    D,E
    LD    E,A
    SET   2,L
A25789:
    CALL  A25812
    POP   IY
    POP   IX
    RET
;*****

```

A25797:

```
LD    D,C
LD    E,B
LD    HL,A16763
LD    B,(HL)
INC   HL
LD    C,(HL)
LD    (HL),E
DEC   HL
LD    (HL),D
CALL  A25692
RET
```

;*****

A25812:

```
EXX
PUSH  BC
PUSH  DE
PUSH  HL
EXX
PUSH  HL
PUSH  BC
EXX
POP   BC
POP   HL
EXX
LD    HL,0
LD    A,D
SRA   A
NEG
LD    C,A
LD    B,255
```

A25834:

```
CALL  A25883
INC   H
PUSH  HL
LD    L,E
LD    H,0
OR    A
```

```
ADC    HL,BC
JP     M,A25861
LD     C,D
LD     B,0
OR     A
SBC    HL,BC
LD     B,H
LD     C,L
POP    HL
INC    L
JP     A25864
```

```
;*****
```

```
A25861:
```

```
LD     B,H
LD     C,L
POP    HL
```

```
A25864:
```

```
LD     A,D
DEC    A
CP     H
JP     NC,A25834
LD     BC,(A26139)
CALL  A26095
EXX
POP    HL
POP    DE
POP    BC
EXX
RET
```

```
;*****
```

```
A25883:
```

```
PUSH  HL
EXX
POP    DE
BIT   2,L
JP     Z,A25894
LD     A,D
LD     D,E
```

```

        LD      E,A
A25894:
        BIT    1,L
        JP     Z,A25903
        LD     A,E
        NEG
        LD     E,A
A25903:
        BIT    0,L
        JP     Z,A25912
        LD     A,D
        NEG
        LD     D,A
A25912:
        LD     A,E
        ADD   A,C
        LD     E,A
        LD     A,D
        ADD   A,B
        LD     D,A
        EXX
        CALL  A25923
        RET
;*****
A25923:
        PUSH  BC
        PUSH  DE
        PUSH  HL
        EXX
        PUSH  DE
        EXX
        POP   BC
        PUSH  BC
        LD     A,(A17008)
        CP    3
        JR    Z,A25947
        LD     A,C
        CP    159

```

```
JR      C,A25947
POP     BC
JP      A7936
```

```
;*****
```

```
A25947:
```

```
LD      HL,A16763
LD      (HL),B
INC     HL
LD      (HL),C
CALL   A26072
LD      HL,8192
ADD    HL,DE
LD      BC,(A26139)
LD      A,C
CP      L
JP      NZ,A26035
LD      A,B
CP      H
JP      NZ,A26035
```

```
A25974:
```

```
POP     BC
LD      A,C
AND     7
LD      E,A
LD      D,0
LD      HL,A26123
ADD    HL,DE
LD      C,(HL)
LD      A,B
AND     7
LD      B,A
INC     B
XOR    A
SCF
```

```
A25993:
```

```
RRA
DJNZ   A25993
LD      B,A
```

```
LD      A, (A16777)
BIT     7, A
LD      A, B
JP      Z, A26012
CPL
AND     C
LD      (HL), A
JP      A26031
```

```
;*****
```

```
A26012:
```

```
OR      C
LD      (HL), A
LD      HL, A26131
ADD     HL, DE
LD      A, 15
AND     (HL)
LD      B, A
LD      A, (A16777)
RLCA
RLCA
RLCA
RLCA
OR      B
LD      (HL), A
```

```
A26031:
```

```
POP     HL
POP     DE
POP     BC
RET
```

```
;*****
```

```
A26035:
```

```
PUSH   DE
PUSH   HL
LD     (A26139), HL
CALL  A26095
POP    HL
EX     DE, HL
LD     HL, A26123
```

```
LD     BC,8
CALL  READ_VRAM
POP   DE
LD     HL,0
ADD   HL,DE
EX    DE,HL
LD     HL,A26131
LD     BC,8
CALL  READ_VRAM
JP    A25974
```

```
;*****
```

```
A26072:
```

```
LD     E,C
LD     D,0
CALL  PX_TO_PTRN_POS
LD     C,E
LD     E,B
LD     D,0
CALL  PX_TO_PTRN_POS
LD     D,C
CALL  CALC_OFFSET
EX    DE,HL
ADD   HL,HL
ADD   HL,HL
ADD   HL,HL
EX    DE,HL
RET
```

```
;*****
```

```
A26095:
```

```
LD     HL,A26123
LD     D,B
LD     E,C
PUSH  DE
LD     BC,8
CALL  WRITE_VRAM
POP   DE
LD     HL,57344      ;start of EOS-5
ADD   HL,DE
```

```

        EX     DE,HL
        LD     HL,A26131
        LD     BC,8
        CALL  WRITE_VRAM
        RET
;*****
A26123:
        DS     8           ;pattern buffer
A26131:
        DS     8           ;color buffer
A26139:
        DS     2           ;temporary pointer to VRAM
;*****
A26141:
        CALL  A29194
        LD     A,128
        LD     (A17004),A
        XOR   A
        LD     (A17006),A
        RET
;*****
A26154:
        CALL  A29194
        LD     (A17004),A
        LD     (A17006),A
        RET
;*****
A26164:
        CALL  A29194
        LD     A,128
        LD     (A17006),A
        XOR   A
        LD     (A17004),A
        RET
;*****
A26177:
        PUSH  DE
        CALL  A18355

```



```

        JP      A28138
;*****
A26184:
        PUSH   DE
        CALL   A18355
        JP     A28141
;*****
A26191:
        PUSH   HL
        PUSH   DE
        CALL   A18355
        LD     A,C
        AND    127
        LD     D,A
        LD     A,C
        RLCA
        RLCA
        RLCA
        AND    7
        ADD    A,E
        LD     E,A
        CALL   A26237
        LD     A,28
        CALL   A17242
        POP    DE
        POP    HL
        RET
;*****
A26219:
        PUSH   HL
        PUSH   DE
        CALL   A18355
        DEC    C
        LD     E,C
        CALL   A26253
        LD     A,28
        CALL   A17242
        POP    DE

```

```

        POP     HL
        RET
;*****
A26237:
        LD     A, (A69)
        CP     D
        RET    Z
        JR     NC, A26250
        LD     A, (A70)
        CP     D
A26248:
        RET    Z
        RET    NC
A26250:
        JP     A7936
;*****
A26253:
        LD     A, (A71)
        CP     E
        RET    Z
        JR     NC, A26250
        LD     A, (A72)
        CP     E
        JR     A26248
;*****
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
;*****
A26272:
        LD     B, 3
        LD     HL, 6144
        XOR    A
A26278:
        PUSH  BC

```

A26279:

```
PUSH AF
PUSH HL
LD DE,1
CALL FILL_VRAM
POP HL
POP AF
INC HL
INC A
JR NZ,A26279
POP BC
DJNZ A26278
RET
```

;*****

A26297:

```
LD A,(A17008)
CALL A10178
LD HL,A16777
SET 7,(HL)
CALL A26317
LD HL,A16777
RES 7,(HL)
RET
```

;*****

A26317:

```
LD A,(A17008)
CALL A10178
LD HL,A16763
LD D,(HL)
INC HL
LD E,(HL)
CALL A26598
RET
```

;*****

A26333:

```
LD A,C
OR A
JP NZ,A26340
```

```

        LD      A,255
A26340:
        LD      (A16765),A
        RET
;*****
A26344:
        LD      B,C
        PUSH   IX
        PUSH   IY
        LD      A,63
        AND    B
        LD      B,A
        SRA   A
        SRA   A
        SRA   A
        LD      C,A
        LD      A,(A26333)
        DEC    A
        JP     NZ,A26372
A26367:
        LD      A,240
A26368 EQU    A26367+1
        JP     A26401
;*****
A26372:
        DEC    A
        JP     NZ,A26381
        LD      A,248
        JP     A26401
;*****
A26381:
        DEC    A
        JP     NZ,A26390
        LD      A,252
        JP     A26401
;*****
A26390:
        DEC    A

```

```

        JP      NZ ,A26399
        LD      A ,254
        JP      A26401
;*****
A26399:
        LD      A ,255
A26401:
        AND     B
        AND     7
        BIT     0 ,C
        JP      Z ,A26413
        LD      B ,A
        LD      A ,8
        SUB     B
A26413:
        LD      B ,A
        LD      IX ,A16768
        LD      A ,C
        CALL    A26451
        LD      A ,2
        ADD     A ,C
        AND     7
        CALL    A26451
        LD      A ,4
        ADD     A ,C
        AND     7
        CALL    A26451
        LD      A ,6
        ADD     A ,C
        AND     7
        CALL    A26451
        POP     IY
        POP     IX
        RET
;*****
A26451:
        LD      HL ,A26464
        ADD     A ,A

```

```

        LD     E,A
        LD     D,0
        ADD    HL,DE
        LD     E,(HL)
        INC    HL
        LD     D,(HL)
        EX     DE,HL
        JP     (HL)
;*****
;ROT vector table.
;
A26464:
        DW     A26480
        DW     A26490
        DW     A26503
        DW     A26513
        DW     A26523
        DW     A26536
        DW     A26546
        DW     A26559
;*****
A26480:
        LD     (IX+0),B
        LD     (IX+1),248
        JP     A26569
;*****
A26490:
        LD     (IX+0),8
        LD     A,B
        NEG
        LD     (IX+1),A
        JP     A26569
;*****
A26503:
        LD     (IX+0),8
        LD     (IX+1),B
        JP     A26569
;*****

```

```

A26513:
    LD    (IX+0),B
    LD    (IX+1),8
    JP    A26569
;*****
A26523:
    LD    A,B
    NEG
    LD    (IX+0),A
    LD    (IX+1),8
    JP    A26569
;*****
A26536:
    LD    (IX+0),248
    LD    (IX+1),B
    JP    A26569
;*****
A26546:
    LD    (IX+0),248
    LD    A,B
    NEG
    LD    (IX+1),A
    JP    A26569
;*****
A26559:
    LD    A,B
    NEG
    LD    (IX+0),A
    LD    (IX+1),248
A26569:
    INC   IX
    INC   IX
    RET
;*****
;Default shape table.
;
A26574:
    DW    1

```

;# of

shapes in table

DW A26578-A26574

;offset to

first shape data

A26578:

DB 036H,03FH,024H,024H,02DH

DB 02DH,036H,036H,03FH

;shape

data

DB 0

;end of

shape

;*****

A26588:

LD A,(A17008)

CALL A10178

LD A,C

LD C,E

LD E,B

LD D,A

A26598:

PUSH IX

PUSH IY

PUSH BC

PUSH DE

CALL A26072

PUSH DE

LD HL,8192

ADD HL,DE

EX DE,HL

LD (A26139),DE

LD HL,A26123

LD BC,8

CALL READ_VRAM

POP DE

LD HL,0

ADD HL,DE

EX DE,HL

LD HL,A26131

LD BC,8

CALL READ_VRAM


```
POP    DE
POP    BC
PUSH   DE
LD     HL, (A16766)
LD     A, C
ADD    A, A
LD     C, A
LD     B, 0
ADD    HL, BC
LD     E, (HL)
INC    HL
LD     D, (HL)
LD     HL, (A16766)
ADD    HL, DE
POP    DE
EXX
PUSH   BC
PUSH   DE
PUSH   HL
EXX
PUSH   DE
EXX
POP    DE
EXX
LD     A, (HL)
```

A26671:

```
CP     0
JP     Z, A26724
LD     B, A
CALL   A26741
PUSH   DE
EXX
POP    BC
EXX
CALL   A26755
LD     A, B
CP     0
JP     Z, A26719
```

```

CALL    A26741
PUSH   DE
EXX
POP    BC
EXX
CALL    A26755
LD     A,B
CP     0
JP     Z,A26719
CALL    A26741
PUSH   DE
EXX
POP    BC
EXX
CALL    A26755
A26719:
INC    HL
LD     A,(HL)
JP     A26671
;*****
A26724:
LD     BC,(A26139)
CALL   A26095
EXX
POP    HL
POP    DE
POP    BC
EXX
POP    IY
POP    IX
RET
;*****
A26741:
AND    3
LD     D,A
SRL    B
SRL    B
LD     A,B

```

```

        AND    1
        LD     E,A
        SRL   B
        RET
;*****
A26755:
        PUSH  BC
        EXX
        LD    HL,A16768
        LD    A,B
        SLA  A
        ADD  A,L
        LD    L,A
        JP   NC,A26769
        INC  H
A26769:
        LD    A,(HL)
        INC  HL
        LD    L,(HL)
        LD    H,A
        LD    A,C
        SET  1,A
        LD    BC,0
        EXX
        PUSH AF
        LD    A,(A16765)
        LD    B,A
        POP  AF
A26786:
        CP    3
        JP   NZ,A26798
        PUSH AF
        CALL A25923
        POP  AF
        RES  1,A
A26798:
        EX    AF,AF'
        EXX

```

```

        LD      A,B
        ADD     A,H
        LD      B,A
        BIT     7,B
        JP      Z,A26810
        NEG
A26810:
        CP      8
        JP      C,A26831
        AND     7
        BIT     7,B
        JP      NZ,A26826
        INC     D
        JP      A26827
;*****
A26826:
        DEC     D
A26827:
        EX      AF,AF'
        SET     1,A
        EX      AF,AF'
A26831:
        BIT     7,B
        JP      Z,A26838
        NEG
A26838:
        LD      B,A
        LD      A,C
        ADD     A,L
        LD      C,A
        BIT     7,C
        JP      Z,A26849
        NEG
A26849:
        CP      8
        JP      C,A26870
        AND     7
        BIT     7,C

```

```

        JP      NZ ,A26865
        INC    E
        JP      A26866
;*****
A26865:
        DEC    E
A26866:
        EX     AF ,AF '
        SET    1 ,A
        EX     AF ,AF '
A26870:
        BIT    7 ,C
        JP      Z ,A26877
        NEG
A26877:
        LD     C ,A
        EXX
        EX     AF ,AF '
        DJNZ   A26786
        POP    BC
        RET
;*****
A26884:
        LD     A , (A17008)
        CALL   A10178
        LD     HL ,A16777
        SET    7 , (HL)
        CALL   A26588
        LD     HL ,A16777
        RES    7 , (HL)
        RET
;*****
A26904:
        PUSH   IX
        PUSH   IY
        PUSH   DE
        PUSH   BC
        CALL   A27104

```

```
LD      A,131
LD      IX,A16778
PUSH   IX
CALL   POLLER
LD      IX,A16778
LD      A,131
CALL   POLLER
POP     IX
POP     BC
LD      B,0
LD      HL,A27068
ADD    HL,BC
ADD    HL,BC
LD      A,(HL)
INC    HL
LD      H,(HL)
LD      L,A
JP     (HL)
```

```
;*****
```

```
A26948:
```

```
LD      A,(IX+1)
```

```
A26951:
```

```
OR      A
JR      Z,A27047
LD      A,1
JP      A27047
```

```
;*****
```

```
A26959:
```

```
LD      A,(IX+6)
JP      A26951
```

```
;*****
```

```
A26965:
```

```
LD      A,(IX+2)
JP      A26951
```

```
;*****
```

```
A26971:
```

```
LD      A,(IX+7)
JP      A26951
```

;*****

A26977:

LD A, (IX+3)

A26980:

ADD A, 48

CP 58

JR C, A27047

JR Z, A26997

CP 59

JR NZ, A27002

LD A, 35

JP A27047

;*****

A26997:

LD A, 42

JP A27047

;*****

A27002:

XOR A

JP A27047

;*****

A27006:

LD A, (IX+8)

JP A26980

;*****

A27012:

LD A, (IX+3)

JP A27047

;*****

A27018:

LD A, (IX+8)

JP A27047

;*****

A27024:

LD A, (IX+0)

JR A27047

;*****

A27029:

```

        LD      A, (IX+5)
        JR      A27047
;*****
A27034:
        LD      A, (IX+4)
        JR      A27047
;*****
A27039:
        LD      A, (IX+9)
        JR      A27047
;*****
A27044:
        LD      A, (A27100)
A27047:
        POP     DE
        POP     IY
        POP     IX
        RET
;*****
A27053:
        LD      A, (A27101)
        JR      A27047
;*****
A27058:
        LD      A, (A27102)
        JR      A27047
;*****
A27063:
        LD      A, (A27103)
        JR      A27047
;*****
;1st PDL vector table.  In function order 0-15.
;
A27068:
        DW      A27044;0
        DW      A27058      ;1
        DW      A27053      ;2
        DW      A27063      ;3

```



```
DW      A27024      ;4
DW      A27029      ;5
DW      A26948      ;6
DW      A26959      ;7
DW      A26965      ;8
DW      A26971      ;9
DW      A26977      ;10
DW      A27006      ;11
DW      A27012      ;12
DW      A27018      ;13
DW      A27034      ;14
DW      A27039      ;15
```

```
;*****
```

```
;PDL data for counters. 0-255, 128 is center.
```

```
;
```

```
A27100:
```

```
        DB      128      ;player 2 joystick
forward-backward counter
```

```
A27101:
```

```
        DB      128      ;player 2 joystick right-left
counter
```

```
A27102:
```

```
        DB      128      ;player 1 joystick
forward-backward counter
```

```
A27103:
```

```
        DB      128      ;player 1 joystick right-left
counter
```

```
;*****
```

```
A27104:
```

```
        PUSH    IX
        PUSH    IY
        LD      IX,A16778
        PUSH    IX
        LD      A,3
        CALL    POLLER
        LD      IX,A16778
        LD      A,3
        CALL    POLLER
```

```

        POP     IX
        LD      A, (IX+0)
        LD      HL, A27355
A27136:
        PUSH   BC
        LD      B, 0
        LD      C, A
        ADD    HL, BC
        ADD    HL, BC
        POP    BC
        LD      A, (HL)
        INC   HL
        LD      H, (HL)
        LD      L, A
        JP     (HL)
;*****
A27148:
        LD      A, (IX+5)
        LD      HL, A27381
        JP     A27136
;*****
A27157:
        POP    IY
        POP    IX
        RET
;*****
A27162:
        LD      HL, A27100
        CALL   A27347
        JP     A27148
;*****
A27171:
        LD      HL, A27100
        CALL   A27338
        JP     A27148
;*****
A27180:
        LD      HL, A27101

```

```
        CALL    A27338
        JP      A27148
;*****
A27189:
        LD      HL,A27101
        CALL    A27347
        JP      A27148
;*****
A27198:
        LD      HL,A27100
        CALL    A27347
        INC     HL
        CALL    A27338
        JP      A27148
;*****
A27211:
        LD      HL,A27100
        CALL    A27347
        INC     HL
        CALL    A27347
        JP      A27148
;*****
A27224:
        LD      HL,A27100
        CALL    A27338
        INC     HL
        CALL    A27338
        JP      A27148
;*****
A27237:
        LD      HL,A27100
        CALL    A27338
        INC     HL
        CALL    A27347
        JP      A27148
;*****
A27250:
        LD      HL,A27102
```

```

        CALL    A27347
        JP      A27157
;*****
A27259:
        LD      HL,A27102
        CALL    A27338
        JP      A27157
;*****
A27268:
        LD      HL,A27103
        CALL    A27338
        JP      A27157
;*****
A27277:
        LD      HL,A27103
        CALL    A27347
        JP      A27157
;*****
A27286:
        LD      HL,A27102
        CALL    A27347
        INC     HL
        CALL    A27338
        JP      A27157
;*****
A27299:
        LD      HL,A27102
        CALL    A27347
        INC     HL
        CALL    A27347
        JP      A27157
;*****
A27312:
        LD      HL,A27102
        CALL    A27338
        INC     HL
        CALL    A27338
        JP      A27157

```

```

;*****
A27325:
    LD     HL,A27102
    CALL  A27338
    INC   HL
    CALL  A27347
    JP    A27157
;*****
A27338:
    LD     A,1
    ADD   A,(HL)
    JR    NC,A27345
    LD     A,255
A27345:
    LD     (HL),A
    RET
;*****
A27347:
    LD     A,(HL)
    SUB   1
    JR    NC,A27353
    XOR   A
A27353:
    LD     (HL),A
    RET
;*****
;2nd PDL vector table.
;
A27355:
    DW    A27148;
    DW    A27162    ;
    DW    A27180    ;
    DW    A27198    ;
    DW    A27171    ;
    DW    A27148    ;
    DW    A27224    ;
    DW    A27148    ;
    DW    A27189    ;

```

```

        DW      A27211      ;
        DW      A27148      ;
        DW      A27148      ;
        DW      A27237      ;
A27381:
        DW      A27157      ;
        DW      A27250      ;
        DW      A27268      ;
        DW      A27286      ;
        DW      A27259      ;
        DW      A27157      ;
        DW      A27312      ;
        DW      A27157      ;
        DW      A27277      ;
        DW      A27299      ;
        DW      A27157      ;
        DW      A27157      ;
        DW      A27325      ;

;*****
;begin SmartBASIC 1.x interpreter segment
;*****

A27407:
        CP      67
        JR      C,A27416
        SUB     67
        LD      HL,A27900
A27416:
        ADD     A,A
        LD      B,0
        LD      C,A
        RET
;*****
A27421:
        LD      HL,A272
        CP      67
        RET     C

```

```

        LD     HL,A27486
        RET
;*****
A27431:
        LD     HL,A818
        CP     190
        RET    C
        LD     HL,A27891
        RET
;*****
A27441:
        LD     HL,A27486
        LD     A,E
        DEC    A
        JR     NZ,A27451
        LD     HL,A27891
A27451:
        LD     A,(HL)
        OR     A
        JP     Z,A13822
        ADD    HL,DE
        LD     C,(HL)
        INC    HL
        EX    DE,HL
        EX    (SP),HL
        CALL  A11851
        EX    (SP),HL
        EX    DE,HL
        JP     Z,A13803
        LD     A,C
        ADD    A,L
        LD     L,A
        JR     NC,A27451
        INC    HL
        JR     A27451
;*****
A27477:
        LD     A,32

```

