SMARTBASIC DISASSEMBLY

The following package contains a detailed disassembly of Coleco's original SMARTBASIC. It has been put together as a guide for the hacker who is interested in knowing what makes SMARTBASIC tick. The listing is not intended to be a re-assembly package to make modifications to SMARTBASIC; it only provides details of the operations, tables, vectors, messages, etc.

I would like to thank all those who knowingly or not have contributed to the compilation of the package. They include:

Programmers	hints have been picked up in some of your programs;
Newsletter Contributors	several interpretations and bug fixes came from newsletters;
ANN DISKS	refinements were possible through articles found there;
Peter and Ben Hinkle	their Hacker's Guide to ADAM was an invaluable tool.

Despite my efforts to precicely document the structure and operation of SMARTBASIC, I cannot guarantee the accuracy of the information contained in these listings. I have just converted the data files from SMARTWRITER to CP/M (with great difficulty I might add). There are still some minor formatting problems; typographical errors and other technical glitches may invalidate some of the information. Should you discover any of these or have additional information to provide about any of the routines, please contact me. I will gladly make the changes and provide you with and updated copy of the listing.

WHAT YOU GET

Map	a general idea of where routines are located
Numerical Listing	pointers to the major routines, tables, and code sections (including bug fixes and enhancements)
Alphabetical Listing	the numerical listing sorted in alphabetical order; while not a true index, it should be helpful.
Disassembly Listing	the detailed analysis of SMARTBASIC

ADDRESS	OPERATION		ascii	DECIMAL			COMMENTS
14991:3A8F	CALL	3C04:15364	..<	205	4	60	get variable
14994:3A92	AND	01:1	..	230	1		
14996:3A94	PUSH	AF	.	245			
14997:3A95	CALL	3E36:15926	.6>	205	54	62	check for =
15000:3A98	POP	AF	.	241			
15001:3A99	JR	Z,3AB4:15028	(.	40	25		check variable type
15003:3A9B	CALL	2E91:11921	...	205	145	46	print the following error
15006:3A9E	13		length of message				
15007:3A9F	Real		first word				
15019:3AAB	Variable		start address of second word				

All addresses are given in decimal and in HEX;
the ASCII section may be useful when compare or other operations take place on ASCII characters;
the decimal section gives you the corresponding POKES associated witht he routines;
the comments section helps define what is happening.
There are several tables in the listing in different formats which should be self explanatory.
I hope you get some use out of the listings. For additional information, contact:

Guy Cousineau
1059 Hindley Street
OTTAWA Canada
K2B 5L9

MAP TO SMARTBASIC

HEX	SECTION	DESCRIPTION
-6B0E		
|	PADDLES	Game controller decoding
-6918		
-6917		
|	GRAPHICS	The high level HGR functions like PLOT and DRAW also INVERSE FLASH NORMAL VTAB HTAB (WHY ARE THEY HEREÉ)
-631A		
-6319		
|	MEDIUM INPUT OUTPUT	This section handles all media related functions CATALOG, DELETE, RENAME, INIT, etc. as well as all file I/O such as: LOAD, SAVE, OPEN, READ, WRITE, etc.
-4C0F		
-4C0E		
|	OUTPUT ROUTINES	This section handles character output to screen and/or printer. Lo-Res (GR) graphics are handled here and some of Hi-Res (HGR)
-4296		
-4295		
|	TABLES	Various tables and pointers used throughout Basic
-3ED9		
-3E9D		
|	BOOT	The cold boot routine resides here; it is later overwriten by the input buffers
-3ED6		
-3E9C		
|	PARSER	This portion interprets commands from the console or. medium. It converts them to tokens and processes syntax errors.
-2DFE		
-2DFD		
|	INTERPRETER	This is the part of Basic which RUNs programs. It interprets commands in token form, performs the necessary set up then calls routines in the input, output, or graphics routines
-1757		
-1756		
|	MATH PROCESSOR	Floating point functions ASCII INTEGER FLOAT conversions Other math functions required required by the Basic interpreter
-05DC		
-05DB		
|		Copyright message, error messages, offset tables, etc
-041E		
-041D		
|		Command tokens and parse vectors
-0110		
-010F		
|		Basic version and integers 10000 1000 100 10 1
-0103		
-0102		
|		Initially a jump to BOOT, later replaced by jump to processing loop.
-0100		
-00FF		
|	ZERO PAGE	Used only for FLASH NMI which starts at 66H
-0000		

```
00102 0066    Non Maskable Interrupt used for FLASH
00159 009F    FLASH reset value
00256 0100    Cold Start address
00260 0104    Version of Basic
00262 0106    Integers 10000,1000,100,10,1
00272 0110    Command Parse Vector pointers
00818 0332    Operator Token table
00938 03AA    Command Parsing Vectors
01055 041F    Hi Cathy
01064 0428    Fatal System Error
01083 0436    Coleco SmartBasic message
01145 0479    Prompt pointer - Count, prompt1, prompt2
01148 047C    Command Separator (the COLON) - count, character
01150 047E    Return pointer - count and character <CR>
01152 0480    Error messages - Length, message
01463 05B7    Errnum Offsets to error messages
01485 05CD    Integers 1 through 15??
01500 05DC    Load HL with a line number
01519 05EF    Load BC with number for GOTO
01553 0611    Print floating Point Number
01569 0621    Add (HL) to FPA1
01575 0627    Subtract FPA2 from FPA1
01583 062F    Add FPA1 and FPA2
01739 06CB    Subtract DEBC from FPA1
01832 0728    Move FPA2 to FPA1
01851 073B    Compare signs of FPA1 and FPA2
01875 0753    Multiply FPA1 by 2
01885 075D    Multiply FPA1 by (HL)
01888 0760    Multiply FPA1 by FPA2
02018 07E2    Divide FPA1 by FPA2
02140 085C    Divide FPA2 by 2
02276 08E4    ABS() Execution - reset sign of FPA1
02285 08ED    SGN() Execution - get sign of FPA1
02326 0916    Toggle FPA1 or FPA2 - if zero make non-zero
02354 0932    Load HL with INT(FPA1)
02407 0967    Load FPA1 or FPA2 (depends on carry) with HL (integer)
02488 09B8    Compare FPA1 and FPA2 carry set if FPA1 bigger
02576 0A10    Load FPA1 with ASCII number from (DE)
02708 0A94    Multiply DEBC by 2
02875 0B3B    Floating point number from 1E-38 to 1E38
03260 0CBC    Write FPA1 in ASCII to input buffer (3F77)
03420 0D5C    Multiply DEBC by 10
03580 0DFC    Divide FPA1 by 10 only if greater than 10
03604 0E14    LOG() Execution
03678 0E5E    SQR() Execution
03696 0E70    Exponent FPA1^FPA2
03816 0EE8    EXP() Execution