```
CALL A11994
LD HL,A16246
RET
```

```
;*****
```

```
;Second primary word table. Format: token, parse
pointer, length, word.
```

```
;
```

```
A27486:
```

```
DB 67
DW A953
DB 2,7,2 ;beep/sadface
```

```
;
```

```
DB 68
DW A1040
DB 2
DB "AF"
```

```
;
```

```
DB 69
DW A1040
DB 2
DB "BC"
```

```
;
```

```
DB 70
DW A1040
DB 2
DB "DE"
```

```
;
```

```
DB 71
DW A1040
DB 2
DB "HL"
```

```
;
```

```
DB 72
DW A1040
DB 2
DB "IX"
```

```
;
```

```
DB 73
```



```
DW      A1040
DB      2
DB      "IY"
;

DB      74
DW      A10824
DB      4
DB      "ADDR"
;

DB      75
DW      A10824
DB      3
DB      "MEM"
;

DB      76
DW      A1040
DB      6
DB      "PROMPT"
;

DB      77
DW      A1040
DB      7
DB      "PRWIDTH"
;

DB      78
DW      A976
DB      4
DB      "BEEP"
;

DB      79
DW      A976
DB      3
DB      "CLS"
;

DB      80
DW      A28606
DB      5
DB      "WHILE"
```

```
;
DB      81
DW      A976
DB      4
DB      "WEND"
;
DB      82
DW      A29149
DB      9
DB      "RANDOMIZE"
;
DB      83
DW      A28369
DB      5
DB      "RENUM"
;
DB      84
DW      A953
DB      6
DB      "SPSIZE"
;
DB      85
DW      A953
DB      5
DB      "SPDEF"
;
DB      86
DW      A953
DB      6
DB      "SPDRAW"
;
DB      87
DW      A953
DB      5
DB      "SOUND"
;
DB      88
DW      A28227
```

```
DB      4
DB      "TIME "
;
DB      89
DW      A28238
DB      4
DB      "DATE "
;
DB      90
DW      A28369
DB      6
DB      "BLREAD "
;
DB      91
DW      A28369
DB      7
DB      "BLWRITE "
;
DB      92
DW      A976
DB      6
DB      "TEXT31 "
;
DB      93
DW      A976
DB      6
DB      "TEXT40 "
;
DB      94
DW      A1494
DB      3
DB      "PUT "
;
DB      95
DW      A983
DB      6
DB      "LINPUT "
;
```

```
DB 96
DW A28437
DB 6
DB "LOCATE"
;
DB 97
DW A28437
DB 3
DB "OUT"
;
DB 98
DW A976
DB 6
DB "TEXT80"
;
DB 99
DW A953
DB 7
DB "SPXDRAW"
;
DB 100
DW A28369
DB 6
DB "SERIAL"
;
DB 101
DW A1045
DB 3
DB "PRN"
;
DB 102
DW A10824
DB 3
DB "SER"
;
DB 103
DW A10824
DB 5
```

```
DB      "DSIZE"  
;  
DB      104  
DW      A1040  
DB      3  
DB      "CLK"  
;  
DB      105  
DW      A1040  
DB      4  
DB      "TERM"  
;  
DB      106  
DW      A976  
DB      5  
DB      "MERGE"  
;  
DB      107  
DW      A976  
DB      7  
DB      "NOMERGE"  
;  
DB      108  
DW      A31576  
DB      6  
DB      "WINDOW"  
;  
DB      109  
DW      A976  
DB      6  
DB      "CLREAD"  
;  
DB      110  
DW      A976  
DB      7  
DB      "CLWRITE"  
;  
DB      111
```

```

        DW      A1494
        DB      6
        DB      "FORMAT"
;
        DB      0                ;end of table
;*****
        DS      13                ;fill out to end
                                   ;(some reserved space)
;*****
;Second secondary word table.  Format:  token,
length, word.
;
A27891:
        DB      190
        DB      3
        DB      "EOS"
;
        DB      191
        DB      1
        DB      92                ;\
;
        DB      0                ;end of table
;*****
;Second primary word vector table.
;
A27900:
        DW      A7948                ;67 beep/sadface
        DW      A30096                ;68 AF
        DW      A30103                ;69 BC
        DW      A30110                ;70 DE
        DW      A30117                ;71 HL
        DW      A30124                ;72 IX
        DW      A30131                ;73 IY
        DW      A30073                ;74 ADDR
        DW      A30057                ;75 MEM
        DW      A11253                ;76 PROMPT
        DW      A31206                ;77 PRWIDTH
        DW      A11260                ;78 BEEP

```

DW	A11090	;79	CLS
DW	A28631	;80	WHILE
DW	A28679	;81	WEND
DW	A29115	;82	RANDOMIZE
DW	A8419	;83	RENUM
DW	A8419	;84	SPSIZE
DW	A8419	;85	SPDEF
DW	A8419	;86	SPDRAW
DW	A8419	;87	SOUND
DW	A28087	;88	TIME
DW	A28150	;89	DATE
DW	A28338	;90	BLREAD
DW	A28355	;91	BLWRITE
DW	A31419	;92	TEXT31
DW	A31435	;93	TEXT40
DW	A28444	;94	PUT
DW	A28761	;95	LINPUT
DW	A28380	;96	LOCATE
DW	A30538	;97	OUT
DW	A34455	;98	TEXT80
DW	A8419	;99	SPXDRAW
DW	A11726	;100	SERIAL
DW	A31017	;101	PRN
DW	A31147	;102	SER
DW	A31818	;103	DSIZE
DW	A29291	;104	CLK
DW	A32312	;105	TERM
DW	A29899	;106	MERGE
DW	A1138	;107	NOMERGE
DW	A31591	;108	WINDOW
DW	A32900	;109	CLREAD
DW	A33606	;110	CLWRITE
DW	A35234	;111	FORMAT
DW	A0	;112	unused
DW	A0	;113	unused
DW	A0	;114	unused
DW	A0	;115	unused
DW	A0	;116	unused

```

;*****
;TEXT31 and TEXT40 offset routine patch.
;
A28000:
    ADD    HL,HL            ;*8
    PUSH  BC                ;save BC
    PUSH  HL                ;save HL*8
    ADD   HL,HL            ;*16
    ADD   HL,HL            ;*32
    POP   BC                ;get HL*8
    LD    A, (A17008)       ;get screen mode
    OR    A                 ;is it TEXTxx?
    JR    NZ,A28021        ;NO, so ignore
    LD    A, (A17988)       ;YES, so get TEXTxx width
    CP    40                ;is it TEXT40?
    JR    NZ,A28021        ;NO, so no more fixup needed
    ADD   HL,BC             ;YES, so add 8 to make 40
    DEC   HL                ;-1
A28021:
    POP   BC                ;restore BC
    RET
;*****
;Part of TEXT31.
;
A28023:
    XOR   A                 ;A=0
    LD    (A17177),A
    LD    (A17166),A
    LD    A,32
    LD    (A17988),A
    LD    A,224
    LD    (A17215),A
    LD    A,30              ;width-1
A28042:
    LD    (A17199),A       ;save screen width
    JP    A11065           ;finish it out
;*****
;Part of TEXT40 command.

```



```

;
A28048:
    LD    A,192
    LD    (A17177),A
    LD    (A17166),A
    LD    A,40
    LD    (A17988),A
    LD    A,240
    LD    (A17215),A
    LD    A,39          ;width-1
    JR    A28042        ;finish it out
;*****
;SERIAL initialization data for Eve/Orphanware and
ADAMlink modem.
;
A28070:
    DB    122          ;Eve/Orphanware E71
A28071:
    DB    78           ;Eve Orphanware N81
A28072:
    DB    251         ;ADAMlink E71
A28073:
    DB    79          ;ADAMlink N81
;*****
;SERIAL temporary data.
;
A28074:
    DS    1           ;port (actual)
A28075:
    DS    2           ;baudrate (as word)
A28077:
    DS    1           ;baudrate (actual data)
A28078:
    DS    1           ;stats (actual data)
A28079:
    DS    1           ;stats (as 0 or 1)
;*****
A28080:

```

```

        DB      68          ;PR #3 port
A28081:
        DB      84          ;TEXT80 port
;*****
;Skip token in crunch code.
;
A28082:
        EXX
        DEC     C           ;one less byte remaining
        EXX
        INC     DE          ;point to next crunch code
byte
        RET
;*****
;TIME hh:mm:ss command.
;
A28087:
        CALL   A1500
        LD     A,L
        CP     24
        JR     NC,A28147
        PUSH  AF
        CALL  A28082
        CALL  A1500
        LD     A,L
        CP     60
        JR     NC,A28147
        PUSH  AF
        CALL  A28082
        CALL  A1500
        LD     A,L
        CP     60
        JR     NC,A28147
        PUSH  AF
        CALL  A31952        ;some TIME patch
        POP   AF
        NOP
A28125:

```

```

        LD     L,A
        LD     H,0
        LD     (A100),HL
        POP   HL
        POP   AF
        LD     L,A
        LD     (A98),HL
        RET
;*****
A28138:
        LD     A,D
A28139:
        POP   DE
        RET
;*****
A28141:
        LD     A,E
        INC   A
        JR    A28139
;*****
        NOP
        NOP
;*****
A28147:
        JP    A7936           ;Illegal Quantity error
;*****
;DATE mm\dd\yyyy\wd command.
;
A28150:
        CALL  A1500
        LD   A,L
        OR   A
        JR   Z,A28147
        CP   13
        JR   NC,A28147
        PUSH AF
        CALL A28082
        CALL A1500

```

```
POP    AF
PUSH   AF
PUSH   HL
LD     C,A
LD     B,0
LD     HL,A86-1
ADD    HL,BC
LD     B,(HL)
POP    HL
LD     A,L
CP     B
JR     NC,A28147
OR     A
JR     Z,A28147
PUSH   AF
CALL   A28082
CALL   A5939
OR     A
JP     NZ,A7939
CALL   A2354
JR     C,A28147
LD     BC,1900
OR     A
SBC    HL,BC
JR     C,A28147
LD     A,H
OR     A
JR     NZ,A28147
LD     A,L
```

A28216:

```
LD     HL,EOS_YEAR
LD     (HL),A
POP    HL
POP    AF
LD     L,A
CALL   A32189
RET
```

;deal with weekday

;*****

```

;TIME parse vector table.
;
A28227:
    DB      5
    DW      A14875      ;numeric equation
    DW      A15911      ;colon
    DW      A14875      ;numeric equation
    DW      A15911      ;colon
    DW      A14875      ;numeric equation
;*****
;DATE parse vector table.
;
A28238:
    DB      5
    DW      A14875      ;numeric equation
    DW      A28249      ;backslash
    DW      A14875      ;numeric equation
    DW      A28249      ;backslash
    DW      A32300      ;yy/wd
;*****
;Parse \.
;
A28249:
    LD      A,191      ;token for "\"
    CALL    A14581      ;is it found?
    RET     C          ;YES
    CALL    A11921      ;NO, so print "'\' Expected"
;*****
    DB      3
    DB      " '"
    DB      92          ;\
    DB      " '"
;*****
;Block I/O setup for BLREAD and BLWRITE.
;Remember, BLxxxx drive,block,DTA.
;On exit, A=EOS device #, HL=DTA, BC=block number.
;
A28262:

```

```

CALL A1500 ;get numeric argument 0-255
in HL (drive)
LD A,L
OR A ;is it >255?
JP Z,A7936 ;YES, so Illegal Quantity
error
CP 8 ;NO, but is it <8?
JP NC,A7936 ;NO, so Illegal Quantity
error
LD C,L
LD B,H ;BC=drive offset
LD HL,A23272 ;drive-to-device table-1
ADD HL,BC ;offset into table
LD A,(HL) ;get EOS device #
PUSH AF ;save it
PUSH BC ;save offset value
CALL A30089 ;skip token (,) and get
block in HL
POP BC ;restore offset
PUSH HL ;save block
LD HL,A31260 ;drive size table
ADD HL,BC
ADD HL,BC ;offset into table
LD A,(HL)
INC HL
LD H,(HL)
LD L,A ;get size in HL
DEC HL ;-1 for 0-base
POP BC ;restore block
OR A ;clear CF
SBC HL,BC ;requested block too big for
medium?
JP C,A7936 ;YES, so Illegal Quantity
error
PUSH BC ;NO, so save requested block
CALL A30089 ;skip token (,) and get DTA
in HL
POP BC ;restore block

```

```

        POP    AF                ;restore EOS device #
        RET
;*****
;SER(n) stats table for Eve/Orphanware and ADAMlink
modem.
;Each entry 5 bytes.
;
A28313:
        DS     25
;*****
A28338:
        CALL   A28262
        PUSH  DE
        LD    E,C
        LD    D,B
        LD    BC,0
        CALL  _READ_BLOCK
A28350:
        POP   DE
        RET   Z
        JP   A24297
;*****
A28355:
        CALL   A28262
        PUSH  DE
        LD    E,C
        LD    D,B
        LD    BC,0
        CALL  _WRITE_BLOCK
        JR    A28350
;*****
;BLREAD/BLWRITE/SERIAL/RENUM parse vector table.
;
A28369:
        DB     5
        DW    A14875        ;numeric equation
        DW    A15939        ;comma
        DW    A14875        ;numeric equation

```

```

        DW      A15939          ;comma
        DW      A14875          ;numeric equation
;*****
;LOCATE line,column command.
;
A28380:
        CALL    A1500           ;get line 0-255 in L
        DEC     L               ;-1
        LD      A,(A71)         ;get max right margin
        CP      L               ;is it <=?
        JR      Z,A28392        ;YES equal
        JR      NC,A28434       ;NO greater than, so Illegal
Quantity error
A28392:
        LD      A,(A72)         ;get min left margin
        CP      L               ;is it >=?
        JR      Z,A28400        ;YES equal
        JR      C,A28434       ;NO less than, so Illegal
Quantity error
A28400:
        PUSH    HL              ;save line in L
        CALL    A28082          ;skip token (,)
        CALL    A1500           ;get column 0-255 in HL
        LD      A,(A69)         ;get max bottom margin
        CP      L               ;is it <=?
        JR      Z,A28415        ;YES equal
        JR      NC,A28434       ;NO greater than, so Illegal
Quantity error
A28415:
        LD      A,(A70)         ;get min top margin
        CP      L               ;is it >=?
        JR      Z,A28423        ;YES equal
        JR      C,A28434       ;NO less than, so Illegal
Quantity error
A28423:
        POP     BC              ;restore line in C
        PUSH    DE              ;save crunch code position
        LD      E,C             ;E=line

```



```

        LD      D,L          ;D=column
        LD      A,28        ;quickmove cursor character
        CALL   A17234       ;print character in A on
screen
        POP    DE          ;restore crunch code
position
        RET
;*****
A28434:
        JP     A7936        ;Illegal Quantity error
;*****
;LOCATE parse vector table.
;
A28437:
        DB     3
        DW    A14875       ;numeric equation
        DW    A15939       ;comma
        DW    A14875       ;numeric equation
;*****
;PUT value command.
;
A28444:
        CALL   A5939       ;evaluate equation
        OR     A           ;was it a string?
        JR     NZ,A28471   ;YES, so check its length
        CALL   A2354       ;move FPA1 to HL
        JR     C,A28484    ;bigger than 65535, so
Illegal Quantity error
        LD     A,H
        OR     H           ;is it<255?
        JR     NZ,A28484   ;NO, so Illegal Quantity
error
        LD     A,L        ;YES, so get character in A
A28460:
        PUSH  AF          ;save character
        LD     A,255      ;set PUT toggle
        LD     (A1497),A  ;save it
        POP   AF          ;restore character

```

```

        CALL  A11994          ;print it via current PR#
        RET
;*****
A28471:
        LD    HL, (A16162)   ;get address of string in
string space
        INC   HL
        INC   HL             ;skip address in variable
table
        LD    A, (HL)        ;get length
        DEC   A              ;is it 1?
        JR    NZ, A28484     ;NO, so Illegal Quantity
error
        INC   HL             ;YES, so point to data byte
        LD    A, (HL)        ;get it
        JR    A28460         ;print it
;*****
A28484:
        JP    A7936          ;Illegal Quantity error
;*****
;PRINT with ^D patch for PUT.
;On entry, A=character to print.
;
A28487:
        PUSH  AF             ;save character
        LD    A, (A1497)     ;get PUT toggle
        INC   A              ;is PUT active?
        LD    A, 0
        LD    (A1497), A     ;clear PUT toggle regardless
        JR    Z, A28508     ;YES, PUT is active, so do
regular print
        POP   AF             ;NO, PUT is inactive, so
restore character...
        CP    4              ;...and check was it ^D?
        RET   NZ             ;NO, so don't worry
        INC   SP             ;YES, so get rid of old
return address
        INC   SP

```

```

        JP      A19538          ;do PRINT with ^D
;*****
A28508:
        POP    AF              ;restore character
        RET
;*****
;MON O/NOMON O patch for PUT.
;On entry, A=character to print.
;
A28510:
        PUSH   AF              ;save character
        LD     A,(A1497)        ;get PUT toggle
        INC    A                ;is PUT active?
        LD     A,0
        LD     (A1497),A        ;clear PUT toggle regardless
        JR     Z,A28531         ;YES, PUT is active, so
divert to print all ctrl chars
        POP    AF              ;NO, PUT is inactive, so
restore character...
        CP     4                ;...and check was it ^D?
        RET    Z                ;YES, so do regular print
A28526:
        INC    SP              ;bypass old return address
        INC    SP
        JP     A21888           ;print all control
characters
;*****
A28531:
        POP    AF              ;restore character
        JR     A28526           ;clear stack and print
;*****
A28534:
        DW     A28612           ;WHILE/WEND stack pointer

A28535      EQU     A28534+1

        DS     70                ;WHILE/WEND stack
; 10 entries, each 7 bytes

```

```

long:
                                ; HL' (pointer to line #)
                                ; DE' (start of line crunch
code)
                                ; C' (bytes left in crunch
code condition)
                                ; DE (crunch code
position--points at WHILE token)

A28612      EQU      A28534+78      ;78=2+70+7-1

;*****
;WHILE parse vector table.
;
A28606:
        DB      1
        DW      A14947
;*****
;SmartBASIC stack setup patch for WHILE/WEND.
;
A28609:
        LD      HL,A28612      ;top of WHILE/WEND stack
        LD      (A28534),HL ;save it to WHILE/WEND stack
pointer
        POP     HL
        LD      SP,SB1X_STACK      ;reset SmartBASIC
stack pointer
        JP      A5979      ;finish out routine
;*****
;Find line # address patch.
;
A28622:
        LD      HL,A28612      ;top of WHILE/WEND stack
        LD      (A28534),HL ;save it to WHILE/WEND stack
pointer
        JP      A6063      ;finish out routine
;*****
;WHILE {condition} command.

```

```

;
A28631:
    LD     HL,(A28534) ;get WHILE/WEND stack
pointer
    LD     BC,7        ;length of each entry on
stack
    OR     A           ;clear CF
    SBC   HL,BC       ;back up to 1st entry
    PUSH  HL          ;save it
    LD     BC,A28535  ;bottom of WHILE/WEND stack
    OR     A           ;clear CF
    SBC   HL,BC       ;did we overflow the stack?
    POP   HL          ;restore 1st entry address
    JP    Z,A7306     ;YES, so Stack Overflow
error
    LD     (A28534),HL ;NO, so save new WHILE/WEND
stack pointer
    LD     (HL),D      ;and all other registers
needed for WEND
    DEC   HL
    LD     (HL),E
    DEC   HL          ;save DE (crunch code
position)
    EXX
    PUSH  HL
    PUSH  DE
    PUSH  BC
    EXX
    POP   BC
    LD     (HL),C
    DEC   HL          ;save C' (bytes left in
crunch code)
    POP   BC
    LD     (HL),B
    DEC   HL
    LD     (HL),C
    DEC   HL          ;save DE' (start of crunch
code line)

```

```

        POP    BC
        LD     (HL),B
        DEC   HL
        LD     (HL),C           ;save HL' (pointer to line
#)
        CALL  A5939           ;evaluate equation (to skip
over condition)
        RET
;*****
;WEND command.
;(1) check for "WEND without WHILE" error
;(2) save current crunch code position on SmartBASIC
stack (HL',DE',C',DE)
;(3) get WHILE condition code position registers
from WHILE/WEND stack
;(4) evaluate condition:
;   TRUE:  clear SmartBASIC stack, continue
executing next statement
;           after WHILE (does a RET)
;   FALSE: reset WHILE/WEND stack pointer (add
7), get crunch code
;           position off SmartBASIC stack
(HL',DE',C',DE) and RET
;
A28679:
        LD     HL,(A28534) ;get WHILE/WEND stack
pointer
        LD     BC,A28612   ;top of WHILE/WEND stack
        OR     A           ;clear CF
        SBC   HL,BC       ;was there a WHILE?
        LD     A,22        ;WEND without WHILE error
code
        JP     Z,A7950     ;NO, so error
        EXX              ;YES, so save current crunch
code position
        PUSH  HL           ;HL' (pointer to line #)
        PUSH  DE           ;DE' (start of crunch code)
        PUSH  BC           ;B'=status byte, C'=bytes

```

left in crunch code

```
    EXX
    PUSH  DE                ;DE position along crunch
code
    LD    HL, (A28534)      ;get current WHILE/WEND
stack pointer
    LD    D, (HL)
    DEC  HL
    LD    E, (HL)
    DEC  HL                ;restore DE
    LD    B, (HL)          ;get C'
    PUSH BC                ;save it
    DEC  HL
    LD    B, (HL)
    DEC  HL
    LD    C, (HL)          ;get DE'
    PUSH BC                ;save it
    DEC  HL
    LD    B, (HL)
    DEC  HL
    LD    C, (HL)          ;get HL'
    PUSH BC                ;save it
    EXX
    POP  HL                ;restore HL'
    POP  DE                ;restore DE'
    POP  AF
    LD    C, A              ;restore C'
    EXX
    CALL A5939              ;evaluate equation--the
WHILE condition
    OR   A                  ;was it TRUE?
    JR   NZ, A28739         ;YES TRUE, so continue with
next statement after WHILE
    LD   A, (A16166)        ;maybe...
    OR   A                  ;is it FALSE?
    JR   Z, A28744         ;YES FALSE, so continue with
next statement after WEND
    DEC  A                  ;maybe...
```

```

        JR      Z,A28744      ;YES FALSE
A28739:
        POP    AF            ;TRUE, so clear stack and
exit
        POP    AF
        POP    AF
        POP    AF
        RET
;*****
A28744:
        LD     HL,(A28534) ;FALSE, so get old
WHILE/WEND stack pointer
        LD     BC,7          ;length of entry
        ADD    HL,BC        ;offset to next (clears
WHILE/WEND stack)
        LD     (A28534),HL ;save new WHILE/WEND stack
pointer
        POP    DE            ;restore DE
        EXX
        POP    BC            ;restore BC'
        POP    DE            ;restore DE'
        POP    HL            ;restore HL'
        EXX
        RET
;*****
;LINPUT command entry.
;
A28761:
        LD     A,255         ;toggle LINPUT
        LD     (A1498),A    ;save toggle
        JP     A8957        ;finish INPUT/LINPUT
;*****
;INPUT/LINPUT command breakin.
;
A28769:
        LD     (A16147),A
        LD     A,(A1498)    ;get INPUT/LINPUT toggle
        INC    A            ;is it INPUT?

```



```

        JP      NZ,A9055      ;YES, so finish INPUT
        CALL   A6555         ;NO, it's LINPUT, so branch
to vectors.
;*****
;LINPUT vector table.
;Only strings are allowed; all others are Type
Mismatch error.
;
A28782:
        DW      A7942         ;FP
        DW      A7942         ;%
        DW      A28792        ;$
        DW      A7942         ;FN
        DW      A7942         ;math command
;*****
;LINPUT a string.
;
A28792:
        POP     AF
        POP     HL
        PUSH    BC
        PUSH    HL
        JR      NC,A28805
A28798:
        LD      A,(HL)        ;get character in string
        OR      A             ;is it end of LINPUT line?
        JR      Z,A28805      ;YES, so continue
        INC     HL            ;NO, so point to next
        JR      A28798        ;keep going 'til we find the
end
;*****
A28805:
        LD      (A16137),DE
        POP     DE
        OR      A
        SBC    HL,DE
        LD      A,L
        POP     BC

```

```

LD      H,B
LD      L,C
CALL    A7264
LD      A,L
LD      (BC),A
INC     BC
LD      A,H
LD      (BC),A
INC     HL
INC     HL
LD      A,(HL)
INC     HL
OR      A
EX      DE,HL
JR      Z,A28838
LD      C,A
LD      B,0
LDIR                    ;copy string
A28838:
LD      B,H
LD      C,L
LD      DE,(A16137)
EXX
LD      A,C                ;get bytes remaining in
crunch code
EXX
OR      A                ;are we at the end?
JR      Z,A28856         ;YES, so restore toggle to
INPUT and exit
CALL    A28082          ;NO, so skip token (,)
JP      A9002           ;get next data
;*****
A28856:
XOR     A                ;A=0
LD      (A1498),A       ;reset toggle to INPUT
RET
;*****
;Error handler breakin.

```

```

;
A28861:
    LD     (A16128),A    ;save error
    CP     22            ;was it 22?
    JR     Z,A28882      ;YES, so print "WEND without
WHILE"
    CP     26            ;was it 26?
    JR     Z,A28887      ;YES, so print "RESUME
without ONERR"
    CP     18            ;was it 18?
    JP     NZ,A75        ;NO, so check out some other
errors
    LD     HL,A28936     ;YES, so print "INPUT Data"
    JR     A28890
;*****
;Print "WEND without WHILE".
;
A28882:
    LD     HL,A28896     ;string address
    JR     A28890        ;print it
;*****
A28887:
    LD     HL,A28915     ;string address
A28890:
    CALL  A12110         ;print string at HL
    JP     A8024         ;return to rest of error
handler
;*****
A28896:
    DB     18
    DB     "WEND without WHILE"
A28915:
    DB     20
    DB     "RESUME without ONERR"
A28936:
    DB     10
    DB     "INPUT Data"
;*****

```

```

NOP
NOP
;*****
A28949:
    DS      2      ;stored SP for RESUME
A28951:
    DS      2      ;stored IX for RESUME
A28953:
    DS      2      ;stored IY for RESUME
;*****
;Execute loop patch for RESUME.
;
A28955:
    EXX
    BIT     0,B      ;is ONERR enabled?
    EXX
    JR      Z,A28979 ;NO, so don't save pointers
    EXX
    BIT     1,B      ;is ONERR executing?
    EXX
    JR      NZ,A28979 ;YES, so don't save pointers

    LD      (A28949),SP ;save SP
    LD      (A28951),IX ;save IX
    LD      (A28953),IY ;save IY
A28979:
    EXX
    DEC     C      ;one less byte of crunch
code
    BIT     4,B      ;is it NOBREAK? (set)
    JP      A6195   ;back to execute loop.
;*****
;RESUME or RESUME line# command.
;
A28986:
    EXX
    BIT     6,B      ;is it immediate mode?
    JR      Z,A29056 ;YES, so Illegal Mode error

```

```

        BIT    1,B          ;is ONERR executing?
        JR     Z,A29060     ;NO, so RESUME without ONERR
error
        RES    1,B          ;YES, so disable ONERR
        EXX
        LD     SP,(A28949)  ;get saved SP
        LD     IX,(A28951)  ;get saved IX
        LD     IY,(A28953)  ;get saved IY
        EXX
        LD     A,C          ;get bytes left in crunch
code
        EXX
        OR     A             ;any more?
        JR     Z,A29043     ;NO, so RESUME
        LD     HL,A6216     ;YES, so RESUME line#
        PUSH  HL            ;save return address
        CALL  A28082        ;skip token (points to line#
crunch code)
        LD     A,(DE)       ;get token
        CALL  A1519         ;evaluate line number into
BC
        LD     (A16207),BC  ;save line number
        CALL  A12528        ;look for line #
        JP    C,A8360       ;FOUND
        CALL  A8109         ;NOT FOUND, so do CLRERR
        JP    A8376         ;error exit to Undefined
Statement error
;*****
A29043:
        EXX                ;regular RESUME
        LD     HL,(A16124)
        EXX
        LD     DE,(A16122)
        DEC   DE
        JP    A6216
;*****
A29056:
        EXX

```

```

        JP      A7915          ;Illegal Mode error
;*****
A29060:
        EXX
        LD      A,26          ;RESUME without ONERR error
        JP      A7950
;*****
;RETURN or RETURN line# command.
;
A29066:
        EXX
        LD      A,C          ;get bytes left in crunch
code
        EXX
        OR      A            ;any arguments?
        JR      NZ,A29079    ;YES, so RESUME line#
        POP     DE          ;NO, so regular RESUME
        EXX
        SET    6,B          ;force into program mode
        JP      A8486
;*****
A29079:
        POP     HL          ;clear old crunch code DE
        POP     HL          ;clear old current line #
        POP     IX
        POP     IY          ;restore pointers
        JP      A8342      ;do GOTO line#
;*****
;RESTORE or RESTORE line# command.
;
A29088:
        EXX
        LD      A,C          ;get bytes left in crunch
code
        EXX
        OR      A            ;any arguments?
        JP      Z,A9482     ;NO, so do regular RESUME
        CALL   A28082      ;YES, so skip token (point

```

```

to line# argument)
    LD      A, (DE)          ;get token
    CALL    A1519           ;evaluate line # into BC
    LD      (A16207), BC    ;save it
    CALL    A12528         ;look for line #
    JP      NC, A8376       ;NOT found, so Undefined
Statement error
    JP      A9485           ;FOUND, so do rest of
RESTORE
;*****
;RANDOMIZE or RANDOMIZE seed1, seed2
;
A29115:
    EXX
    LD      A, C            ;get bytes left in crunch
code
    EXX
    OR      A               ;any arguments?
    JR      NZ, A29133      ;YES, so do RANDOMIZE
seed1, seed2
    LD      HL, A16190      ;NO, so do regular RANDOMIZE
    LD      B, 4            ;4 bytes needed
A29126:
    LD      A, R            ;get memory refresh register
    LD      (HL), A         ;save seed byte
    INC     HL              ;point to next
    DJNZ   A29126          ;do all 4
    RET
;*****
A29133:
    CALL    A9987           ;get seed1 from FPA1 to HL
    LD      (A16190), HL    ;save seed1
    CALL    A28082         ;skip token (,)
    CALL    A9987           ;get seed2 from FPA1 to HL
    LD      (A16192), HL    ;save seed2
    RET
;*****
;RANDOMIZE parse vector table.

```

```

;
A29149:
        DB      1
        DW      A30525
;*****
;Show cursor during Input Line patch.
;
A29152:
        LD      A, (A17008)    ;screen mode
        OR      A              ;is it TEXTxx?
        LD      A, 95          ;underline cursor
        JR      Z, A29162      ;YES, TEXTxx so keep
underline
        LD      A, 223         ;NO, so bias +128 for GR/HGR
mode
A29162:
        LD      (A16953), A    ;save cursor
        LD      A, (A17006)    ;rest of breakin
        RET
;*****
;Hide cursor after Input Line patch.
;
A29169:
        LD      (A17006), A    ;rest of breakin
A29172:
        LD      A, (A17008)    ;screen mode
        OR      A              ;is it TEXTxx?
        LD      A, 0           ;ASCII base to hide cursor
        JR      Z, A29182      ;YES, TEXTxx so keep 0
        LD      A, 160         ;NO, so use 160 in GR/HGR
mode
A29182:
        LD      (A16953), A    ;save cursor
        RET
;*****
;PLOT/SCRN in GR only patch.
;
A29186:

```



```

        LD      A, (A17008)    ;screen mode
        DEC    A                ;was it 1? (GR)
        RET    Z                ;YES okay
        JP     A7915           ;NO, so Illegal Mode error
;*****
;INVERSE/NORMAL/FLASH mode check.
;
A29194:
        LD      A, (A17008)    ;screen mode
        OR     A                ;is it TEXTxx?
        RET    Z                ;YES okay
        JP     A7915           ;NO, so Illegal Mode error
;*****
;Parse variable patch for extra commands.
;On exit, if the name matched, then CF=1, else CF=0.
;
A29202:
        CALL   A13826
        RET    NC
        LD     HL, (A16097)    ;start of variable command
name ASCII
        LD     A, NVARs       ;number of variables
A29211:
        PUSH  AF                ;save counter
        CALL  A29226           ;check variable command
table
        JR    Z, A29223        ;FOUND, so Illegal Variable
Name.
        POP   AF                ;NOT found
        DEC  A                ;one less to count
        JR   NZ, A29211        ;not done yet
        SCF
        RET
;*****
A29223:
        POP   AF                ;clear stack
        XOR  A                ;clear CF
        RET

```

```

;*****
;Check variable command table.
;On entry, HL points to an entry in the variable
command name ASCII table.
;On exit, HL is unchanged and ZF=1 if a match was
found, else HL points
;to the next entry and ZF=0. BC is preserved.
;
A29226:
    PUSH    BC
    LD      B,C
    LD      C,(HL)        ;get length of string
    INC     HL            ;point to 1st byte of ASCII
    CALL    A11851        ;compare string
    JR      Z,A29244      ;MATCHED, so exit ZF=1
    LD      A,C           ;NO MATCH, so get string
length
    ADD     A,L           ;point to next entry
    LD      L,A
    JR      NC,A29241
    INC     H
A29241:
    POP     BC
    OR      A
    RET
;*****
A29244:
    POP     BC
    RET
;*****
;Get numeric argument 0-255 in A and L.
;
A29246:
    JP      NZ,A7939      ;string argument, so Type
Mismatch error
    CALL    A2354        ;load FPA1 to HL as integer
    LD      A,H
    OR      A

```

```

        JR      NZ,A29264      ;>255 so Illegal Quantity
error
        LD      A,L           ;argument into A
        RET
;*****
;Get numeric argument 0-1 in A and L.
;
A29258:
        CALL   A29246        ;get numeric argument 0-255
in A and L
A29261:
        CP      2            ;more than 2?
        RET     C            ;NO, so okay
A29264:
        JP      A7936        ;YES, so Illegal Quantity
error
;*****
;Get numeric argument 0-3 in A and L.
;
A29267:
        CALL   A29246        ;get numeric argument 0-255
in A and L
        CP      4            ;more than 4?
        RET     C            ;NO, so okay
        JR      A29264        ;YES, so Illegal Quantity
error
;*****
;TIME(n) function where n={0,1,2,3} gives
{sixtieths,seconds,minutes,hours}.
;
A29275:
        CALL   A29267        ;get numeric argument 0-3 in
A and L
        LD      HL,A101      ;system clock data base
        LD      C,A
        LD      B,0
        OR      A
        SBC    HL,BC        ;back up to required byte

```

```

A29287:
    LD    L, (HL)        ;get it
A29288:
    JP    A10818        ;move into FPA1
;*****
;CLK= command.
;
A29291:
    CALL  A11214        ;skip token and get number
0-255 in A and L
    LD    (A253),A      ;save new CLK value
    RET
;*****
;Part of DATE(n) function. On entry, A=n argument
;where n={0,1,2} gives {day,month,year}.
;
A29298:
    LD    HL,EOS_DAY   ;EOS day
    LD    C,A
    LD    B,0
    OR    A
    SBC   HL,BC        ;offset to data
    LD    L, (HL)      ;get it
    CP    2            ;did we want the year?
    JR    C,A29288     ;NO, so exit with the data
    LD    H,0          ;YES, so add 1900 to make a
year
    LD    BC,1900
    ADD   HL,BC
    JP    A10820        ;exit and put into FPA1
;*****
;Unsigned integer division 9005.12.
;On entry, HL=dividend, DE=divisor
;On exit, HL=quotient, DE=remainder
;(internally, BC=quotient, HL=remainder)
;AF and BC are preserved, others destroyed.
;
A29321:

```

```

        PUSH    AF
        PUSH    BC
        LD      BC,0           ;quotient=0
        XOR     A              ;zero counter
A29327:
        PUSH    DE
        EX      DE,HL
        OR      A
        SBC     HL,DE
        EX      DE,HL
        POP     DE             ;is DE>=HL?
        JR      Z,A29365       ;equal YES
        JR      NC,A29365      ;greater YES
        OR      A
        SLA     E
        RL      D              ;DE=DE*2
        INC     A              ;count; doesn't affect CF!
        JR      C,A29353       ;DE*2 was >65535, signed
overflow
        CP      16             ;division by zero?
        JP      Z,A7918        ;YES, Divide By Zero error
        JR      A29327         ;NO, so keep going
;*****
A29353:
        RR      D              ;signed overflow
        RR      E              ;DE=DE/2 (CF is shifted back
in)
        OR      A
        SLA     C
        RL      B              ;BC=BC*2
        INC     A              ;count
        JR      A29401
;*****
A29365:
        INC     A              ;count
A29366:
        PUSH    DE
        EX      DE,HL

```

```

OR      A
SBC     HL,DE
EX      DE,HL
POP     DE           ;is DE<=HL?
JR      Z,A29379    ;equal YES
JR      C,A29379    ;less than YES
JR      A29411      ;NO greater than
;*****
A29379:
OR      A
SBC     HL,DE       ;HL=HL-DE
OR      A
SLA     C
RL      B           ;BC=BC*2
OR      A
PUSH    AF
LD      A,1
ADD     A,C
LD      C,A
JR      NC,A29396   ;BC=BC+1
INC     B           ;thus BC=1+(BC*2)
A29396:
POP     AF
A29397:
SRL     D
RR      E           ;DE=DE/2
A29401:
DEC     A           ;are we done?
JR      NZ,A29366   ;NO not yet
LD      E,L         ;YES, so do final register
setups and exit
LD      D,H
LD      L,C
LD      H,B
POP     BC
POP     AF
RET
;*****

```

```

A29411:
    OR     A             ;greater than handler
    SLA   C
    RL    B
    OR    A
    JR    A29397
;*****
;Common getter for MOD(x,y) and a bungled INT2(x,y).
;On exit, HL=divisor, BC=dividend.
;
A29419:
    JP    NZ,A7939      ;string argument, so Type
Mismatch error
    CALL  A9990         ;get memory address into HL
from FPA1
    PUSH  HL           ;save dividend
    CALL  A28082        ;skip token (,)
    CALL  A5939         ;get equation from crunch
code
    OR    A             ;is it a string?
    JP    NZ,A7939      ;YES, so Type Mismatch error
    CALL  A9990         ;get memory address into HL
from FPA1 (divisor)
    POP   BC           ;restore dividend
    RET
;*****
;INT2(x,y) function: returns INT(x/y).
;Historically in the development of SmartBASIC 1.x,
this was mistakenly
;called MOD(x,y), and the current MOD(x,y) was
called REM(x,y) (for
;REMainder). I was confused...and when I realized
my error, it was too
;late to delete the "MOD" code, since the true
MOD(x,y) used the end of
;it. There is thus no entry point provided to this
function, which for
;documentation purposes I am calling INT2, since

```

that is what it ends up
;calculating. I will clean this up when SmartBASIC
1.x is regenerated
;from this assembly listing.

```
;
A29441:
    CALL  A29419      ;common getter:  HL=divisor,
BC=dividend
    PUSH  DE          ;save current crunch code
position
    EX    DE,HL      ;DE=divisor
    LD    L,C
    LD    H,B        ;HL=dividend
    CALL  A29321      ;unsigned integer division:
quotient in HL
    POP   DE          ;restore crunch code
position
A29452:
    EX    DE,HL      ;quotient in DE, crunch code
position in HL
A29453:
    INC   SP          ;skip over return address,
pointing to
    INC   SP          ;original DE crunch code
position
                                ;NOTE:  the variable command
interpreter saves DE
                                ;to point to here:  MOD(x,y)
                                ;
    because it originally
                                ;expected only single
arguments, e.g. POS(x).
                                ;To work with multiple
arguments, we must replace
                                ;the original value with the
correct value.
    EX    (SP),HL    ;put *NEW* crunch code
position on stack
```



```

        EX      DE,HL          ;restore remainder (MOD) or
quotient (INT2) to HL
        OR      A
        CALL    A2407          ;move HL to FPA1
        JP      A5194          ;POP DE and finish routine
;*****
;MOD(x,y) function:  returns remainder of x/y.
;
A29464:
        CALL    A29419          ;common getter:  HL=divisor,
BC=dividend
        PUSH    DE              ;save crunch code position
        EX      DE,HL          ;DE=divisor
        LD      L,C
        LD      H,B            ;HL=dividend
        CALL    A29321          ;unsigned integer division:
remainder in DE
        POP     HL              ;crunch code position
        JR      A29453          ;finish it out
;*****
;Argumentless variable command patch for DATE$,
;TIME$, SPEED, ROT, SCALE, PROMPT, TERM, CLK, HDRV,
etc.
;
A29477:
        BIT     6,A              ;is it FN?
        JP      NZ,A5218        ;YES
        BIT     3,A              ;NO, is it DIM array?
        JP      NZ,A5176        ;YES
        LD      L,C
        LD      H,B            ;HL=vector
        PUSH    AF              ;save status
        PUSH    DE              ;save crunch code position
        CALL    A5938           ;JP (HL)
        POP     DE
        POP     AF
        BIT     5,A              ;was it a string command?
        JR      NZ,A29502       ;YES

```

```

        XOR    A            ;NO, so number in FPA1
        RET
;*****
A29502:
        LD     A,255        ;string in FPA1
        RET
;*****
;Write digit in L to string buffer.  Used by DATE$
and TIME$.
;On entry, BC points to the current position in the
string buffer.
;On exit, BC points at the next position.
;
A29505:
        LD     A,48         ;ASCII base for digit
        ADD    A,L          ;make into digit
        LD     (BC),A       ;write it
        INC    BC           ;point to next
        RET
;*****
;Write 10's and 1's digit to string buffer.  Used by
DATE$ and TIME$.
;On entry, A=number 0-99, BC points to current
position in string buffer.
;BC is incremented on exit.
;
A29511:
        LD     L,A
A29512:
        LD     H,0          ;dividend
        LD     DE,10        ;divisor
        CALL  A29321        ;unsigned integer division:
HL=quotient, DE=remainder
        CALL  A29505        ;write the 10's digit to
string buffer
        LD     L,E          ;HL=remainder
        CALL  A29505        ;write the 1's digit to
string buffer

```

```

        RET
;*****
;Delay for NMI.  Used by DATE$ and TIME$.
;Dodges the system clock roll-over to prevent update
in mid-routine.
;
A29528:
        PUSH   BC                ;save current string buffer
position
        CALL   A31952            ;wait
        POP    BC                ;restore
        RET
;*****
        NOP
        NOP
;*****
;Month string for DATE$.
;
A29536:
        DB     "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"

A29533      EQU    A29536-3

;*****
;TIME$ function.  Returns system time as "hh:mm:ss"
in 24-hour format.
;
A29572:
        PUSH   DE                ;save current crunch code
position
        LD     A,8
        LD     (A16246),A        ;length of string
        LD     BC,A16247        ;start of string
        CALL   A29528            ;delay to avoid roll-over
        LD     HL,A100          ;address of seconds
        LD     A,(HL)           ;get seconds
        LD     E,A              ;into E
        DEC   HL                ;point to minutes

```