03900 0F3C    Invert FPA1 (1/FPA1)
03912 0F48    TAN() Execution - SIN/COS
03946 0F6A    COS() Execution - SIN(x+pi/2)
03954 0F72    SIN() Execution
04156 103C    Power series calculator for 2 squares algorithms
04180 1054    ATN() Execution
04270 10AE    Power series calculator for multiplication algorithms
04356 1104    Move FPA1 to FPA2
04357 1117    Move (HL) to FPA2
04348 1120    Load FPA1 with 1
04396 112C    Push FPA1 on Stack
04411 113B    Pop FPA1 from Stack
04426 114A    Push FPA2 on Stack
04441 1159    Pop FPA2 from Stack
04479 117F    ABS of FPA1 i.e. make it positive
04497 1191    Constants FP and INT used by power series calculators
04696 1258    RND() Execution
04789 12B5    Push FPA1 to Stack while checking for string pointer
04835 12E3    Pop FPA1 from Stack while checking for string pointer
04869 1305    Move FPA1 to FPA2 while checking for string pointer
04899 1323    Move FPA2 to FPA1 while checking for string pointer
04929 1341    Check token of incoming number to decide return address
04982 1376    Get integer from 0-9 from token line to FPA1 or FPA2
04995 1383    Get integer from 10-255 from token line to FPA1 or FPA2
05008 1390    Get integer >255 from token line to FPA1 or FPA2
05023 139F    Load value from token line to FPA1
05041 13B1    Load value from token line to FPA2
```

```
05059 13C3    Move FP number from (BC) to (HL)
05078 13D6    Move integer from (BC) to (HL)
05095 13E7    Move string number from (BC) to (HL)
05135 140F    Move variable value from token line to FPA1
05141 1415    Move variable value from token line to FPA2
05167 142F    Move variable to FPA1
05374 14FE    Move number from token line (DE) to FPA1 or FPA2
05407 151F    Move string from (HL) to (DE)
05417 1529    ADD (+) Execution
05468 155C    SUBTRACT (-) Execution
05475 1563    MULTIPLY (*) Execution
05482 156A    DIVIDE (/) Execution
05489 1571    EXPONENT (^) Execution
05496 1578    AND Execution
05513 1589    OR Execution
05527 1597    LESS THAN (<) Execution
05541 15A5    GREATER THAN (>) Execution
05557 15B5    LESS EQUAL (<=) Execution
05573 15C5    GREATER EQUAL (>=) Execution
05587 15D3    NOT EQUAL (<>) Execution
05617 15F1    EQUAL (=) Execution
05636 1604    Compare strings pointed to by FPA1 and FPA2
05654 1616    NOT Execution
05760 1680    Find operator in token line
05797 16A5    Operator order of precedence and jump table for execute
05837 16CD    Get a number (int, FP, STRING, etc) from token line to FPA1
05861 16E5    Get number from token line to FPA2
05885 16FD    Evaluate equation
05939 1733    Get equation from token line
05975 1757    Stack set up
05994 176A    Get line number for RUN
06016 1780    Get next line number
06047 179F    END Execution
06069 17B5    TRACE check during RUN
06112 17E0    Command Execution routine
06159 180F    RUN Execution
06190 182E    Get next program line
06247 1867    LET Execution
06336 18C0    TRACE Execution
06341 18C5    NOTRACE Execution
06346 18CA    BREAK Execution
06351 18CF    NOBREAK Execution
06356 18D4    NEW Execution - POKE 201 to enable MERGE, POKE 205 to restore
06378 18EA    STOP Execution
06387 18F3    CONT Execution
06423 1917    Command Execution jump table by token
06555 199B    Check variable type
06729 1A49    DIM for Floating Point
06942 1B1E    DIM Execution
07066 1B9A    Multiply HL by DE
07097 1BB9    DIM for strings
07173 1C05    Check length of DATA or DIM
07264 1C60    Define string
07299 1C83    Stack check for underflow
07311 1C8F    String space check
07357 1CBD    LIST a range of lines
07407 1CEF    LIST Execution
07555 1D83    DEL Execution
07705 1E19    IF Execution
07740 1E3C    PRINT Execution