```

LD      A, (HL)           ;get minutes
LD      D,A              ;into D
DEC     HL               ;point to hours
LD      A, (HL)          ;get hours
PUSH    DE               ;save minutes/seconds
CALL    A29511           ;write the hours
LD      A,58             ;colon
LD      (BC),A           ;write colon
INC     BC               ;point to next
POP     AF               ;restore minute in A
PUSH    AF               ;save seconds in F
CALL    A29511           ;write the minutes
LD      A,58
LD      (BC),A
INC     BC               ;write colon and point to
next
POP     HL               ;restore seconds in L
CALL    A29512           ;write seconds
JP      A10428           ;copy string to string space
and exit (POP DE)
;*****
A29618:
PUSH    DE
LD      A,15
LD      (A16246),A
LD      DE,A16247
CALL    A29528
LD      A, (A63)         ;weekday
LD      HL,A32214
CALL    A32238
LD      A, (EOS_DAY)
LD      C,E
LD      B,D
PUSH    DE
CALL    A29511
POP     DE
INC     DE
INC     DE

```

```

LD      A,45
LD      (DE),A
INC     DE
LD      A,(EOS_MONTH)
LD      HL,A29533
CALL    A32238
LD      A,(EOS_YEAR)
LD      C,A
LD      HL,1900
ADD     HL,BC
PUSH    DE
LD      DE,1000
CALL    A29321
POP     BC
CALL    A29505
EX      DE,HL
LD      DE,100
CALL    A29321
CALL    A29505
LD      A,E
CALL    A29511
JP      A10428

```

```

;*****

```

```

NOP
NOP
NOP
NOP
NOP
NOP

```

```

;*****

```

```

;Write byte in hex to string buffer.  Used by
HEX$(n).

```

```

;On entry, A=value, DE=string buffer position.

```

```

;On exit, DE is incremented.

```

```

;

```

```

A29706:

```

```

LD      C,A          ;save entry A
SRL     A

```

```

        SRL    A
        SRL    A
        SRL    A           ;high nibble to low nibble
        CALL   A29725      ;write the digit
        LD     A,C         ;restore entry A
        AND    15          ;get low nibble
        CALL   A29725      ;write the digit
        RET
;*****
;Write 1 hex digit.  Used by HEX$(n).
;On entry, A=digit, DE=string buffer position.
;On exit, DE is incremented.
;
A29725:
        LD     B,48        ;decimal ASCII base
        CP     10          ;is it 0-9?
        JR     C,A29735    ;YES
        LD     B,65        ;NO, so hex ASCII base
        SUB    10          ;bias for hex
A29735:
        ADD    A,B         ;make ASCII character
        LD     (DE),A      ;write it to string buffer
        INC    DE          ;point to next position
        RET
;*****
;HEX$(n) function where n={0 to 65535} or n={-32768
to 32767}.
;
A29739:
        JP     NZ,A7939    ;string argument, so Type
Mismatch error.
        LD     A,2         ;minimum length of string
        LD     (A16246),A  ;save it
        CALL   A9990       ;get memory address into HL
from FPA1
        CALL   A28082      ;skip token
        LD     A,(DE)      ;get current token
        CP     184         ;is it )?

```

```

        JP      NZ,A7948      ;NO, so Syntax error
        PUSH   DE            ;save crunch code position
        LD     DE,A16247     ;point to first slot in
string buffer
        LD     A,H
        OR     A             ;is argument >255?
        JR     Z,A29776     ;YES, so just 2 digits
        LD     A,4          ;NO, so 4 digits
        LD     (A16246),A   ;save new string length
        LD     A,H          ;get hbyte
        CALL  A29706        ;write it to string buffer
A29776:
        LD     A,L          ;get lobyte
        CALL  A29706        ;write it to string buffer
        JP     A10428       ;copy string to string space
and exit (POP DE)
;*****
;STRING$(ASCII code,count) function.
;
A29783:
        JP     NZ,A7939     ;string argument, so Type
Mismatch error
        CALL  A2354        ;get argument from FPA1 into
HL
        LD     A,H
        OR     A            ;is it >255?
        JP     NZ,A7936     ;YES, so Illegal Quantity
error
        LD     A,L          ;get character
        PUSH  AF            ;save it
        CALL  A28082        ;skip token (,)
        CALL  A5939         ;evaluate equation
        OR     A            ;is it a string?
        JP     NZ,A7939     ;YES, so Type Mismatch error
A29806:
        CALL  A2354        ;get argument from FPA1 into
HL
        LD     A,H

```

```

OR      A           ;is it >255?
JP      NZ,A7945    ;YES, so String Too Long
error
CALL    A28082      ;skip token
LD      A,(DE)      ;get current token
CP      184         ;is it )?
JP      NZ,A7948    ;NO, so Syntax error
LD      A,L         ;get length of string
LD      (A16246),A  ;save length of string
LD      B,A         ;make a counter
POP     AF          ;restore character
PUSH    DE          ;save crunch code position
LD      DE,A16247   ;point to first string slot
A29833:
LD      (DE),A      ;save character
INC     DE          ;point to next
DJNZ    A29833      ;keep going 'til done
JP      A10428      ;copy string to string space
and exit (POP DE)
;*****
;SPC$(count) function.
;
A29840:
JP      NZ,A7939    ;string argument, so Type
Mismatch error
LD      A,32        ;character is a space
PUSH    AF          ;save it
JR      A29806      ;use STRING$ routine to exit
;*****
;MEM(n) function, where n={0,1,2,3} gives
;{default LOMEM,current LOMEM,default HIMEM,current
HIMEM}
;
A29848:
CALL    A29267      ;get numeric argument 0-3 in
A and L
LD      BC,A29861   ;MEM(n) function vector
table

```



```

A29854:
    ADD    HL,HL           ;make argument into offset
A29855:
    ADD    HL,BC           ;offset into table
    LD     A,(HL)
    INC    HL
    LD     H,(HL)
    LD     L,A             ;get vector
    JP     (HL)           ;go there
;*****
;MEM(n) function vector table.
;
;*****
A29861:
    DW     A29869         ;MEM(0)
    DW     A29874         ;MEM(1)
    DW     A29879         ;MEM(2)
    DW     A29885         ;MEM(3)
;*****
;MEM(0) function:  default LOMEM.
;
A29869:
    LD     HL,LOMEM
    JR     A29888
;*****
;MEM(1) function:  current LOMEM.
;
A29874:
    LD     HL,(A16095)
    JR     A29888
;*****
;MEM(2) function:  default HIMEM.
;
A29879:
    LD     HL,(A16089)
    DEC    HL
    JR     A29888
;*****

```

```

;MEM(3) function:  current HIMEM.
;
A29885:
        LD      HL,(A16109)
A29888:
        JP      A10820
;*****
;ADDR(n) function, where n={0,1,2,3,4,5,6,7} gives
;{default shape table addr,current shape table addr,
;default USR addr,current USR addr,
;
A29891:
        CALL   A31343      ;get numeric argument 0-3 in
A and L
        LD      BC,A31373  ;ADDR(n) vector table
A29897:
        JR      A29854      ;get the vector and go there
;*****
;MERGE command.
;Patches LOAD so that programs are MERGEed in memory.
;It bypasses the usual jump to NEW.
;NOMERGE enters at A29902 to save the vector.
;
A29899:
        LD      HL,A16035  ;central loop of SmartBASIC
1.x
A29902:
        LD      (A24010),HL ;change vector
        RET
;*****
        NOP
;*****
;ADDR(0) function:  default shape table address.
;
A29907:
        LD      HL,A26574
        JR      A29888
;*****

```

```

;ADDR(1) function:  current shape table address.
;
A29912:
        LD      HL,(A16766)
        JR      A29888
;*****
;ADDR(2) function:  default USR address.
;
A29917:
        LD      HL,A6345
        JR      A29888
;*****
;ADDR(3) function:  current USR address.
;
A29922:
        LD      HL,(A16130)
        JR      A29888
;*****
;VER(n) function, where n={0,1} gives
{SmartBASIC,EOS} versions.
;
A29927:
        CALL   A29258      ;get numeric argument 0-1 in
A and L
        OR     A           ;is it EOS?
        JR     NZ,A29940   ;YES
        LD     A,(A1499)   ;NO, so get SmartBASIC 1.x
version
A29936:
        LD     L,A         ;into L (H already 0)
        JP     A10820      ;copy to FPA1 and exit
;*****
A29940:
        LD     A,(REV_NUM) ;get EOS version
        JR     A29936      ;save and exit
;*****
;WIDTH(n) function, where n={0,1} gives {TEXT,PR}
widths.

```

```

;
A29945:
    CALL  A29258      ;get numeric argument 0-1 in
A and L
    OR     A          ;is it printer?
    JR     NZ,A29961  ;YES
    LD     A,(A17988) ;NO, so get TEXT width
    CP     32         ;was it 32?
    JR     NZ,A29936  ;NO, so no adjustment needed
    DEC   A          ;YES, so make into TEXT31
    JR     A29936     ;save and exit
;*****
A29961:
    LD     A,(A16176) ;get printer width
    JR     A29936     ;save and exit
;*****
;SPEED function.
;
A29966:
    LD     A,(A16129) ;get current SPEED
A29969:
    LD     L,A        ;into L
    JP     A10818     ;H=0, save and exit
;*****
;ROT function.
;
A29973:
    LD     A,(A1498)  ;get current ROT
    JR     A29969     ;save and exit
;*****
;SCALE function.
;
A29978:
    LD     A,(A16765) ;get current SCALE
    JR     A29969     ;save and exit
;*****
;PROMPT function.
;

```

```

A29983:
    LD      A, (A1146)    ;get current prompt
character
    JR      A29969        ;save and exit
;*****
;ROT patch to save argument.
;
A29988:
    LD      A,C           ;ROT value in A
    LD      (A1498),A     ;save it
    JP      A26344        ;back to ROT main
;*****
A29995:
    CALL    A29172
    JP      A6159
;*****
;COLOR= or COLOR(n)= command, where
n={0,1,2,3,4,5,6,7} gives
;{COLOR,HCOLOR,brdr,NORMALtxt,NORMALbkgd,INVERSEtxt,
INVERSEbkgd,graphics}
;
A30001:
    CALL    A30039        ;look for which COLOR
command
    JP      Z,A11099      ;COLOR=
    CALL    A11214        ;get numeric argument 0-255
in A and L
    CP      8             ;is it <8?
    JP      NC,A7936      ;NO, so Illegal Quantity
error
    CALL    A28082        ;YES, so skip token (=)
    LD      BC,A30023     ;COLOR(n)= command vector
table
A30021:
    JR      A29897        ;get vector and go there
;*****
;COLOR(n)= command vector table.
;

```

```

A30023:
    DW      A11099          ;COLOR
    DW      A11106          ;HCOLOR
    DW      A11113          ;border
    DW      A11126          ;NORMAL text
    DW      A11149          ;NORMAL background
    DW      A11159          ;INVERSE text
    DW      A11176          ;INVERSE background
    DW      A11186          ;graphics screen
;*****
;Look for which COLOR command.
;On exit, ZF=1 if COLOR=, ZF=0 if COLOR(n)=
;
A30039:
    INC     DE              ;point to next token in
crunch code
    LD      A, (DE)        ;get it
    DEC     DE              ;back up
    CP      170            ;is it =?
    RET
;*****
;Parameter getter for MEM(n)= and ADDR(n)=.
;On exit, DE points to the value to be assigned.
;
A30045:
    CALL    A11214         ;get numeric argument 0-255
in A and L
    CP      4              ;is it less than 4?
A30050:
    JP      NC, A7936       ;NO, so Illegal Quantity
error
    CALL    A28082         ;YES, so skip token (=)
    RET
;*****
;MEM(n)= command.
;
A30057:
    CALL    A30045         ;parameter getter for

```

```

MEM(n)= and ADDR(n)=
    LD    BC,A30065    ;base of MEM(n)= command
vector table
    JR    A30021      ;get vector and go there
;*****
;MEM(n)= command vector table.
;
A30065:
    DW    A7933      ;Undefined Function
    DW    A10874     ;current LOMEM
    DW    A7933      ;Undefined Function
    DW    A11014     ;current HIMEM
;*****
;ADDR(n)= command.
;
A30073:
    CALL  A11214     ;get numeric argument 0-255
in A and L
    CP    8          ;is it <8?
    CALL  A30050     ;check parameter and skip
token
    LD    BC,31403   ;ADDR(n)= command vector
table
    JR    A30021     ;get vector and go there
;*****
A30086:
    DW    A33743     ;ADDR(7)
    NOP
;*****
;Parameter getter for register= commands.
;
A30089:
    CALL  A28082     ;skip token (=)
    CALL  A9987      ;get memory address from
FPA1 into HL
    RET
;*****
;AF= command.

```

```

;
A30096:
    CALL  A30089      ;get parameter
    LD    (A30987),HL ;save it
    RET
;*****
;BC= command.
;
A30103:
    CALL  A30089      ;get parameter
    LD    (A30989),HL ;save it
    RET
;*****
;DE= command.
;
A30110:
    CALL  A30089      ;get parameter
    LD    (A8334),HL  ;save it
    RET
;*****
;HL= command.
;
A30117:
    CALL  A30089      ;get parameter
    LD    (A8336),HL  ;save it
    RET
;*****
;IX= command.
;
A30124:
    CALL  A30089      ;get parameter
    LD    (A8338),HL  ;save it
    RET
;*****
;IY= command.
;
A30131:
    CALL  A30089      ;get parameter

```



```

        LD      (A8340),HL  ;save it
        RET
;*****
;AF function.
;
A30138:
        LD      HL,(A30987) ;get value
A30141:
        JP      A10820      ;save and exit
;*****
;BC function.
;
A30144:
        LD      HL,(A30989) ;get value
        JR      A30141      ;save and exit
;*****
;DE function.
;
A30149:
        LD      HL,(A8334)  ;get value
        JR      A30141      ;save and exit
;*****
;HL function.
;
A30154:
        LD      HL,(A8336)  ;get value
        JR      A30141      ;save and exit
;*****
;IX function.
;
A30159:
        LD      HL,(A8338)  ;get value
        JR      A30141      ;save and exit
;*****
;IY function.
;
A30164:
        LD      HL,(A8340)  ;get value

```

```

        JR      A30141          ;save and exit
;*****
;CALL nnnn or CALL EOS(nn) command.
;
A30169:
        INC    DE              ;point to next token in
crunch code
        LD     A,(DE)          ;get it
        DEC    DE              ;back up
        CP     190             ;is it EOS?
        JP     NZ,A10042       ;NO, so do regular CALL nnnn
        CALL   A7299           ;check stack to prevent
blowups
        CALL   A11214          ;get numeric argument 0-255
in A and L
        CP     101             ;is it <101?
        JP     NC,A7936        ;NO, so Illegal Quantity
error
        LD     BC,_EOS_START   ;YES, so get base of
EOS jump table
        ADD    HL,HL           ;*2
        OR     A
        ADD    A,L             ;*3 for final offset (each
JP nnnn is 3 bytes)
        LD     L,A
        JR     NC,A30198
        INC    H
A30198:
        ADD    HL,BC           ;offset into jump table
        LD     (A30237),HL     ;save address
                                ;I hate self-modifying code,
but I couldn't
                                ;think of a way to do it
without changing
                                ;registers. I suppose I
could have double-
                                ;pushed the stack, put my
EOS vector on the

```

address below ;top, and the desired return
I'm wiser now than ;it, then done a RET...but
;I was in 1991...

```
PUSH DE
EXX
PUSH DE
PUSH BC
PUSH HL
EXX
PUSH IY
PUSH IX ;save all registers
LD HL, (A30987) ;get AF from storage...
PUSH HL
POP AF ;...into AF register
LD BC, (A30989) ;get BC
LD DE, (A8334) ;get DE
LD HL, (A8336) ;get HL
LD IX, (A8338) ;get IX
LD IY, (A8340) ;get IY
A30236:
CALL A0 ;EOS function call

A30237 EQU A30236+1

LD (A8340), IY ;save IY
LD (A8338), IX ;save IX
LD (A8336), HL ;save HL
LD (A8334), DE ;save DE
LD (A30989), BC ;save BC
PUSH AF ;get AF register...
POP HL
LD (A30987), HL ;and put into AF storage
JP A10061 ;finish regular CALL
(restores registers)
;*****
;CALL parse vector table
```

```

;
A30266:
    DB      1
    DW      A30269      ;parse CALL
;*****
;Parse CALL.
;
A30269:
    LD      A,190      ;token for "EOS"
    CALL    A14581     ;is it found?
    JP      NC,A14875  ;NO, so parse numeric
equation (CALL nnnn)
    CALL    A14575     ;YES, so parse "("
    JP      NC,A1122   ;not found, so print "'('
Expected"
    CALL    A14875     ;parse numeric equation
    JP      A15532     ;parse ")"
;*****
;Common getter for BIT(arg,bit), SET(arg,bit), and
RES(arg,bit).
;On exit, H=0, L=argument, A=bit mask.
;
A30289:
    CALL    A29246     ;get numeric argument 0-255
in A and L
    PUSH   HL         ;save it
    CALL    A28082     ;skip token (,
    CALL    A5939      ;evaluate equation
    OR     A          ;is it a number?
    CALL    A29246     ;get numeric argument 0-255
in A and L
    CP     8          ;is bit <8?
    JP      NC,A7936   ;NO, so Illegal Quantity
Error
    POP    HL         ;restore argument
    INC    A          ;one more
    LD     B,A        ;make a counter
    XOR    A          ;A=0

```

```

        SCF                ;set carry (gives a bit)
A30313:
        RLA                ;shift over one bit at a
time to make a mask
        DJNZ  A30313      ;count times
        RET
;*****
;BIT(argument,bit) function.
;
A30317:
        CALL  A30289      ;get parameter
        AND   L           ;mask all but the desired
bit
        OR    A           ;is it set?
        JR    Z,A30326    ;YES, so return zero
        LD   A,1         ;NO, so return one
A30326:
        LD   L,A         ;value in L, H=0
        JP   A29452      ;save and exit
;*****
;SET(argument,bit) function.
;
A30330:
        CALL  A30289      ;get parameteer
        OR    L           ;set the desired bit
        JR    A30326      ;exit with value
;*****
;RES(argument,bit) function.
;
A30336:
        CALL  A30289      ;get parameter
        CPL                    ;1's complement the mask
        AND   L           ;clear the desired bit
        JR    A30326      ;exit with value
;*****
;Common getter for AND(arg1,arg2), OR(arg1,arg2),
and XOR(arg1,arg2).
;On exit, A=arg1, L=arg2, H=0.

```

```

;
A30343:
    CALL    A29246        ;get numeric arg1 0-255 in A
and L
    PUSH    AF            ;save arg1
    CALL    A28082        ;skip token (,)
    CALL    A5939         ;evaluate equation
    OR      A              ;is it a number?
    CALL    A29246        ;get numeric arg2 0-255 in A
and L
    POP     AF            ;restore arg1
    RET
;*****
;AND(arg1,arg2) function.
;
A30359:
    CALL    A30343        ;get arguments
    AND     L              ;do the AND
    JR      A30326        ;save and exit
;*****
;OR(arg1,arg2) function.
;
A30365:
    CALL    A30343        ;get arguments
    OR      L              ;do the OR
    JR      A30326        ;save and exit
;*****
;XOR(arg1,arg2) function.
;
A30371:
    CALL    A30343        ;get arguments
    XOR     L              ;do the XOR
    JR      A30326        ;save and exit
;*****
;CPL(argument) function.
;
A30377:
    CALL    A29246        ;get numeric argument 0-255

```

```

in A and L
    CPL                ;do the 1's complement
    LD      L,A        ;into L (H=0)
    JP      A10820     ;save and exit
;*****
;ZF function.
;
A30385:
    LD      A, (A30987) ;get flag byte from stored
AF
    BIT     6,A        ;is ZF set?
A30390:
    LD      HL,1       ;start assuming it is set
    JR      NZ,A30397  ;YES it was set
    LD      L,0        ;NO it was clear, so make
zero
A30397:
    JP      A10820     ;save and exit
;*****
;CF function.
;
A30400:
    LD      A, (A30987) ;get flag byte from stored
AF
    BIT     0,A        ;is CF set?
    JR      A30390     ;finish using ZF routine
;*****
;Argumentless RND function. Returns the next random
number in the series.
;
A30407:
    LD      BC, (A16190) ;get seed1
    LD      HL, (A16192) ;get seed2
    JP      A4723       ;do RND
;*****
;COLOR(n) function, where n={0,1,2,3,4,5,6,7} gives
;{COLOR,HCOLOR,brdr,NORMALtxt,NORMALbkgd,INVERSEtxt,
INVERSEbkgd,graphics}.

```

```

;
A30417:
    CALL  A29246      ;get argument 0-255 in A and
L
    CP    8           ;is it <8?
    JP    NC,A7936    ;NO, so Illegal Quantity
error
    LD    BC,A30431   ;YES, so get base of
COLOR(n) function vector table
    JP    A29854      ;get vector and go there
;*****
;COLOR(n) function vector table.
;
A30431:
    DW    A30447      ;COLOR
    DW    A30452      ;HCOLOR
    DW    A30457      ;border
    DW    A30462      ;NORMAL text
    DW    A30477      ;NORMAL background
    DW    A30484      ;INVERSE text
    DW    A30489      ;INVERSE background
    DW    A30494      ;graphics
;*****
;COLOR(0) function:  COLOR.
;
A30447:
    LD    A,(A16776)
    JR    A30473      ;save and exit
;*****
;COLOR(1) function:  HCOLOR.
;
A30452:
    LD    A,(A16777)
    JR    A30473      ;save amd exit
;*****
;COLOR(2) function:  border.
;
A30457:

```



```

        LD      A,(A17059)
        JR      A30473      ;save and exit
;*****
;COLOR(3) function:  NORMAL text.
;
A30462:
        LD      A,(A17115)
A30465:
        SRL     A
        SRL     A
        SRL     A
        SRL     A      ;get hi nibble into lo
nibble
A30473:
        LD      L,A      ;into L (H=0)
        JP      A10818    ;save and exit
;*****
;COLOR(4) function:  NORMAL background.
;
A30477:
        LD      A,(A17115)
A30480:
        AND     15      ;get low nibble
        JR      A30473    ;save and exit
;*****
;COLOR(5) function:  INVERSE text.
;
A30484:
        LD      A,(A17126)
        JR      A30465    ;get hi nibble, save and
exit
;*****
;COLOR(6) function:  INVERSE background.
;
A30489:
        LD      A,(A17126)
        JR      A30480    ;get lo nibble, save and
exit

```

```

;*****
;COLOR(7) function:  graphic screen.
;
A30494:
        LD      A,(A18633)
        JR      A30465      ;save and exit
;*****
;Remove spaces patch for LIST.
;
A30499:
        CP      55
        JP      C,A31188
        CP      60
        RET     C
        CP      62
        JP      C,A31188
        CP      64
        RET     C
        CP      68
        JP      C,A31188
        CP      78
        JP      A32177      ;check for more
commands/spaces
;*****
;Parse RANDOMIZE or RANDOMIZE seed1,seed2.
;
A30525:
        CALL   A13659      ;check type of character
        RET    NC          ;no more characters, so
plain RANDOMIZE
        CALL   A14875      ;parse numeric equation
        CALL   A15939      ;parse ",", "
        JP     A14875      ;parse numeric equation
;*****
;OUT port,value.
;
A30538:
        CALL   A1500       ;get port 0-255 in A and L

```

```

        PUSH    HL                ;save it
        CALL    A28082            ;skip token
        CALL    A1500             ;get value 9-255 in A and L
        POP     BC                ;restore port in C
        OUT     (C),L            ;send the data
        RET
;*****
;Parse IN #n or IN variable,port
;
A30552:
        LD      A,182             ;token for "#"
        CALL    A14581            ;is it found?
        JP      C,A14875         ;YES, so parse numeric
equation and exit (IN #n)
        CALL    A15364           ;NO, so parse numeric
variable
        CALL    A15939           ;parse ",",
        JP      A14875           ;parse numeric equation
;*****
;IN #n or IN variable,port command.
;
A30569:
        INC     DE                ;point to next byte of
crunch code
        LD      A,(DE)           ;get it
        DEC     DE                ;back up
        CP      182              ;was it "#"?
        JP      Z,A12084         ;YES, so do IN #n
        CALL    A6555            ;NO, so branch on variable
type
;*****
;IN variable,port vector table.
;Only FP and % are valid; others give Type Mismatch
error.
;
        DW      A30590           ;FP
        DW      A30607           ;%
        DW      A7942            ;$

```

```

        DW      A7942          ;FN
        DW      A7942          ;math command
;*****
;FP variable for IN variable,port.
;
A30590:
        PUSH   BC              ;save variable address
        CALL   A28082          ;skip token (#)
        CALL   A1500           ;get port 0-255 in A and L
        LD     C,L             ;port in C
        IN     A,(C)           ;read port
        LD     L,A             ;data in L (H=0)
        CALL   A2407           ;copy HL to FPA1
        JP     A6268           ;write FPA1 to variable
;*****
;% variable for IN variable,port.
;
A30607:
        PUSH   BC              ;save variable address
        CALL   A28082          ;skip token (#)
        CALL   A1500           ;get port 0-255 in A and L
        LD     C,L             ;port in C
        IN     A,(C)           ;read port
        POP    HL              ;restore variable address
        LD     (HL),A          ;save data lobyte
        INC    HL              ;point to hibyte
        LD     (HL),0          ;zero out hibyte
        RET
;*****
A30623:
        DB     14
        DB     "Device Timeout"
;*****
;Parallel printer send (PR#2) with timeout check.
;On entry, A=character to send.
;On exit, AF, BC, and HL are preserved.
;
A30638:

```

```

        PUSH    AF                ;save character
        PUSH    HL
        PUSH    BC
        LD      B,9                ;big loop counter
A30643:
        LD      HL,65535          ;little loop counter
A30646:
        IN      A,(64)            ;read the parallel port
        AND     1                  ;is it on-line?
        JR      NZ,A30668         ;YES
A30652:
        DEC     HL                ;NO, so decrement little
loop counter
        LD      A,H
        OR      L                  ;are we down to zero?
        JR      NZ,A30646         ;NO, so keep trying
        DEC     B                  ;YES, so decrement big loop
counter
        JR      NZ,A30643         ;not done yet, so restart
little loop
A30660:
        LD      HL,A16213         ;timed out, so start to
restore to PR#0
        LD      A,(HL)
        INC     HL
        JP      A32142            ;exit to Device Timeout
error
;*****
A30668:
        IN      A,(64)            ;read the parallel port
        AND     2                  ;is it ready to receive
another character?
        JR      NZ,A30652         ;NO, so keep trying
        POP     BC
        POP     HL
        POP     AF                ;restore character
        OUT     (64),A            ;send it
        RET

```

```

;*****
;SERIAL parameters for Eve/Orphanware and ADAMlink
modem.
;Stored after SERIAL command is executed.
;
A30680:
    DB    68    ;Eve/Orphanware port
    DW    0     ;baudrate
    DB    0     ;stats
;
    DB    76    ;Eve/Orphanware port
    DW    0     ;baudrate
    DB    0     ;stats
;
    DB    84    ;Eve/Orphanware port
    DW    0     ;baudrate
    DB    0     ;stats
;
    DB    92    ;Eve/Orphanware port
    DW    0     ;baudrate
    DB    0     ;stats
;
    DB    94    ;ADAMlink modem port
    DW    0     ;baudrate
    DB    0     ;stats
;*****
;Eve/Orphanware serial port baudrate initialization
data.
;
A30700:
    DW    300
    DB    53
;
    DW    1200
    DB    55
;
    DW    2400
    DB    58

```

```

;
    DW      4800
    DB      60
;
    DW      9600
    DB      62
;
    DW      19200
    DB      63
;*****
;HDRV function.
;
A30718:
    LD      A, (A30724)    ;current HDRV
A30721:
    JP      A29969        ;save and exit
;*****
A30724:
    DB      0              ;current HDRV
;*****
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
;*****
;Rest of SERIAL port,baudrate,stats command.
;Main entry is at A11726.
;
A30731:
    CP      (HL)
    JR      Z,A30740
    ADD     HL,DE
    DJNZ   A30731
A30737:
    JP      A7936
;*****

```

A30740:

```
POP    DE
PUSH   HL
LD     (A28074),A
NOP
CALL   A28082
CALL   A9987
PUSH   DE
EX     DE,HL
CALL   A33899
LD     B,6
```

A30759:

```
LD     A,(HL)
CP     E
JR     Z,A30770
INC    HL
INC    HL
INC    HL
```

A30766:

```
DJNZ   A30759
JR     A30737
```

;*****

A30770:

```
INC    HL
LD     A,(HL)
CP     D
JR     NZ,A30766
LD     A,C
CP     94
JR     NZ,A30790
LD     HL,300
OR     A
SBC   HL,DE
JR     NZ,A30737
JR     A30795
```

;*****

A30790:

```
INC    HL
```



```

        LD     A, (HL)
        LD     (A28077), A
A30795:
        LD     (A28075), DE
        POP   DE
        CALL  A28082
        CALL  A1500
        LD     A, L
        CP     2
        JR     NC, A30737
        LD     (A28079), A
        CALL  A11754
        NOP
        CP     94
        JR     NZ, A30825
        LD     HL, A28072
A30825:
        LD     A, (A28079)
        ADD   A, L
        LD     L, A
        JR     NC, A30833
        INC   H
A30833:
        LD     A, (HL)
        LD     (A28078), A
        POP   HL
        INC   HL
        LD     BC, (A28075)
        LD     (HL), C
        INC   HL
        LD     (HL), B
        INC   HL
        LD     A, (A28079)
        LD     (HL), A
        CALL  A11767
        CP     94
        JR     NZ, A30872
        LD     A, 128

```

```
OUT    (95),A
LD     A,64
OUT    (95),A
LD     A,(A28078)
OUT    (95),A
RET
```

```
;*****
A30872:
```

```
ADD    A,3
LD     C,A
IN     A,(C)
XOR    A
OUT    (C),A
DEC    C
LD     A,(A28078)
OUT    (C),A
LD     A,(A28077)
OUT    (C),A
INC    C
LD     A,39
OUT    (C),A
RET
```

```
;*****
```

```
;PR #3 command.
```

```
;On entry, A=character to send.
```

```
;On exit, A=character, and CR/LFs are forced if end  
of line is reached.
```

```
;If printer width is 255, then end-of-line checking  
is ignored.
```

```
;
```

```
A30897:
```

```
LD     HL,A28080    ;PR #3 baseport number
LD     C,(HL)      ;get baseport
INC    C           ;make into status port for
```

```
Eve/Orphanware
```

```
CALL   A30945      ;serial send with timeout
```

```
check
```

```
PUSH  AF           ;save character
```

```

        LD      A, (A16176)    ;get printer width
        CP      255           ;is it "infinite"?
        JR      Z, A30941     ;YES, so don't bother to
force CR/LFs at "end of line"
        POP     AF           ;NO, so restore character
        PUSH    AF           ;and save it again
        CP      13           ;was it a CR?
        JR      Z, A30930     ;YES, so add a LF
        LD      A, (A16177)   ;NO, so get current printer
head position
        DEC     A            ;one less
        JR      NZ, A30938    ;not at the end yet
        LD      A, 13        ;at the end of the line, so
print a CR
        CALL    A30945        ;serial send with timeout
check
A30930:
        LD      A, 10        ;print a LF
        CALL    A30945        ;call serial send with
timeout check
        LD      A, (A16176)   ;get printer width
A30938:
        LD      (A16177), A   ;reset current position to
max width
A30941:
        POP     AF           ;restore character
        JP      A12043        ;fall through to echo
character on the screen
;*****
;Serial send (PR #3) with timeout check.
;On entry, A=character to send, C=status port
(Eve/Orphanware)
;or data port (MIB2). On exit, AF, DE and C are
preserved.
;
A30945:
        JP      A35397        ;check Eve/Orphanware versus
MIB2 serial ports.

```

```

;*****
;Rest of routine, for Eve/Orphanware serial ports.
;On entry, D=big loop counter, C=status port,
A=character to send.
;On exit, AF, DE and C are preserved.
;
        NOP
A30949:
        LD     HL,65535      ;little loop counter
A30952:
        IN     A,(C)        ;read status port
        LD     B,A          ;save it in B
        CP     255          ;does port exist?
        JR     Z,A30969     ;NO, so keep trying
        AND    56           ;YES, but are there
framing/parity/overrun errors?
        JP     NZ,A32164    ;YES
        LD     A,B          ;NO, so restore data read in
from port
        AND    1            ;is it ready to receive a
character?
        JR     NZ,A30980   ;YES, so send it
A30969:
        DEC    HL          ;NO, not yet, so decrement
little loop counter
        LD     A,H
        OR     L            ;are we down to zero?
        JR     NZ,A30952   ;NO, so keep trying
        DEC    D           ;YES, so decrement big loop
        JR     NZ,A30949   ;not done yet, so reset
little loop and try again
        JP     A30660      ;all done, so reset to PR#0
and Device Timeout error
;*****
A30980:
        POP    DE
        POP    AF          ;restore character
        DEC    C           ;back up to data port

```

```

        OUT      (C),A      ;send character
        INC      C          ;back to status port
        RET
;*****
A30987:
        DW      0          ;AF storage
A30989:
        DW      0          ;BC storage
;*****
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
;*****
;PRN #n vector table.
;
A31000:
        DW      A32157     ;PRN #1 ADAM printer
        DW      A30638     ;PRN #2 parallel printer
        DW      A10184     ;PRN #3 serial printer
        DW      A32157     ;PRN #4 ADAM printer
        DW      A32157     ;PRN #5 ADAM printer
        DW      A32157     ;PRN #6 ADAM printer
        DW      A32157     ;PRN #7 ADAM printer
;*****
A31014:
        DW      A32157     ;current PRN #n vector
A31016:
        DB      255        ;^P indicator:
255=inactive, 0=active
;*****
;PRN #n command.  n={1,2,3,4,5,6,7}.
;

```

```

A31017:
    CALL    A28082
    CALL    A1500
    XOR     A
    OR      L
    JR      Z,A31045
    LD      A,7
    CP      L
    JR      C,A31045
    ADD     HL,HL
    LD      BC,A31000-2 ;PRN #n vector table-2
    ADD     HL,BC
    LD      A,(HL)
    INC     HL
    LD      H,(HL)
    LD      L,A
    LD      (A31014),HL ;save current vector
    RET
;*****
A31045:
    JP      A7936          ;Illegal Quantity error
;*****
A31048:
    INC     HL
A31049:
    LD      A,(HL)
    CP      3
    JR      Z,A31060
    CALL    A31076
    DEC     C
    JR      NZ,A31048
A31060:
    LD      A,13
    CALL    A31076
    LD      A,10
    CALL    A31076
    LD      A,255
    LD      (A31016),A

```

```

        RET
;*****
;Do print to PRN #n device.
;
A31076:
        PUSH   AF
        PUSH   BC
        PUSH   DE
        PUSH   HL
        LD     HL,(A31014)
        JP     A12001
;*****
;SER(n) function.
;
A31086:
        CALL   A29246
        LD     B,5
        LD     HL,A30680
A31094:
        CP     (HL)
        JR     Z,A31117
        INC    HL
        INC    HL
        INC    HL
        INC    HL
        DJNZ   A31094
        CALL   A35358
        LD     HL,A28080
        ADD    A,L
        LD     L,A
        JR     NC,A31114
        INC    H
A31114:
        JP     A29287
;*****
A31117:
        PUSH   HL
        CALL   A28082

```

```

        CALL    A5939
        OR      A
        CALL    A29258
        POP     HL
        INC     HL
        OR      A
        JR      Z,A31141
        INC     HL
        INC     HL
        LD      L,(HL)
        LD      H,0
A31138:
        JP      A29452
;*****
A31141:
        LD      A,(HL)
        INC     HL
        LD      H,(HL)
        LD      L,A
        JR      A31138
;*****
;SER(n)=port command.
;
A31147:
        CALL    A11214
        CP      2
        JR      NC,A31172
        PUSH   AF
        CALL    A11214
        LD      B,4
        JP      A35337
;*****
A31163:
        CP      (HL)
        JR      Z,A31175
        INC     HL
        INC     HL
        INC     HL

```



```

        INC     HL
        DJNZ   A31163
A31172:
        JP     A7936
;*****
A31175:
        LD     B,A
        POP   AF
        LD     HL,A28080
        ADD   A,L
        LD     L,A
        JR    NC,A31185
        INC   H
A31185:
        LD     A,B
        LD     (HL),A
        RET
;*****
A31188:
        LD     A,32
        CALL  A11994
        RET
;*****
        NOP
        NOP
        NOP
;*****
A31197:
        PUSH  AF
        DEC   DE
        LD    A,(DE)
        INC   DE
        CP    15
        JP    A1096
;*****
;PRWIDTH=value command.
;
A31206:

```

```

        CALL    A11214
        OR     A
        JP     Z,A7936
        LD     H,A
        LD     (A16176),HL
        RET
;*****
A31218:
        CALL    A30638
        PUSH   AF
        LD     A,(A16176)
        CP     255
        JR     Z,A31257
        POP    AF
        PUSH   AF
        CP     13
        JR     Z,A31246
        LD     A,(A16177)
        DEC    A
        JR     NZ,A31254
        LD     A,13
        CALL    A30638
A31246:
        LD     A,10
        CALL    A30638
        LD     A,(A16176)
A31254:
        LD     (A16177),A
A31257:
        POP    AF
        JP     A12043
;*****
;DSIZE(n) table.
;
A31261:
        DB     1                ;DSIZE(0)=1 directory block

A31260      EQU     A31261-1

```

```
DW      256      ;D1  tape 1
DW      256      ;D2  tape 2/hard drive
DW      256      ;D3  tape 3
DW      256      ;D4  tape 4
DW      160      ;D5  disk 1
DW      160      ;D6  disk 2
DW      63       ;D7  RAM disk
```

```
;*****
```

```
A31276:
```

```
LD      DE,0
PUSH   HL
LD      HL,A23273
```

```
A31283:
```

```
CP      (HL)
JR      Z,A31290
INC     HL
INC     DE
JR      A31283
```

```
;*****
```

```
A31290:
```

```
LD      HL,A31261+1
ADD     HL,DE
ADD     HL,DE
LD      E,(HL)
INC     HL
LD      D,(HL)
LD      HL,A31261
LD      C,(HL)
POP     HL
RET
```

```
;*****
```

```
A31304:
```

```
PUSH   AF
JP      A171
```

```
;*****
```

```
NOP
NOP
```

```

        NOP
        NOP
        NOP
        NOP
;*****
A31314:
        CALL    A12009
        PUSH   BC
        PUSH   DE
        PUSH   HL
        PUSH   IX
        PUSH   IY
        LD     HL,(A254)
        CALL   A5938
        POP    IY
        POP    IX
        POP    HL
        POP    DE
        POP    BC
        LD     A,(A17008)
        JP     A106
;*****
A31343:
        CALL   A29246
        CP     8
        JP     NC,A7936
        RET
;*****
;ADDR(4) function:  return default user NMI routine
vector.
;
A31352:
        LD     HL,A6345      ;a RET
A31355:
        JP     A29888
;*****
;ADDR(5) function:  return current user NMI routine
vector.

```

```

;
A31358:
    LD    HL,(A254)    ;get the address
    JR    A31355
;*****
;ADDR(6) function:  return default TERM emulation
pointer table address.
;
A31363:
    LD    HL,A33743
    JR    A31355
;*****
;ADDR(7) function:  return current TERM emulation
pointer table address.
;
A31368:
    LD    HL,(A30086)
    JR    A31355
;*****
;ADDR(n) function vector table.
;
A31373:
    DW    A29907      ;0  return default shape
table address
    DW    A29912      ;1  return current shape
table address
    DW    A29917      ;2  return default USR
routine vector
    DW    A29922      ;3  return current USR
routine vector
    DW    A31352      ;4  return default user NMI
routine vector
    DW    A31358      ;5  return current user NMI
routine vector
    DW    A31363      ;6  return default TERM
emulation pointer table
    DW    A31368      ;7  return current TERM
emulation pointer table

```

```

;*****
;ADDR(5)= command.  Set current user NMI routine
vector.
;
A31389:
    CALL  A9987
    LD    (A254),HL    ;set the address
    RET
;*****
;ADDR(7)= command.  Set current TERM emulator
pointer table.
;
A31396:
    CALL  A9987
    LD    (A30086),HL
    RET
;*****
;ADDR(n)= command vector table.
;Even-numbered functions {0,2,4,6} are Undefined
Function errors.
;
A31403:
    DW    A7933        ;0
    DW    A10171       ;1  set current shape table
address
    DW    A7933        ;2
    DW    A10164       ;3  set current USR routine
vector
    DW    A7933        ;4
    DW    A31389       ;5  set current NMI routine
vector
    DW    A7933        ;6
    DW    A31396       ;7  set current TERM
emulation pointer table
;*****
;TEXT31 main entry.
;WINDOW patch.
;

```

```

A31419:
    LD      BC,7937      ;left margin=1, right
margin=31
    LD      HL,5888      ;top margin=0, bottom
margin=23
    LD      (A69),BC
    LD      (A71),HL
    JP      A28023      ;rest of TEXT31
;*****
;TEXT40 main entry.
;WINDOW patch.
;
A31435:
    LD      BC,10241     ;left margin=1, right
margin=40
    LD      HL,5888      ;top margin=0, bottom
margin=23
    LD      (A69),BC
    LD      (A71),HL
    JP      A28048
;*****
;GR/HGR patch for WINDOW.
;
A31451:
    LD      BC,7937      ;left margin=1, right
margin=31
    LD      HL,5908      ;top margin=20 bottom
margin=23
    LD      (A69),BC
    LD      (A71),HL
    RET
;*****
;GR command.
;
A31465:
    CALL   A31451
    JP     A11070
;*****

```

;HGR command.

;

A31471:

CALL A31451

JP A11075

;*****

A31477:

CP L

JP Z,A17644

LD A,(A72)

CP L

JP Z,A17644

JP A17634

;*****

A31491:

CP L

JP Z,A17581

LD A,(A71)

CP L

JP Z,A17581

JP A17667

;*****

A31505:

CP H

JP Z,A17687

LD A,(A70)

CP H

JP Z,A17687

JP A17683

;*****

A31519:

JP Z,A17581

LD A,(A72)

CP L

JP Z,A17581

JP A17694

;*****

A31532:


```

        CP      H
        JP      Z,A17715
        LD      A,(A69)
        CP      H
        JP      Z,A17715
        JP      A17711
;*****
A31546:
        JP      Z,A17581
        LD      A,(A71)
        CP      L
        JP      Z,A17581
        JP      A17722
;*****
;CLK function.
;
A31559:
        LD      A,(A253)      ;current CLK
A31562:
        JP      A29969
;*****
;TERM function.
;
A31565:
        LD      A,(A64)      ;current TERM
        JR      A31562
;*****
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
;*****
;WINDOW parse vector table.
;
A31576:
        DB      7

```

```

        DW      A14875      ;
        DW      A15939      ;
        DW      A14875      ;
        DW      A15963      ;
        DW      A14875      ;
        DW      A15939      ;
        DW      A14875      ;
;*****
;WINDOW command main entry.
;
A31591:
        LD      A, (A17008)  ;current screen mode
        OR      A           ;is it TEXTxx?
        JP      NZ,A7915    ;NO, so Illegal Mode error
        CALL   A33860
        LD      (A66),A     ;
        CALL   A28082
        CALL   A31696
        LD      (A65),A     ;
        CALL   A28082
        CALL   A31708
        LD      (A68),A     ;
        CALL   A28082
        CALL   A31696
        LD      (A67),A     ;
        LD      A, (A65)     ;
        CP      L
        JP      NC,A7936
        LD      A, (A68)     ;
        LD      L,A
        LD      A, (A66)     ;
        CP      L
        JP      NC,A7936
        DEC    L
        LD      H,L
        DEC    A
        LD      L,A
        LD      (A16958),HL

```