07900 1EDC    Error handler for runtime errors
08109 1FAD    CLRERR Execution
08144 1FB2    ONERR Execution
08141 1FCD    CLEAR Execution
08244 2034    DEF Execution
08313 2079    RESUME Execution
08342 2096    GOTO Execution
08381 20BD    ON Execution
08419 20E3    DATA Execution
08419 20E3    REM Execution
08427 20EB    GOSUB Execution
08477 211D    RETURN Execution
08493 212D    POP Execution
08516 2144    ON GOSUB Execution
08557 216D    FOR Execution
08749 222D    NEXT Execution
```

```
08957 22FD    INPUT Execution
09003 232B    INPUT prompt character
09068 236C    INPUT floating Point
09229 240D    INPUT string
09336 2478    INPUT integer
09378 24A2    GET Execution
09395 24B3    GET number byte
09464 24F8    GET string byte
09482 250A    RESTORE Execution
09499 251B    READ Execution (read data)
09591 2577    READ integer
09602 2582    READ string
09705 25E9    READ quoted string
09778 2632    READ floating point
09987 2703    Get integer from token line
10042 273A    CALL Execution
10073 2759    USR Execution
10091 276B    PEEK Execution
10104 2778    POKE Execution
10126 278E    WAIT Execution
10164 27B4    & Execution
10192 27D0    FRE() Execution
10309 2845    VAL() Execution
10351 286F    ASC() Execution
10371 2883    CHR$() Execution
10411 28AB    STR$() Execution
10454 28D6    LEN() Execution
10464 28E0    String length determination (for LEFT RIGHT MID)
10508 290C    LEFT$() Execution
10529 2921    RIGHT$() Execution
10553 2939    MID$() Execution
10616 2978    String cut routine
10672 29B0    INT() Execution
10814 2A3E    ERRNUM() Execution
10832 2A50    SPEED Execution
10844 2A5C    POS() Execution
10857 2A69    VPOS() Execution
10870 2A76    LOMEM Execution
11010 2B02    HIMEM Execution
11050 2B2A    FLASH Execution vector
11055 2B2F    INVERSE Execution vector
11060 2B34    NORMAL Execution vector
11065 2B39    TEXT Execution vector
11070 2B3E    GR Execution vector
11075 2B43    HGR Execution vector
11080 2B48    HGR2 Execution vector
11085 2B4D    SHLOAD Execution vector (points to a return)
11090 2B52    HOME Execution vector
11099 2B5B    COLOR Execution
11119 2B6F    HCOLOR Execution
11139 2B83    PLOT Execution
11170 2BA2    HLIN Execution
11219 2BD3    VLIN Execution
11268 2C04    SCRN() Execution
11320 2C38    HTAB Execution
11330 2C42    VTAB Execution
11351 2C57    DRAW Execution
11405 2C8D    XDRAW Execution
11459 2CC3    ROT Execution
11473 2CD1    SCALE Execution
11487 2CDF    HPLOT Execution
11619 2D63    PDL() Execution
11756 2DEC    STORE Execution
11764 2DF4    RECALL Execution
11776 2E00    Get non-control character from (DE)
11786 2E0A    Compare strings at (HL) and (DE)
11838 2E3E    Compare strings of unknown length
11851 2E4B    Compare strings of known length
11864 2E58    Check (DE) for letter
11884 2E6C    Check (DE) for number
11892 2E74    NEW initialization
11920 2E90    Print parse errors
11994 2EDA    Print character in 'A' to current PR routine
12010 2EEA    Print character in 'A' to screen and printer using SPEED
12043 2F0B    Print character in 'A' to screen only using SPEED
12058 2F1A    PR initialization
12084 2F34    IN initialization
```

```
12110 2F4E    Print length encoded message in (HL)
12128 2F60    Print <CR> as defined at 047E using SPEED
12137 2F69    Get a byte using IN
12150 2F76    Print prompt at 0479
12159 2F7F    Read command into input buffer
12185 2F99    Input buffer - set length - can safely increase to about 239 from 128
12196 2FA4    Read command - null re entry
12221 2FBD    Read command - <CR> re entry
12247 2FD7    Read command -check control character table
12273 2FF1    Read command - ^N re entry