```

LD      (A16995),A
LD      A,H
SUB     L
INC     A
LD      (A16993),A
LD      A,(A67)      ;
LD      H,A
LD      A,(A65)      ;
LD      L,A
LD      (A16956),HL
LD      (A16996),A
LD      A,H
SUB     L
INC     A
LD      (A16994),A
RET
;*****
      NOP
      NOP
;*****
A31688:
      CALL  A1500
      OR    L
      JP    Z,A7936
      RET
;*****
A31696:
      CALL  A31688
      LD    A,(A70)
A31702:
      CP    L
      JP    C,A7936
      LD    A,L
      RET
;*****
A31708:
      CALL  A31688
      LD    A,(A72)

```

```

        INC     A
        JR      A31702
;*****
A31717:
        LD      HL,(A17001)
        LD      A,(A16958)
        CP      L
        JR      Z,A31761
        JR      C,A31755
A31728:
        LD      A,(A70)
        INC     H
        CP      H
        JP      NC,A17492
        LD      A,(A69)
        LD      H,A
        INC     L
        LD      A,(A72)
        CP      L
        JP      NC,A17492
        LD      A,(A71)
        LD      L,A
        JP      A17492
;*****
A31755:
        LD      A,(A16959)
        CP      L
        JR      C,A31728
A31761:
        LD      A,(A16957)
        CP      H
A31765:
        JR      C,A31728
        LD      A,(A16956)
        CP      H
        JP      Z,A17465
        JP      C,A17465
        INC     H

```

```

        CP      H
        JR      Z,A31784
        DEC     H
        JR      A31728
;*****
A31784:
        LD      A,(A16957)
        LD      H,A
        JR      A31728
;*****
;DSIZE(n) function.
;
A31790:
        CALL    A29246
        CP      8
        JP      NC,A7936
        LD      BC,A31261
        OR      A
        JR      NZ,A31808
        LD      A,(BC)
        JP      A29969
;*****
A31808:
        ADD     HL,HL
        DEC     BC
        ADD     HL,BC
        LD      A,(HL)
        INC     HL
        LD      H,(HL)
        LD      L,A
        JP      A10820
;*****
;DSIZE(n) = command.
;
A31818:
        CALL    A11214
        CP      8
        JP      NC,A7936

```

```
PUSH AF
CALL A28082
CALL A30089
POP AF
PUSH HL
LD HL,A31261
OR A
JR NZ,A31854
POP BC
XOR A
OR B
JP NZ,A7936
LD A,C
OR A
JP Z,A7936
LD (HL),A
RET
```

```
;*****
```

```
A31854:
```

```
DEC HL
LD C,A
LD B,0
ADD HL,BC
ADD HL,BC
POP BC
LD A,C
OR B
JP Z,A7936
LD (HL),C
INC HL
LD (HL),B
RET
```

```
;*****
```

```
;PEEK(address) or PEEK(address,latch) function.
```

```
;
```

```
A31870:
```

```
JP NZ,A7939
CALL A9990
```

```
INC    DE
LD     A, (DE)
DEC    DE
CP     184
JP     Z, A10097
CALL   A28082
PUSH   HL
CALL   A1500
LD     A, L
CP     21
JP     NC, A7936
POP    HL
LD     C, A
DEC    C
PUSH   AF
LD     A, H
CP     128
JR     NC, A31910
POP    AF
JP     A32768
```

```
;*****
```

```
A31910:
```

```
POP    AF
OR     A
JR     Z, A31920
CP     17
JR     C, A31926
JR     Z, A31944
```

```
A31920:
```

```
LD     L, (HL)
```

```
A31921:
```

```
LD     H, 0
JP     A29452
```

```
;*****
```

```
A31926:
```

```
CALL   A31951
LD     A, C
OUT    (66), A
```

```

        LD      A,9
A31934:
        OUT    (127),A
        LD      L,(HL)
        LD      A,1
        OUT    (127),A
A31941:
        EI
        JR      A31921
;*****
A31944:
        CALL   A31951
        LD      A,13
        JR      A31934
;*****
A31951:
        DI
A31952:
        LD      A,(A17008)
        OR      A
        RET    NZ
        LD      A,(A101)
        JP      A32528
;*****
;POKE address,value [,latch] command.
;
A31963:
        CALL   A9987
        PUSH   HL
        CALL   A28082
        CALL   A1500
        NOP
        LD      B,L
        EXX
        LD      A,C
        EXX
        OR      A
        JR      NZ,A31984

```



```

A31981:
    POP    HL
    LD     (HL),B
    RET
;*****
A31984:
    CALL  A28082
    PUSH  BC
    CALL  A1500
    POP   BC
    LD    A,L
    OR    A
    JR    Z,A31981
    LD    C,A
    DEC   C
    POP   HL
    PUSH  DE
    LD    D,B
    PUSH  AF
    LD    A,H
    CP    128
    JR    NC,A32011
    POP   AF
    JP    A32826
;*****
A32011:
    POP   AF
    CP    17
    JR    C,A32021
    JR    Z,A32039
    POP   DE
    LD    (HL),B
    RET
;*****
A32021:
    CALL  A31951
    LD    A,C
    OUT   (66),A

```

```

        LD      A,9
A32029:
        OUT    (127),A
        LD     (HL),D
        LD     A,1
        OUT    (127),A
        EI
        POP    DE
        RET
;*****
A32039:
        CALL   A31951
        LD     A,13
        JR     A32029
;*****
A32046:
        INC    B
A32047:
        POP    AF
        PUSH   AF
        CALL   A12110
        LD     HL,A11980
        POP    AF
        CALL   C,A12110
        LD     A,B
        OR     A
        JP     Z,A16035
        LD     DE,A16247
        CALL   A11777
        CALL   A11884
        JP     NC,A16035
        CALL   A12451
        CALL   A11777
        PUSH   DE
        LD     H,D
        LD     L,E
        XOR    A
        LD     B,255

```

```
CPIR
DEC    HL
SBC    HL,DE
LD     A,L
CP     248
JP     NC,A16035
LD     C,L
LD     B,0
INC    BC
LD     DE,A16507
POP    HL
LDIR
LD     HL,A16503
ADD    A,5
LD     (A16205),A
SUB    2
LD     (HL),A
INC    HL
LD     (HL),67
INC    HL
LD     (HL),144
INC    HL
SUB    3
LD     (HL),A
LD     HL,(A16207)
LD     DE,A16503
CALL  A12618
JP     A16035
```

```
;*****
```

```
A32142:
```

```
LD     H,(HL)
LD     L,A
LD     (A16201),HL
LD     A,255
LD     (A31016),A
LD     A,34
JP     A7950
```

```
;*****
```

;PRN #1,4,5,6,7.

;

A32157:

CALL _PR_CH

RET Z

JP A30660

;*****

A32164:

LD HL,A16213

LD A,(HL)

INC HL

LD H,(HL)

LD L,A

LD (A16201),HL

JP A24321

;*****

A32177:

RET C

CP 102

JP C,A31188

CP 106

RET C

JP A31188

;*****

A32189:

LD (EOS_MONTH),HL

EXX

LD A,C

EXX

OR A

RET Z

CALL A28082

CALL A1500

LD A,L

OR A

JP Z,A7936

CP 8

JP NC,A7936

```

        LD      (A63),A
        RET
;*****
A32217:
        DB      "SUNMONTUEWEDTHUFRISAT" ;day-of-week
string

A32214      EQU      A32217-3      ;backup for offset

;*****
A32238:
        PUSH   HL
        PUSH   AF
        ADD    A,A
        LD     L,A
        POP    AF
        ADD    A,L
        LD     C,A
        LD     B,0
        POP    HL
        ADD    HL,BC
        LD     BC,3
        LDIR
        LD     A,45
        LD     (DE),A
        INC    DE
        RET
;*****
;DATE(n) function.
;
A32259:
        CALL   A29246
        CP     4
        JP     NC,A7936
        CP     3
        JP     C,A29298
        LD     A,(A63)
        LD     L,A

```

```

        LD      H,0
        JP      A10820
;*****
;Weekday patch for DATE$.
;
A32281:
        LD      A,(A63)
        INC     A
        CP      8
        JR      C,A32291
        LD      A,1
A32291:
        LD      (A63),A
        XOR     A
        LD      (HL),A
        LD      HL,A86-1      ;month table base
        RET
;*****
A32300:
        CALL    A14875
        LD      A,191
        CALL    A14581
        RET     NC
        JP      A14875
;*****
;TERM= command.
;
A32312:
        CALL    A11214
        LD      (A64),A
        RET
;*****
;Digitizer SmartClock data.
;
A32319:
        DB      197,58,163,92,197,58,163,92      ;wakeup
data
A32327:

```

```

        DS      10                      ;clock
buffer
;*****
A32337:
        DS      6                      ;temporary stack (for
bank-switch)
A32343:
        DW      0                      ;temporary stack pointer
;*****
A32345:
        PUSH   DE
        CALL   A31951
        LD     (A32343),SP
        LD     SP,A32343
        CALL   A32426
        CALL   A32466
        LD     A,1
        OUT    (127),A
        LD     SP,(A32343)
        LD     HL,A32327
        LD     DE,A32890
        LD     BC,10
        LDIR
        CALL   A33113
        EI
        POP    DE
        RET
;*****
A32387:
        PUSH   DE
        CALL   A33247
        LD     HL,A32890
        LD     DE,A32327
        LD     BC,10
        LDIR
        LD     (A32343),SP
        LD     SP,A32343
        CALL   A32426

```

```

CALL    A32499
LD      A,1
OUT     (127),A
LD      SP,(A32343)
EI
POP     DE
RET
;*****
A32426:
LD      A,13
OUT     (127),A
LD      B,64
A32432:
LD      A,(49156)    ;not a program address
                        ;it's the address bus wakeup
for the clock
DJNZ    A32432
LD      HL,49152    ;not a program address
                        ;it's the address bus wakeup
for the clock
LD      DE,A32319
LD      C,8
A32445:
LD      B,8
LD      A,(DE)
A32448:
PUSH    AF
PUSH    HL
AND     1
ADD     A,L
LD      L,A
LD      A,(HL)
POP     HL
POP     AF
SRL     A
DJNZ    A32448
INC     DE
DEC     C

```



```

        JR      NZ,A32445
        RET
;*****
A32466:
        LD      HL,49156      ;not a program address
                                ;it's the address bus wakeup
for the clock
        LD      DE,A32327
        LD      C,8
A32474:
        LD      B,8
        LD      A,0
A32478:
        PUSH   HL
        PUSH   AF
        LD      A,(HL)
        AND    1
        LD      L,A
        POP    AF
        RRC    A
        OR     L
        POP    HL
        DJNZ   A32478
        RRC    A
        LD     (DE),A
        INC    DE
        DEC    C
        JR     NZ,A32474
        RET
;*****
A32499:
        LD      HL,49152      ;not a program address
                                ;it's the address bus wakeup
for the clock
        LD      DE,A32327
        LD      C,8
A32507:
        LD      B,8

```

```

        LD      A, (DE)
A32510:
        PUSH   AF
        PUSH   HL
        AND    1
        ADD    A, L
        LD     L, A
        LD     A, (HL)
        POP    HL
        POP    AF
        SRL   A
        DJNZ  A32510
        INC   DE
        DEC   C
        JR    NZ, A32507
        RET
;*****
A32528:
        LD     B, A
A32529:
        LD     A, (A101)
        CP    B
        JR    Z, A32529
        RET
;*****
A32536:
        DB    "BOOT.SYSA", 3
A32546:
        DB    31, 12, "Coleco ADAM SmartBASIC 1.x R20"
A32578:
        DB    30, "(c) 1991 by Richard F. Drushel"
;*****
;Cold boot entry.
;
A32609:
        LD     SP, SB10_STACK
        EXX
        LD     B, 0

```

```

        EXX
        LD     A,255
        LD     A,(A17007)
        LD     BC,257           ;B=1, C=1:  code for January
1st
        LD     D,91           ;code for 1991
        CALL  _SET_DATE
        LD     A,(CURRENT_DEV)
        LD     (A16821),A
        LD     HL,A16885
        LD     DE,A32536
        CALL  _QUERY_FILE
        JR     NZ,A32659
        LD     HL,A32536
        LD     DE,A16797
        LD     BC,10
        LDIR
        JR     A32721
;*****
A32659:
        CALL  A33631
        LD     HL,A32546
        CALL  A12110
        CALL  A12128
        CALL  A12128
        LD     HL,A32578
        CALL  A12110
        LD     A,28
        LD     DE,278
        CALL  A11994
        JR     A32727
;*****
;Wipe out the boot routine. (hahaha!)
;
A32690:
        LD     HL,201           ;make A0: JP A32609 into A0:
RET /NOP /NOP
        LD     (A0),HL

```

```

        XOR    A
        LD     (A2),A
        LD     HL,18627      ;NOT an address
                                ;the missing instructions
for NMI start
        LD     (A102),HL
        CALL  READ_REGISTER
        LD     HL,A32536
        LD     BC,180
        XOR    A
A32716:
        LD     (HL),A
        INC   HL
        DJNZ  A32716
        RET
;*****
A32721:
        CALL  A32690
        JP    A24020      ;do RUN BOOT.SYS
;*****
A32727:
        CALL  A32690
        JP    A16035      ;go to Central Loop
;*****
;        DS    35
        DS    32768-$      ;fill out rest of space to
32768 boundary
                                ;must guarantee that no-slot
clock code is
                                ;in upper 32K RAM
;*****
A32768:
        OR     A
        JP    Z,A31920
        CP    17
        JR    C,A32816
        JP    Z,A31920
        CP    19

```

```

        JR      Z,A32802
        JR      C,A32809
        CALL   A31951
        XOR    A
A32789:
        OUT    (63),A
        XOR    A
A32792:
        OUT    (127),A
        LD     L,(HL)
        LD     A,1
        OUT    (127),A
        JP     A31941
;*****
A32802:
        CALL   A31951
        LD     A,2
        JR     A32789
;*****
A32809:
        CALL   A31951
        LD     A,3
        JR     A32792
;*****
A32816:
        CALL   A31951
        LD     A,C
        OUT    (66),A
        LD     A,2
        JR     A32792
;*****
A32826:
        CP     17
        JR     C,A32869
        JR     Z,A32879
        CP     19
        JR     C,A32862
        JR     Z,A32855

```

```

        CALL    A31951
        XOR     A
A32842:
        OUT     (63),A
        XOR     A
A32845:
        OUT     (127),A
        LD      (HL),D
        LD      A,1
        OUT     (127),A
A32852:
        EI
        POP     DE
        RET
;*****
A32855:
        CALL    A31951
        LD      A,2
        JR      A32842
;*****
A32862:
        CALL    A31951
        LD      A,3
        JR      A32845
;*****
A32869:
        CALL    A31951
        LD      A,C
        OUT     (66),A
        LD      A,2
        JR      A32845
;*****
A32879:
        LD      (HL),D
        JR      A32852
;*****
;Motherboard no-slot clock (CLK=1) data.
;

```

```
A32882:
      DB      197,58,163,92,197,58,163,92      ;wakeup
data
```

```
A32890:
      DS      10                                ;clock
buffer
```

```
A32891      EQU      A32890+1
A32893      EQU      A32890+3
A32894      EQU      A32890+4
A32896      EQU      A32890+6
```

```
;*****
;CLREAD command entry.
;
```

```
A32900:
      LD      A, (A253)      ;get CLK
      OR      A              ;no hardware?
      JP      Z,A32974      ;YES
      CP      1              ;no-slot?
      JP      Z,A33048      ;YES
      CP      2              ;digitizer?
      JP      Z,A32345      ;YES
      CP      3              ;Eve/Orphanware?
      JP      Z,A33427      ;YES
      JP      A24321        ;NO, so I/O error
```

```
;*****
```

```
A32925:
      CALL   A31951
```

```
A32928:
      LD      A,2
      OUT    (63),A
      NOP
      NOP
      OUT    (63),A
      XOR    A
      OUT    (127),A
      LD      B,64
```

A32941:
LD A, (A0+4)
DJNZ A32941
LD HL, 0
LD DE, A32882
LD C, 8

A32954:
LD B, 8
LD A, (DE)

A32957:
PUSH AF
PUSH HL
AND 1
ADD A, L
LD L, A
LD A, (HL)
POP HL
POP AF
SRL A
DJNZ A32957
INC DE
DEC C
JR NZ, A32954

A32974:
RET

;*****

A32975:
LD HL, A0
LD DE, A32890
LD C, 8

A32983:
LD B, 8
LD A, (DE)

A32986:
PUSH AF
PUSH HL
AND 1
ADD A, L


```
LD L,A
LD A,(HL)
POP HL
POP AF
SRL A
DJNZ A32986
INC DE
DEC C
JR NZ,A32983
RET
```

```
;*****
```

```
A33004:
```

```
LD HL,A0+4
LD DE,A32890
LD C,8
```

```
A33012:
```

```
LD B,8
LD A,0
```

```
A33016:
```

```
PUSH HL
PUSH AF
LD A,(HL)
AND 1
LD L,A
POP AF
RRC A
OR L
POP HL
DJNZ A33016
RRC A
LD (DE),A
INC DE
DEC C
JR NZ,A33012
RET
```

```
;*****
```

```
A33037:
```

```
XOR A
```

```
    OUT    (63),A
    NOP
    NOP
    OUT    (63),A
    INC    A
    OUT    (127),A
    RET
```

```
;*****
```

```
A33048:
```

```
    PUSH  DE
    CALL  A32925
    CALL  A33004
    CALL  A33037
    CALL  A33113
    EI
    POP   DE
    RET
```

```
;*****
```

```
A33064:
```

```
    PUSH  DE
    CALL  A33247
    CALL  A32928
    CALL  A32975
    CALL  A33037
    EI
    POP   DE
    RET
```

```
;*****
```

```
A33080:
```

```
    LD    A, (HL)
```

```
A33081:
```

```
    PUSH  BC
    PUSH  AF
    AND   15
    CP    10
    RET   NC
    LD    C,A
    POP   AF
```

```
SRL    A
SRL    A
SRL    A
SRL    A
CP     10
RET    NC
ADD    A,A
PUSH   AF
ADD    A,A
ADD    A,A
LD     B,A
POP    AF
ADD    A,B
ADD    A,C
LD     (HL),A
POP    BC
SCF
RET
```

```
;*****
```

```
A33113:
```

```
LD     HL,A32891
LD     B,2
```

```
A33118:
```

```
CALL   A33080
JR     NC,A33136
INC    HL
DJNZ   A33118
LD     A,(HL)
BIT    7,A
JR     NZ,A33139
CALL   A33080
JR     C,A33167
```

```
A33136:
```

```
JP     A24321
```

```
;*****
```

```
A33139:
```

```
BIT    5,A
JR     NZ,A33160
```

```

        AND    31
        CALL  A33081
        JR    NC,A33136
        ADD   A,12
        CP    24
        JR    C,A33157
        XOR   A
A33157:
        LD    (HL),A
        JR    A33167
;*****
A33160:
        AND    31
        CALL  A33081
        JR    NC,A33136
A33167:
        INC   HL
        LD    B,4
A33170:
        CALL  A33080
        JR    NC,A33136
        INC   HL
        DJNZ  A33170
        DEC   HL
        CP    83
        JR    NC,A33186
        ADD   A,100
        LD    (HL),A
A33186:
        LD    A,(A32894)
        LD    (A63),A
        LD    HL,A33209
        PUSH  HL
        LD    HL,A32893
        LD    A,(HL)
        PUSH  AF
        DEC   HL
        LD    A,(HL)

```

```
PUSH AF
DEC HL
LD A, (HL)
JP A28125
```

```
;*****
```

```
A33209:
```

```
LD HL, A32896
LD A, (HL)
PUSH AF
DEC HL
LD A, (HL)
PUSH AF
INC HL
INC HL
LD A, (HL)
JP A28216
```

```
;*****
```

```
A33223:
```

```
PUSH HL
LD L, A
LD H, 0
LD DE, 10
CALL A29321
LD A, L
SLA A
SLA A
SLA A
SLA A
OR E
POP HL
LD (HL), A
INC HL
RET
```

```
;*****
```

```
A33247:
```

```
CALL A31951
LD HL, A32890
XOR A
```

```

LD      (HL),A
INC     HL
LD      A,(A100)
CALL    A33223
LD      A,(A99)
CALL    A33223
LD      A,(A98)
CALL    A33223
LD      A,(A63)
LD      (HL),A
INC     HL
LD      A,(EOS_DAY)
CALL    A33223
LD      A,(EOS_MONTH)
CALL    A33223
LD      A,(EOS_YEAR)
A33294:
CP      100
JR      C,A33302
SUB     100
JR      A33294
;*****
A33302:
CALL    A33223
RET
;*****
;Eve/Orphanware clock buffer.
;
A33306:
DS      13
;*****
A33319:
CALL    A31951
A33322:
LD      A,129
OUT     (75),A
XOR     A
OUT     (72),A

```

```

        LD     A,143
        OUT   (73),A
        LD     A,15
        OUT   (73),A
        RET
;*****
A33338:
        LD     A,145
        OUT   (75),A
        LD     B,0
        LD     HL,A33306
        LD     A,16
        OUT   (74),A
        PUSH  AF
        LD     A,30
A33354:
        DEC   A
        NOP
        JR    NZ,A33354
        POP  AF
A33359:
        LD     A,48
        OUT   (74),A
        LD     A,B
        OUT   (73),A
        NOP
        INC   B
        IN    A,(72)
        AND   15
        LD    (HL),A
        INC   HL
        LD    A,B
        CP    13
A33377:
        JR    NZ,A33359
        LD    A,128
        OUT   (74),A
        RET

```

;*****

A33384:

LD HL,A33306
LD B,0
LD A,128
OUT (75),A

A33393:

LD A,16
OUT (74),A
PUSH AF
LD A,30

A33400:

DEC A
NOP
JR NZ,A33400
POP AF
LD A,B
OUT (73),A
LD A,(HL)
OUT (72),A
LD A,80
OUT (74),A
INC HL
INC B
LD A,128
OUT (74),A
LD A,B
CP 13
JR NZ,A33393
RET

;*****

A33427:

PUSH DE
CALL A33319
CALL A33338
CALL A33470
EI
POP DE


```

    RET
;*****
A33440:
    PUSH    DE
    CALL    A33545
    CALL    A33322
    CALL    A33384
    EI
    POP     DE
    RET
;*****
A33453:
    LD      A, (BC)
    LD      E, A
    INC     BC
    LD      A, (BC)
    SLA    A
    SLA    A
    SLA    A
    SLA    A
    OR      E
    LD      (HL), A
    INC     BC
    INC     HL
    RET
;*****
A33470:
    LD      BC, A33306
    LD      HL, A32891
    CALL    A33453
    CALL    A33453
    CALL    A33453
    DEC     HL
    BIT     7, A
    JP      NZ, A33525
    SET     7, A
    BIT     6, A
    JR      NZ, A33499

```

```

A33497:
    SET    5,A
A33499:
    RES    6,A
    LD     (HL),A
    INC    HL
    LD     A,(BC)
    INC    A
    LD     (HL),A
    INC    HL
    INC    BC
    CALL   A33453
    CALL   A33453
    DEC    HL
    AND    127
    LD     (HL),A
    INC    HL
    CALL   A33453
    JP     A33113
;*****
A33525:
    JP     A33652
;*****
A33528:
    PUSH   HL
    LD     L,A
    LD     H,0
    LD     DE,10
    CALL   A29321
    LD     A,L
    POP    HL
    LD     (HL),E
    INC    HL
    LD     (HL),A
    INC    HL
    RET
;*****
A33545:

```

```

CALL    A31951
LD      HL,A33306
LD      A,(A100)
CALL    A33528
LD      A,(A99)
CALL    A33528
LD      A,(A98)
CALL    A33528
DEC     HL
SET     3,(HL)
INC     HL
LD      A,(A63)
DEC     A
LD      (HL),A
INC     HL
LD      A,(EOS_DAY)
CALL    A33528
LD      A,(EOS_MONTH)
CALL    A33528
LD      A,(EOS_YEAR)
A33594:
CP      100
JR      C,A33602
SUB     100
JR      A33594
;*****
A33602:
CALL    A33528
A33605:
RET
;*****
;CLWRITE command.
;
A33606:
LD      A,(A253)    ;get CLK
OR      A           ;no hardware?
JP      Z,A33605    ;YES
CP      1           ;no-slot?

```

```

        JP      Z,A33064      ;YES
        CP      2            ;digitizer?
        JP      Z,A32387      ;YES
        CP      3            ;Eve/Orphanware?
        JP      Z,A33440      ;YES
        JP      A24321        ;NO, so I/O error
;*****
;TEXT command.
;
A33631:
        LD      A,(A17988)    ;current screen width
        CP      32            ;TEXT31?
        JP      Z,A31419      ;YES
        CP      40            ;NO, TEXT40?
        JP      Z,A31435      ;YES
        CP      80            ;NO, TEXT80?
        JP      Z,A34455      ;YES
        JP      A13086        ;NO, so FATAL SYSTEM ERROR
;*****
A33652:
        RES     7,A
        BIT     6,A
        JP      Z,A33499
        JP      A33497
;*****
;80-column screen buffer.
;
A33662:
        DS      80

A33661      EQU    A33662-1
A33740      EQU    A33662+78
A33741      EQU    A33740+1

        DB      3            ;end of buffer
;*****
;TERM terminal emulation vector table.
;

```

A33743:

```
DW      A33759      ;TERM=0 TTY
DW      A33770      ;TERM=1 Heath-19
DW      A33810      ;TERM=2 External (defaults
to TTY)
DW      A33810      ;TERM=3 External (defaults
to TTY)
DW      A33810      ;TERM=4 External (defaults
to TTY)
DW      A33810      ;TERM=5 External (defaults
to TTY)
DW      A33810      ;TERM=6 External (defaults
to TTY)
DW      A33810      ;TERM=7 External (defaults
to TTY)
;*****
;TERM=0 TTY emulation.
;
```

A33759:

```
DB      TTY0-A33759 ;offset to 12 (clear screen
and home cursor)
DB      TTY0-A33759 ;offset to 22 (erase to
right of screen)
DB      TTY0-A33759 ;offset to 24 (erase to
bottom of screen)
DB      TTY0-A33759 ;offset to 28 (absolute
cursor move x,y)
DB      TTY0-A33759 ;offset to 128 (home cursor)
DB      TTY0-A33759 ;offset to 151 (delete
character to right of cursor)
DB      TTY0-A33759 ;offset to INVERSE ON
DB      TTY0-A33759 ;offset to INVERSE OFF
DB      TTY0-A33759 ;offset to FLASH ON
DB      TTY0-A33759 ;offset to FLASH OFF
```

TTY0:

```
DB      0           ;null data (this is dumb
TTY)
;*****
```

;Heath-19 terminal emulation.

;

A33770:

```
    DB      H190-A33770 ;offset to 12 (clear screen
and home cursor)
    DB      H191-A33770 ;offset to 22 (erase to
right of screen)
    DB      H192-A33770 ;offset to 24 (erase to
bottom of screen)
    DB      H193-A33770 ;offset to 28 (absolute
cursor move x,y)
    DB      H194-A33770 ;offset to 128 (home cursor)
    DB      H195-A33770 ;offset to 151 (delete
character to right of cursor)
    DB      H196-A33770 ;offset to INVERSE ON
    DB      H197-A33770 ;offset to INVERSE OFF
    DB      H198-A33770 ;offset to FLASH ON
    DB      H199-A33770 ;offset to FLASH OFF
```

;data here is in (length,data bytes) format.

H190:

```
    DB      2,27,69      ;data for 12 (clear screen
and home cursor)
```

H191:

```
    DB      2,27,75      ;data for 22 (erase to right
of screen)
```

H192:

```
    DB      2,27,74      ;data for 24 (erase to
bottom of screen)
```

H193:

```
    DB      2,27,89      ;data for 28 (absolute
cursor move x,y)
```

H194:

```
    DB      2,27,72      ;data for 128 (home cursor)
```

H195:

```
    DB      2,27,78      ;data for 151 (delete
character to right of cursor)
```

```

H196:
    DB      2,27,112      ;data for INVERSE ON
H197:
    DB      2,27,113      ;data for INVERSE OFF
H198:
    DB      2,27,112      ;data for FLASH ON
H199:
    DB      2,27,113      ;data for FLASH OFF
;*****
;TERM=2,3,4,5,6,7 External emulation.
;Defaults to TTY.
;
A33810:
    DB      TTY1-A33810 ;offset to 12 (clear screen
and home cursor)
    DB      TTY1-A33810 ;offset to 22 (erase to
right of screen)
    DB      TTY1-A33810 ;offset to 24 (erase to
bottom of screen)
    DB      TTY1-A33810 ;offset to 28 (absolute
cursor move x,y)
    DB      TTY1-A33810 ;offset to 128 (home cursor)
    DB      TTY1-A33810 ;offset to 151 (delete
character to right of cursor)
    DB      TTY1-A33810 ;offset to INVERSE ON
    DB      TTY1-A33810 ;offset to INVERSE OFF
    DB      TTY1-A33810 ;offset to FLASH ON
    DB      TTY1-A33810 ;offset to FLASH OFF
TTY1:
    DB      0              ;null data
;*****
    DS      39             ;rest of External space
;*****
;WINDOW patch.
;Disables WINDOW in TEXT80.
;
A33860:
    LD      A,(A17988)     ;get screen width

```

```

        CP      80          ;is it TEXT80?
        JP      Z,A7915    ;YES, so WINDOW not allowed
        JP      A31708     ;NO, so go to WINDOW
;*****
;SERIAL port status for MIB2.
;
A33871:
        DB      1          ;port
        DW      0          ;baudrate
        DB      0          ;stats
;
        DB      2          ;port
        DW      0          ;baudrate
        DB      0          ;stats
;*****
;MIB2 baudrate data for initialization.
;Format:  baudrate (word), data.
;
;*****
A33879:
        DW      300
        DB      68
;
        DW      1200
        DB      102
;
        DW      2400
        DB      136
;
        DW      4800
        DB      153
;
        DW      9600
        DB      187
;
        DW      19200
        DB      204
;*****

```


;MIB2 parity/data/stop data for initialization.

;

A33897:

DB 2 ;E71

DB 19 ;N81

;*****

A33899:

LD A,(A28074)

LD C,A

LD HL,A30700

CP 68

RET NC

LD HL,A33879

RET

;*****

NOP

NOP

;*****

;Control character table for TEXT80.

;

A33915:

DB 12 ;clear screen and home cursor

DB 22 ;erase to right of screen

DB 24 ;erase to bottom of screen

DB 28 ;absolute cursor move to x,y

DB 128 ;home cursor

A33920:

DB 151 ;delete character to right of
cursor

DB 7 ;bell

DB 8 ;backspace

DB 9 ;horizontal tab

DB 10 ;linefeed

DB 13 ;carriage return

DB 16 ;^P for screen dump to printer

DB 148 ;insert

DB 160 ;up arrow

DB 161 ;right arrow

```

        DB      162      ;down arrow
        DB      163      ;left arrow
;*****
;Control character vector table for TEXT80.
;
A33932:
        DW      A34670      ;163
        DW      A34395      ;162
        DW      A34644      ;161
        DW      A34599      ;160
        DW      A34858      ;148
        DW      A34976      ;166
        DW      A35154      ;13
        DW      A34395      ;10
        DW      A34185      ;9
        DW      A34670      ;8
        DW      A35033      ;7
        DW      A34805      ;151
        DW      A34368      ;128
        DW      A34534      ;28
        DW      A34757      ;24
        DW      A34748      ;22
        DW      A34348      ;12
;*****
A33966:
        PUSH    AF
        JP      A34444
;*****
        NOP
        NOP
A33972:
        LD      A, (A17988)
        CP      80
        JP      Z, A34132
A33980:
        POP     AF
        PUSH    AF
        EX      AF, AF'

```

```

        LD     A, (A17005)
        JP     A17247
;*****
A33989:
        PUSH  AF
        PUSH  BC
        PUSH  HL
        LD     A, (A17008)
        CP     1
        JP     NC, A17399
        LD     A, (A17988)
        CP     80
        JP     NZ, A17399
        POP   HL
        POP   BC
        POP   AF
        RET
;*****
A34012:
        PUSH  AF
        LD     A, (A17008)
        CP     1
        JR     C, A34024
A34020:
        POP   AF
        JP     A17416
;*****
A34024:
        LD     A, (A17988)
        CP     80
        JR     NZ, A34020
        POP   AF
        JP     A34263
;*****
A34035:
        LD     E, H
        LD     D, 0
        LD     H, D

```

```

    ADD    HL,HL
    ADD    HL,HL
    ADD    HL,HL
    PUSH   HL
    ADD    HL,HL
    ADD    HL,HL
    POP    BC
    ADD    HL,BC
    ADD    HL,HL
    ADD    HL,DE
    LD     DE,2815      ;VRAM address of TEXT80
screen image
    ADD    HL,DE
    EX     DE,HL
    RET
;*****
A34055:
    LD     HL,(A17001)
    CALL   A34035
    LD     HL,A33662
    LD     (HL),A
    LD     BC,1
    JP     WRITE_VRAM
;*****
A34071:
    PUSH   AF
    LD     A,(A28081)
    LD     C,A
    CP     68
    JP     C,A35459
    INC    C
    LD     D,8
A34084:
    LD     HL,65535
A34087:
    IN     A,(C)
    LD     B,A
    CP     255

```

```
JR      Z,A34103
AND     56
JR      NZ,A34118
LD      A,B
AND     1
JR      NZ,A34113
```

A34103:

```
DEC     HL
LD      A,H
OR      L
JR      NZ,A34087
DEC     D
JR      NZ,A34084
JR      A34124
```

;*****

A34113:

```
POP     AF
DEC     C
OUT     (C),A
RET
```

;*****

A34118:

```
CALL    A31435
JP      A24321
```

;*****

A34124:

```
CALL    A31435
LD      A,34
JP      A7950
```

;*****

A34132:

```
POP     AF
CP      32
JP      C,A34263
CP      128
JP      NC,A34263
PUSH    AF
PUSH    BC
```

```

        PUSH    DE
        PUSH    HL
        PUSH    IX
        PUSH    IY
A34151:
        CALL    A34071
        CALL    A34055
        LD      HL, (A17001)
        INC     H
        LD      A, (A16957)
        CP      H
        JR      NC, A34182
        LD      A, (A16956)
        LD      H, A
        INC     L
        CALL    A35017
        CP      L
        JR      NC, A34182
        DEC     L
A34179:
        CALL    A34194
A34182:
        LD      (A17001), HL
A34185:
        JP      A34612
;*****
        NOP
A34189:
        POP     HL
        POP     DE
        JP      A34436
;*****
A34194:
        PUSH    HL
        LD      DE, 2896      ;VRAM address
        LD      B, 23
A34200:
        PUSH    BC

```

```
PUSH DE
LD HL,A33662
LD BC,80
CALL READ_VRAM
POP HL
LD BC,80
OR A
SBC HL,BC
PUSH HL
LD DE,A33662
EX DE,HL
LD BC,80
CALL WRITE_VRAM
POP DE
LD HL,160
ADD HL,DE
EX DE,HL
POP BC
```

A34236:

```
DJNZ A34200
LD B,80
LD A,32
LD HL,A33662
```

A34245:

```
LD (HL),A
INC HL
DJNZ A34245
LD HL,A33662
LD DE,4656 ;VRAM address
LD BC,80
CALL WRITE_VRAM
POP HL
RET
```

;*****

A34263:

```
PUSH AF
PUSH BC
PUSH DE
```

```
PUSH HL
PUSH IX
PUSH IY
LD HL,A33915
LD BC,17
CPIR
JP NZ,A34694
LD HL,A33932
JP A17440
```

```
;*****
```

```
A34288:
```

```
PUSH AF
LD HL,A33920
LD BC,6
CPDR
PUSH BC
```

```
A34298:
```

```
LD A,(A64)
LD C,A
LD HL,(A30086)
ADD HL,BC
ADD HL,BC
LD E,(HL)
INC HL
LD D,(HL)
LD L,E
LD H,D
POP BC
ADD HL,BC
LD L,(HL)
LD H,B
ADD HL,DE
LD A,(HL)
OR A
JR Z,A34335
LD B,A
```

```
A34322:
```

```
INC HL
```



```

        LD      A, (HL)
        PUSH   BC
        PUSH   HL
        CALL   A34071
        POP    HL
        POP    BC
        DJNZ   A34322
        POP    AF
        RET

;*****
A34335:
        POP    AF
        XOR    A
        RET

;*****
A34338:
        LD      A, B
        CALL   A34288
        OR     A
        RET    NZ

A34344:
        POP    AF
        JP     A34185

;*****
A34348:
        CALL   A35181
        LD     A, 32
        LD     DE, 1920           ;byte count
        LD     HL, 2816          ;VRAM address
        CALL   FILL_VRAM

A34362:
        LD     HL, 256
        JP     A34182

;*****
A34368:
        CALL   A34338
        JR     A34362

;*****

```

A34373:

```
XOR    A
OR     L
CALL   NZ,A35087
POP    HL
LD     (A35085),HL
LD     A,13
CALL   A34071
LD     A,(A16956)
LD     (A17002),A
LD     B,10
```

A34395:

```
LD     HL,(A17001)
LD     A,(A16959)
CP     L
JR     Z,A34415
PUSH   HL
CALL   A35219
NOP
NOP
POP    HL
INC    L
JP     A34182
```

;*****

A34415:

```
LD     A,10
CP     B
JP     NZ,A34182
PUSH   HL
CALL   A35219
POP    HL
JP     A34179
```

;*****

```
NOP
NOP
NOP
NOP
NOP
```

```

        NOP
        NOP
;*****
A34436:
        LD     A,255
        LD     (A17009),A
        POP    BC
        POP    AF
        RET
;*****
A34444:
        LD     A,(A17008)
        CP     1
        JP     NC,A33980
        JP     A33972
;*****
;TEXT80 command.
;
A34455:
        PUSH   DE
        CALL   A28023
        LD     A,80
        LD     (A17988),A
        CALL   A35127
        LD     A,32
        LD     DE,1920
        LD     HL,2816
        CALL   FILL_VRAM
        LD     BC,20481
        LD     HL,5888
        LD     (A69),BC
        LD     (A71),HL
        LD     BC,20247
        PUSH   DE
        LD     DE,256
        INC    B
        INC    C
        LD     (A16993),BC

```

```
LD      (A16995),DE
LD      A,D
LD      (A16956),A
ADD     A,B
DEC     A
LD      (A16957),A
LD      A,E
POP     DE
LD      (A16958),A
ADD     A,C
DEC     A
LD      (A16959),A
LD      A,12
CALL    A11994
POP     DE
RET
```

```
;*****
```

```
A34534:
```

```
LD      A,(A16956)
CP      D
JR      Z,A34542
JR      NC,A34596
```

```
A34542:
```

```
LD      A,(A16957)
CP      D
JR      Z,A34550
JR      C,A34596
```

```
A34550:
```

```
LD      A,(A16958)
CP      E
JR      Z,A34558
JR      NC,A34596
```

```
A34558:
```

```
LD      A,(A16959)
CP      E
JR      Z,A34566
JR      C,A34596
```

```
A34566:
```

```

        PUSH    DE
        LD      A, 28
        CALL   A34288
        POP    DE
        OR     A
A34574:
        JP     Z, A34185
A34577:
        PUSH   DE
        LD     A, D
        ADD   A, 31
        PUSH  AF
        LD     A, E
        ADD   A, 32
        CALL  A34071
        POP   AF
        CALL  A34071
        POP   HL
        JP    A34182
;*****
A34596:
        JP    A34185
;*****
A34599:
        LD    HL, (A17001)
        LD    A, (A16958)
        CP    L
        JR    Z, A34574
        DEC   L
A34609:
        EX   DE, HL
        JR   A34566
;*****
A34612:
        CALL A34622
        POP  IY
        POP  IX
        JP  A34189

```

;*****

A34622:

```
LD HL,(A17001)
CALL A34035
LD HL,A33662
LD BC,1
CALL READ_VRAM
LD A,(A33662)
LD (A16955),A
RET
```

;*****

A34644:

```
LD HL,(A17001)
LD A,(A16957)
CP H
JR Z,A34656
INC H
```

A34654:

```
JR A34609
```

;*****

A34656:

```
LD A,(A16959)
CP L
```

A34660:

```
JP Z,A34185
INC L
LD A,(A16956)
```

A34667:

```
LD H,A
JR A34654
```

;*****

A34670:

```
LD HL,(A17001)
LD A,(A16956)
CP H
JR Z,A34682
DEC H
JR A34654
```

;*****

A34682:

LD A, (A16958)
CP L
JR Z, A34660
DEC L
LD A, (A16957)
JR A34667

;*****

A34694:

LD A, 32
JP A34151

;*****

A34699:

LD A, (A17001)
LD L, A
LD H, 1
CALL A34035
PUSH DE
LD HL, A33662
LD BC, 80
CALL READ_VRAM
LD A, (A17002)
LD C, A
LD B, 0
LD HL, A33661
ADD HL, BC
LD A, 81
SUB C
LD B, A
LD A, 32

A34734:

LD (HL), A
INC HL
DJNZ A34734
POP DE

A34739:

LD HL, A33662

```

        LD     BC,80
        JP     WRITE_VRAM
;*****
A34748:
        CALL  A34338
        CALL  A34699
A34754:
        JP     A34185
;*****
A34757:
        CALL  A34338
        CALL  A34699
        LD   HL,(A17001)
        LD   A,(A16959)
        CP   L
        JR   Z,A34754
        PUSH HL
        INC  L
        LD   A,(A16993)
        SUB  L
        LD   B,A
        LD   HL,0
        LD   DE,80
A34785:
        ADD  HL,DE
        DJNZ A34785
        POP  DE
        PUSH HL
        EX  DE,HL
        LD  H,1
        CALL A34035
        EX  DE,HL
        POP  DE
        LD  A,32
        CALL FILL_VRAM
        JR  A34754
;*****
A34805:

```



```
CALL A34338
LD A, (A17001)
LD L, A
LD H, 1
CALL A34035
PUSH DE
LD HL, A33662
LD BC, 80
CALL READ_VRAM
LD A, (A17002)
LD C, A
LD B, 0
LD HL, A33661
ADD HL, BC
LD E, L
LD D, H
INC HL
LD A, 80
SUB C
LD C, A
LDIR
LD A, 32
LD (A33741), A
POP DE
CALL A34739
JP A34185
```

```
;*****
```

```
A34858:
```

```
LD HL, (A17001)
PUSH HL
LD H, 1
CALL A34035
PUSH DE
LD HL, A33662
LD BC, 80
CALL READ_VRAM
LD A, (A17002)
LD C, A
```

```
LD      A,80
SUB     C
LD      C,A
LD      B,0
PUSH   BC
LD      HL,A33740
LD      DE,A33741
LDDR
INC     HL
LD      A,32
LD      (HL),A
POP    BC
LD      B,C
INC     B
```

A34903:

```
PUSH   BC
PUSH   HL
LD      A,28
CALL   A34288
OR     A
JR     Z,A34969
LD      HL,(A17001)
LD      A,H
ADD    A,31
PUSH   AF
LD      A,L
ADD    A,32
CALL   A34071
POP    AF
CALL   A34071
POP    HL
LD      A,(HL)
INC    HL
PUSH   HL
CALL   A34071
CALL   A34055
LD      HL,(A17001)
INC    H
```

```

        LD      (A17001),HL
        POP     HL
        POP     BC
A34949:
        DJNZ   A34903
        POP     DE
        CALL   A34739
        POP     HL
        LD      (A17001),HL
        PUSH   HL
        LD      A,28
        CALL   A34288
        POP     DE
        JP      A34577
;*****
A34969:
        POP     HL
        POP     BC
        POP     DE
        POP     HL
        JP      A34185
;*****
A34976:
        LD      DE,2816      ;VRAM address
        LD      B,24
A34981:
        PUSH   BC
        PUSH   DE
        LD      HL,A33662
        PUSH   HL
        LD      BC,80
        CALL   READ_VRAM
        POP     HL
        XOR    A
        LD      (A31016),A
        LD      A,(A17988)
        LD      C,A
        CALL   A31049

```