12227 2FF5    Read command - ^O re entry
12283 2FFB    Read command - up/down arrow and HOME re entry
12295 3007    Read command - <BS> re entry
12310 3016    Read command - <right> re entry
12324 3024    Read command - <TAB> re entry
12356 3044    Read command - ^X re entry
12369 3051    Read command -control characters
12386 3062    Read command -control character jump table
12420 3084    Print character in 'A' to screen
12435 3093    Print <CR> to screen as defined in 047E
12451 30A3    Convert ASCII in (DE) to integer in HL, aborts if >FFFF
12528 30F0    Get address for line number
12618 314A    Insert line into table
12779 31EB    Delete line at HL
12959 329F    Print value in (HL) or DE
13038 32EE    Print token word in (DE)
13075 3313    Print token number from (DE)
13134 334E    Print number in A less than 10
13142 3356    Print number in (DE) less than 256
13153 3361    Print number in (DE) grater than 255
13164 336C    Print variable name from (DE)
13264 33D0    Print 'FN'
13273 33D9    Print REM or DATA line from (DE)
13285 33E5    Print string from (DE)
13312 3400    Print word from token line
13349 3425    Print Separator after comma (use 00 for none)
13357 342D    Print Separator after semicolon (use 00 for none)
13379 3443    Find word for token in A in primary word table
13389 344D    Find word for token in A in secondary word table
13413 3465    Print token line to ":"
13436 347C    Print token line to end of line
13459 3493    Print line number and line
13477 34A5    Print floating point in (DE)
13513 34C9    Make room for new string
13592 3518    Make new string
13659 355B    Check character type in (DE)
13766 35C6    Check primary token table
13774 35CE    Check secondary token table
13826 3602    Check both token tables
13833 3609    Parse immediate command
13992 36A8    Parse input line
14082 3702    Find variable in table
14218 378A    Make one new variable
14417 3851    Put new variable in temporary storage
14483 3893    Make new variables
14530 38C2    Fill token buffer
14549 38D5    Add C + DE
14555 38DB    Look for selected token , ; + - not ( )
14581 38F5    Look for token in A
14605 390D    Look for NOT + or -
14628 3924    Look for + or -
14651 393B    Parse line with DATA
14716 397C    Parse secondary tokens
14742 3996    Order of precedence for operators for PARSING
14777 39B9    Parse equation in token line
```

```
14875 3A1B    Parse ON(1) - look for equation
14875 3A1B    Parse CALL(1) - look for equation
14875 3A1B    Parse HTAB(1) - look for equation
14875 3A1B    Parse VTAB(1) - look for equation
14875 3A1B    Parse WAIT(1) - look for equation
14875 3A1B    Parse WAIT(3) - look for equation
14875 3A1B    Parse PLOT(1) - look for equation
14875 3A1B    Parse DRAW(1) - look for equation
14875 3A1B    Parse XDRAW(1) - look for equation
14875 3A1B    Parse HIMEM(1) - look for equation
14875 3A1B    Parse LOMEM(1) - look for equation
14875 3A1B    Parse HLIN(1) - look for equation
14875 3A1B    Parse VLIN(1) - look for equation
14875 3A1B    Parse HCOLOR(2) - look for equation
14875 3A1B    Parse PR(2) - look for equation
14875 3A1B    Parse IN(2) - look for equation
14875 3A1B    Parse PLOT(3) - look for equation
14875 3A1B    Parse HLIN(3) - look for equation
14875 3A1B    Parse VLIN(3) - look for equation
14875 3A1B    Parse HLIN(5) - look for equation
14875 3A1B    Parse VLIN(5) - look for equation
14947 3A63    Parse IF(1) - look for equation
14969 3A79    Parse WAIT(4) - look for comma followed by equation
14976 3A80    Parse DRAW(2) - look for AT x,y
14976 3A80    Parse XDRAW(2) - look for AT x,y
14991 3A8F    Parse FOR(1) - look for FP variable and '='