```
POP    DE
LD     HL,80
ADD    HL,DE
EX     DE,HL
POP    BC
DJNZ   A34981
JP     A34185
```

```
;*****
```

```
A35017:
```

```
PUSH   HL
LD     A,13
CALL   A34071
CALL   A35219
NOP
NOP
POP    HL
LD     A,(A16959)
RET
```

```
;*****
```

```
A35033:
```

```
LD     A,7
CALL   A34071
JP     A17945
```

```
;*****
```

```
A35041:
```

```
LD     A,(A17008)
OR     A
JP     NZ,A17581
LD     A,(A17988)
CP     80
JP     NZ,A17581
JP     A34185
```

```
;*****
```

```
A35059:
```

```
LD     A,255
PUSH   AF
LD     HL,6
ADD    HL,BC
```

```

        PUSH    HL
        JP     A34298
;*****
A35070:
        LD     A, (A17008)
        OR     A
        JR     NZ, A35082
        LD     A, (A17988)
        CP     80
        RET    Z
A35082:
        POP    HL
        POP    HL
        JP     (HL)
;*****
A35085:
        DB     0           ;
A35086:
        DB     0           ;
;*****
A35087:
        LD     HL, A26141
        PUSH   HL
        CALL  A35070
        POP    HL
        LD     BC, 0
        CALL  A35059
        LD     A, 255
        LD     (A35085), A
        RET
;*****
A35107:
        LD     HL, A26164
        PUSH   HL
        CALL  A35070
        POP    HL
        LD     BC, 2
        CALL  A35059

```

```
LD    A,255
LD    (A35086),A
RET
```

```
;*****
```

```
A35127:
```

```
LD    HL,A26154
PUSH  HL
CALL  A35070
POP   HL
LD    BC,1
CALL  A35059
LD    BC,3
CALL  A35059
LD    HL,0
LD    (A35085),HL
RET
```

```
;*****
```

```
A35154:
```

```
LD    HL,(A35085)
PUSH  HL
CALL  A35127
LD    A,22
CALL  A34288
CALL  A34699
POP   HL
PUSH  HL
XOR   A
OR    H
CALL  NZ,A35107
POP   HL
PUSH  HL
JP    A34373
```

```
;*****
```

```
A35181:
```

```
LD    HL,(A35085)
PUSH  HL
CALL  A35127
LD    A,12
```

```

        CALL  A34288
        OR    A
        JR    Z,A35215
        POP   HL
A35197:
        PUSH  HL
        XOR   A
        OR    H
        CALL  NZ,A35107
        POP   HL
        PUSH  HL
        XOR   A
        OR    L
        CALL  NZ,A35087
        POP   HL
        LD    (A35085),HL
        RET
;*****
A35215:
        POP   HL
        JP    A34344
;*****
A35219:
        LD    HL,(A35085)
        PUSH  HL
        CALL  A35127
        LD    A,10
        CALL  A34071
        POP   HL
        JR    A35197
;*****
;FORMAT drive command.
;
A35234:
        CALL  A1500
        PUSH  DE
        LD    A,5
        CP    L

```

```

        JR      Z,A35248
        INC    A
        CP     L
        JP     NZ,A7936
A35248:
        LD     C,A
        LD     B,0
        LD     HL,A23272      ;drive-to-device table
        ADD    HL,BC
        LD     A,(HL)
        PUSH  AF
        PUSH  AF
        LD     HL,(FCB_DATA_ADDR)      ;EOS start of
DTA2
        PUSH  HL
        LD     BC,1024
A35265:
        LD     A,255
        LD     (HL),A
        INC   HL
        DEC   BC
        LD     A,C
        OR    B
A35272:
        JR     NZ,A35265
        POP   HL
        POP   AF
        LD     DE,0FACEH      ;Ye Olde Magic FACE
                                ;writing a block full of
0FFH data to this
                                ;block invokes the format
operation in the
                                ;floppy disk firmware (6801
microcontroller)
        CALL  _WRITE_BLOCK
        JP     NZ,A24321
        POP   AF
        LD     HL,A35294

```



```

        CALL  A25307
        POP   DE
        RET
;*****
A35294:
        DB    "SB1.x"      ;default FORMAT INIT volume
        DB    3
;*****
;MIB2 serial board initialization data.
;Format:  port, # bytes to send, data.  255=end of
table.
;
A35300:
        DB    1,2,255,247      ;wakeup
        DB    16,2,19,7       ;
A35306  EQU    A35300+6       ;data/parity/stop port
2
        DB    24,2,19,7       ;
A35310  EQU    A35300+10      ;data/parity/stop port
1
        DB    17,1,204        ;
A35314  EQU    A35300+14      ;baudrate port 2
        DB    25,1,136        ;
A35317  EQU    A35300+17      ;baudrate port 1
        DB    18,1,5          ;
        DB    26,1,5          ;
        DB    20,1,240        ;
        DB    30,1,255        ;
        DB    255             ;end of table
;*****
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
;*****
;Check for SERIAL port status:  Eve/Orphanware/MIB2.

```

```

;On entry, B=4.
;
A35337:
    LD    HL,A30680    ;Eve/Orphanware/ADAMlink
table
A35340:
    CP    (HL)        ;is it there?
    JP    Z,A31175    ;YES
    INC   HL          ;NO, so skip ahead to next
    INC   HL
    INC   HL
    INC   HL
    DJNZ  A35340      ;keep going 'til done
    LD    B,2         ;not found, so try MIB2
    LD    HL,A33871   ;MIB2 table
    JP    A31163      ;keep looking
;*****
A35358:
    LD    HL,A33871   ;MIB2 table
    LD    B,2         ;counter
A35363:
    CP    (HL)        ;is it there?
    JR    Z,A35375    ;YES
    INC   HL          ;NO, so skip ahead to next
    INC   HL
    INC   HL
    INC   HL
    DJNZ  A35363      ;keep going 'til done
A35372:
    JP    A29261      ;not found
;*****
A35375:
    CP    1           ;was it MIB2 port 1?
    JR    Z,A35383    ;YES
A35379:
    POP   BC          ;NO, so clear return address
from stack
    JP    A31117

```

;*****

A35383:

PUSH AF
INC DE
LD A, (DE)
DEC DE
CP 185
JR Z, A35394
POP AF
JR A35372

;*****

A35394:

POP AF
JR A35379

;*****

A35397:

PUSH AF
PUSH DE
LD D, 8
LD A, C
CP 68
JP NC, A30949
LD C, 25
CP 2
JR Z, A35415
LD C, 17

A35415:

LD HL, 65535

A35418:

CALL A35542
IN A, (C)
CP 255
JR Z, A35431
BIT 3, A
JR NZ, A35442

A35431:

DEC HL
LD A, H

```

OR      L
JR      NZ,A35418
DEC     D
JR      NZ,A35415
JP      A30660
;*****
A35442:
POP     DE
A35443:
INC     C
INC     C
POP     AF
OUT     (C),A
PUSH   AF
LD      A,C
LD      C,2
CP      27
JR      Z,A35457
INC     C
A35457:
POP     AF
RET
;*****
A35459:
NOP
LD      D,8
LD      C,17
CP      2
JR      Z,A35470
LD      C,25
A35470:
LD      HL,65535
A35473:
CALL   A35542
IN     A,(C)
CP     255
JR     Z,A35486
BIT    3,A

```

JR NZ,A35443
A35486:

DEC HL
LD A,H
OR L
JR NZ,A35473
DEC D
JR NZ,A35470
JP A34124

;*****

A35497:

LD HL,A28077
LD B,(HL)
INC HL
LD C,(HL)
CP 2
JR Z,A35517
LD A,C
LD (A35310),A
LD A,B
LD (A35317),A
JR A35525

;*****

A35517:

LD A,C
LD (A35306),A
LD A,B
LD (A35314),A

A35525:

LD HL,A35300

A35528:

LD C,(HL)
LD A,C
INC A
INC HL
JR Z,A35540
LD B,(HL)
INC HL

```

        OTIR
        JR      A35528
;*****
A35540:
        POP    AF
        RET
;*****
A35542:
        IN     A,(C)
        AND    240
        RET    Z
        JP     A34118
;*****
LOMEM:
;
NVAR$ EQU    71      ;total number of variable commands
;
;*****
;Types, vectors and offsets for variable commands.
;Format:  type, vector, offset into ASCII table to
find string.
;
;SPC()
        DB    136      ;numeric/argument
        DW    A7948
        DW    $SPC-VASCII
;TAB()
        DB    136      ;numeric/argument
        DW    A7948
        DW    $TAB-VASCII
;ERRNUM()
        DB    136      ;numeric/argument
        DW    A10814
        DW    $ERRNUM1-VASCII
;ABS()
        DB    136      ;numeric/argument
        DW    A2276
        DW    $ABS-VASCII

```

```

;ASC ()
    DB      136      ;numeric/argument
    DW      A10351
    DW      $ASC-VASCII
;ATN ()
    DB      136      ;numeric/argument
    DW      A4180
    DW      $ATN-VASCII
;CHR$ ()
    DB      168      ;string/argument
    DW      A10371
    DW      $CHR$-VASCII
;COS ()
    DB      136      ;numeric/argument
    DW      A3946
    DW      $COS-VASCII
;EXP ()
    DB      136      ;numeric/argument
    DW      A3816
    DW      $EXP-VASCII
;FRE ()
    DB      136      ;numeric/argument
    DW      A10192
    DW      $FRE1-VASCII
;INT ()
    DB      136      ;numeric/argument
    DW      A10672
    DW      $INT-VASCII
;LEFT$ ()
    DB      168      ;string/argument
    DW      A10508
    DW      $LEFT$-VASCII
;LEN ()
    DB      136      ;numeric/argument
    DW      A10454
    DW      $LEN-VASCII
;LOG ()
    DB      136      ;numeric/argument

```

```

        DW      A3604
        DW      $LOG-VASCII
;MID$ ( )
        DB      168      ;string/argument
        DW      A10563
        DW      $MID$-VASCII
;PDL ( )
        DB      136      ;numeric/argument
        DW      A11619
        DW      $PDL-VASCII
;PEEK ( )
        DB      136      ;numeric/argument
        DW      A31870
        DW      $PEEK-VASCII
;POS ( )
        DB      136      ;numeric/argument
        DW      A10844
        DW      $POS1-VASCII
;RIGHT$ ( )
        DB      168      ;string/argument
        DW      A10529
        DW      $RIGHT$-VASCII
;RND ( )
        DB      136      ;numeric/argument
        DW      A4696
        DW      $RND1-VASCII
;SCRN ( )
        DB      136      ;numeric/argument
        DW      A11268
        DW      $SCRN-VASCII
;SGN ( )
        DB      136      ;numeric/argument
        DW      A2285
        DW      $SGN-VASCII
;SIN ( )
        DB      136      ;numeric/argument
        DW      A3954
        DW      $SIN-VASCII

```



```

;SQR ()
    DB      136      ;numeric/argument
    DW      A3678
    DW      $SQR-VASCII
;STR$ ()
    DB      168      ;string/argument
    DW      A10411
    DW      $STR$-VASCII
;TAN ()
    DB      136      ;numeric/argument
    DW      A3912
    DW      $TAN-VASCII
;USR ()
    DB      136      ;numeric/argument
    DW      A10073
    DW      $USR-VASCII
;VAL ()
    DB      136      ;numeric/argument
    DW      A10309
    DW      $VAL-VASCII
;VPOS ()
    DB      136      ;numeric/argument
    DW      A10857
    DW      $VPOS1-VASCII
;TIME ()
    DB      136      ;numeric/argument
    DW      A29275
    DW      $TIME-VASCII
;TIME$
    DB      160      ;string/no argument
    DW      A29572
    DW      $TIME$-VASCII
;DATE ()
    DB      136      ;numeric/argument
    DW      A32259
    DW      $DATE-VASCII
;DATE$
    DB      160      ;string/no argument

```

```

        DW      A29618
        DW      $DATE$-VASCII
;SER ()
        DB      136      ;numeric/argument
        DW      A31086
        DW      $SER-VASCII
;MOD ()
        DB      136      ;numeric/argument
        DW      A29464
        DW      $MOD-VASCII
;HEX$ ()
        DB      168      ;string/argument
        DW      A29739
        DW      $HEX$-VASCII
;COLOR ()
        DB      136      ;numeric/argument
        DW      A30417
        DW      $COLOR-VASCII
;ADDR ()
        DB      136      ;numeric/argument
        DW      A29891
        DW      $ADDR-VASCII
;MEM ()
        DB      136      ;numeric/argument
        DW      A29848
        DW      $MEM-VASCII
;SPEED
        DB      128      ;numeric/no argument
        DW      A29966
        DW      $SPEED-VASCII
;ROT
        DB      128      ;numeric/no argument
        DW      A29973
        DW      $ROT-VASCII
;SCALE
        DB      128      ;numeric/no argument
        DW      A29978
        DW      $SCALE-VASCII

```

```

;WIDTH
    DB      136      ;numeric/argument
    DW      A29945
    DW      $WIDTH-VASCII
;PROMPT
    DB      128      ;numeric/no argument
    DW      A29983
    DW      $PROMPT-VASCII
;VER ()
    DB      136      ;numeric/argument
    DW      A29927
    DW      $VER-VASCII
;SPC$ ()
    DB      168      ;string/argument
    DW      A29840
    DW      $SPC$-VASCII
;STRING$ ()
    DB      168      ;string/argument
    DW      A29783
    DW      $STRING$-VASCII
;AF
    DB      128      ;numeric/no argument
    DW      A30138
    DW      $AF-VASCII
;BC
    DB      128      ;numeric/no argument
    DW      A30144
    DW      $BC-VASCII
;DE
    DB      128      ;numeric/no argument
    DW      A30149
    DW      $DE-VASCII
;HL
    DB      128      ;numeric/no argument
    DW      A30154
    DW      $HL-VASCII
;IX
    DB      128      ;numeric/no argument

```

```

        DW      A30159
        DW      $IX-VASCII
;IY
        DB      128      ;numeric/no argument
        DW      A30164
        DW      $IY-VASCII
;BIT ()
        DB      136      ;numeric/argument
        DW      A30317
        DW      $BIT-VASCII
;SET ()
        DB      136      ;numeric/argument
        DW      A30330
        DW      $SET-VASCII
;RES ()
        DB      136      ;numeric/argument
        DW      A30336
        DW      $RES-VASCII
;HDRV
        DB      128      ;numeric/no argument
        DW      A30718
        DW      $HDRV-VASCII
;AND ()
        DB      136      ;numeric/argument
        DW      A30359
        DW      $AND-VASCII
;OR ()
        DB      136      ;numeric/argument
        DW      A30365
        DW      $OR-VASCII
;XOR ()
        DB      136      ;numeric/argument
        DW      A30371
        DW      $XOR-VASCII
;CPL ()
        DB      136      ;numeric/argument
        DW      A30377
        DW      $CPL-VASCII

```

```

;ZF
    DB      128      ;numeric/no argument
    DW      A30385
    DW      $ZF-VASCII

;CF
    DB      128      ;numeric/no argument
    DW      A30400
    DW      $CF-VASCII

;ERRNUM
    DB      128      ;numeric/no argument
    DW      A10814
    DW      $ERRNUM2-VASCII

;POS
    DB      128      ;numeric/no argument
    DW      A10847
    DW      $POS2-VASCII

;VPOS
    DB      128      ;numeric/no argument
    DW      A10860
    DW      $VPOS2-VASCII

;FRE
    DB      128      ;numeric/no argument
    DW      A10192
    DW      $FRE2-VASCII

;RND
    DB      128      ;numeric/no argument
    DW      A30407
    DW      $RND2-VASCII

;DSIZE()
    DB      136      ;numeric/argument
    DW      A31790
    DW      $DSIZE-VASCII

;CLK
    DB      128      ;numeric/no argument
    DW      A31559
    DW      $CLK-VASCII

;TERM
    DB      128      ;numeric/no argument

```

```

        DW      A31565
        DW      $TERM-VASCII
;*****
;Variable command ASCII strings.
;Format: length, word.
;
VASCII:
;
VTABLEN      EQU      VASCII-LOMEM          ;length of
permanent variable table
;
$SPC:
        DB      3
        DB      "SPC"
;
$TAB:
        DB      3
        DB      "TAB"
;
$ERRNUM1:
        DB      6
        DB      "ERRNUM"
;
$ABS:
        DB      3
        DB      "ABS"
;
$ASC:
        DB      3
        DB      "ASC"
;
$ATN:
        DB      3
        DB      "ATN"
;
$CHR$:
        DB      3
        DB      "CHR"

```

```
;  
$COS:      DB      3  
           DB      "COS"  
;  
$EXP:      DB      3  
           DB      "EXP"  
;  
$FRE1:     DB      3  
           DB      "FRE"  
;  
$INT:      DB      3  
           DB      "INT"  
;  
$LEFT$:    DB      4  
           DB      "LEFT"  
;  
$LEN:      DB      3  
           DB      "LEN"  
;  
$LOG:      DB      3  
           DB      "LOG"  
;  
$MID$:     DB      3  
           DB      "MID"  
;  
$PDL:      DB      3  
           DB      "PDL"  
;  
$PEEK:
```

```
        DB      4
        DB      "PEEK"
;
$POS1:
        DB      3
        DB      "POS"
;
$RIGHT$:
        DB      5
        DB      "RIGHT"
;
$RND1:
        DB      3
        DB      "RND"
;
$SCRN:
        DB      4
        DB      "SCRN"
;
$SGN:
        DB      3
        DB      "SGN"
;
$SIN:
        DB      3
        DB      "SIN"
;
$SQR:
        DB      3
        DB      "SQR"
;
$STR$:
        DB      3
        DB      "STR"
;
$TAN:
        DB      3
        DB      "TAN"
```



```
;
$USR:
      DB      3
      DB      "USR "
;
$VAL:
      DB      3
      DB      "VAL "
;
$VPOS1:
      DB      4
      DB      "VPOS "
;
$TIME:
      DB      4
      DB      "TIME "
;
$TIME$:
      DB      4
      DB      "TIME "
;
$DATE:
      DB      4
      DB      "DATE "
;
$DATE$:
      DB      4
      DB      "DATE "
;
$SER:
      DB      3
      DB      "SER "
;
$MOD:
      DB      3
      DB      "MOD "
;
$HEX$:
```

```
        DB      3
        DB      "HEX"
;
$COLOR:
        DB      5
        DB      "COLOR"
;
$ADDR:
        DB      4
        DB      "ADDR"
;
$MEM:
        DB      3
        DB      "MEM"
;
$SPEED:
        DB      5
        DB      "SPEED"
;
$ROT:
        DB      3
        DB      "ROT"
;
$SCALE:
        DB      5
        DB      "SCALE"
;
$WIDTH:
        DB      5
        DB      "WIDTH"
;
$PROMPT:
        DB      6
        DB      "PROMPT"
;
$VER:
        DB      3
        DB      "VER"
```

```
;
$SPC$:
    DB      3
    DB      "SPC"
```

```
;
$string$:
    DB      6
    DB      "STRING"
```

```
;
$AF:
    DB      2
    DB      "AF"
```

```
;
$BC:
    DB      2
    DB      "BC"
```

```
;
$DE:
    DB      2
    DB      "DE"
```

```
;
$HL:
    DB      2
    DB      "HL"
```

```
;
$IX:
    DB      2
    DB      "IX"
```

```
;
$IY:
    DB      2
    DB      "IY"
```

```
;
$BIT:
    DB      3
    DB      "BIT"
```

```
;
$SET:
```

```
        DB      3
        DB      "SET"
;
$RES:
        DB      3
        DB      "RES"
;
$HDRV:
        DB      4
        DB      "HDRV"
;
$AND:
        DB      3
        DB      "AND"
;
$OR:
        DB      2
        DB      "OR"
;
$XOR:
        DB      3
        DB      "XOR"
;
$CPL:
        DB      3
        DB      "CPL"
;
$ZF:
        DB      2
        DB      "ZF"
;
$CF:
        DB      2
        DB      "CF"
;
$ERRNUM2:
        DB      6
        DB      "ERRNUM"
```

```

;
$POS2:
    DB      3
    DB      "POS"
;
$VPOS2:
    DB      4
    DB      "VPOS"
;
$FRE2:
    DB      3
    DB      "FRE"
;
$RND2:
    DB      3
    DB      "RND"
;
$DSIZE:
    DB      5
    DB      "DSIZE"
;
$CLK:
    DB      3
    DB      "CLK"
;
$TERM:
    DB      4
    DB      "TERM"
;*****
;Start of string space.
;
$SPACE:
    DB      "SmartBASIC 1.x version 20Y",0
    DB      "(c) 1991 by Richard F. Drushel",0
    DB      "All Rights Reserved",0
    DB      "Hi Joan",0
    DB      "Hi Christina",0
    DB      "Hi Elanor",0

```

```
DB      "Hi Herman M.",0
DB      "Hi George K.",0
DB      "Hi George H.",0
DB      "Hi Ron C.",0
DB      "Hi Alan N.",0
DB      "Hi Steve M.",0
DB      "Hi Zonker",0
DB      "Hi Mark G.",0
DB      "Hi Jim N.",0
DB      "Cathy and Jan can go to @#$%!",0
DB      "If you are reading this, hats off, but
sorry, no prize...",0
DB      "You must be *SOMEWHAT* clever, however,
if you've gotten this far.",0
DB      "In fact, you're probably a hacker.",0
DB      "Good luck if you want to hack around in
*HERE*!",0
DB      "Rumor has it that there is an EOS virus
embedded in the code.",0
DB      "If your system crashes, you can blame
it on me.",0
DB      "There, doesn't *THAT* make you feel
better now?",0
```

.DEPHASE

END

→