15020 3AAC    Parse LET(1) - look for variable '=' and equation/string
15035 3ABB    Parse IF(2) - look for THEN or GOTO
15093 3AF5    Parse FOR(3) - look for STEP and equation
15102 3AFE    Parse HPLOT(1) - look for "x,y" then TO and more "x,y" if any
15125 3B15    Parse DEF(1) - look for FN etc.
15209 3B69    Parse ON(2) - look for GOTO or GOSUB followed by line numbers
15232 3B80    Parse RUN(1) - look for line number if any
15243 3B8B    Parse LIST(1) - look for line range if any
15247 3B8F    Parse DEL(1) - lokk for line or range of lines
15307 3BCB    Check variable type FP INT STR or array
15364 3C04    Parse GET(1) - look for variable name
15364 3C04    Parse RECALL(1) - look for variable name
15364 3C04    Parse STORE(1) - look for variable name
15488 3C80    Move long variable name into memory (more than 2 letters)
15543 3CB7    Parse INPUT(1) - look for string prompt and/or variable
15567 3CCF    Parse NEXT(1) - look for variable name, comma, and more if any
15574 3CD6    Parse READ(1) - look for variable, comma, and more if any
15574 3CD6    Parse DATA(1) - look for variable, comma, and more if any
15574 3CD6    Parse DIM(1) - look for variable,comma, and mre if any
15574 3CD6    Parse Quoted string - look for variable, comma, and more if any
15580 3CDC    Parse PRINT(1) - look for equations, commas, semicolons, strings, etc
15653 3D25    Parse ASCII number into FP
15756 3D8C    Parse GOSUB(1) - look for line number
15756 3D8C    Parse GOTO(1) - look for line number
15756 3D8C    Parse ONERR(2) - look for line number
15814 3DC6    Parse &(1) - ignore rest of command
15817 3DC9    Parse REM(1) - ignore rest of command
15902 3E1E    Look for '=' token
15911 3E27    Parse HIMEM(1) - look for ':'
15911 3E27    Parse LOMEM(1) - look for ':'
15926 3E36    Parse ROT(1) - look for '='
15926 3E36    Parse SCALE(1) - look for '='
15926 3E36    Parse COLOR(1) - look for '='
15926 3E36    Parse HCOLOR(1) - look for '='
15939 3E43    Parse WAIT(2) - look for comma
15939 3E43    Parse PLOT(2) - look for comma
15939 3E43    Parse HLIN(2) - look for comma
15939 3E43    Parse VLIN(2) - look for comma
15950 3E4E    Parse IN(1) - look for '#'
15950 3E4E    Parse PR(1) - look for '#'
15963 3E5B    Parse FOR(2) - look for TO
15977 3E69    Parse HLIN(4) - look for AT
15977 3E69    Parse VLIN(4) - look for AT
15991 3E77    Parse ONERR(1) - look for GOTO
16029 3E9D    Cold Start routine
16035 3EA3    Main Processing Loop
16089 3ED9    Line number table
16089 3ED9    HIMEM setting
16091 3EDB    Number of program lines
16093 3EDD    Length of line number table
```

16095	3EDF	LOMEM
16095	3EDF	Start of variables
16097	3EE1	End of variables
16099	3EE3	End of variable commands
16099	3EE3	Start of numeric variables
16101	3EE5	Start of token buffer
16103	3EE7	End of token buffer
16105	3EE9	Temporary variables used while parsing line
16107	3EEB	Number of variables
16109	3EED	End of numeric variables
16111	3EEF	Start of string space
16115	3EF3	End of string space - remember strings allocated downward
16117	3EF5	DATA line number
16119	3EF7	DATA token buffer
16121	3EF9	DATA bytes remaining in line
16122	3EFA	CONTinue storage for DE
16124	3EFC	CONTinue storage for HL'
16126	3EFE	ONERR goto line number
16128	3F00	Error number
16129	3F01	SPEED setting
16130	3F02	USR Address
16132	3F04	& Address
16134	3F06	BREAK ASCII value ^C
16135	3F07	Pause ASCII value ^S
16136	3F08	Pause on flag - POKE 0 to force a pause in a program
16148	3F14	Indenting code (usually space)
16149	3F15	POKE limit
16151	3F17	Floating Point workspace
16162	3F22	Floating Point accumulator (1)
16171	3F2B	Floating Point accumulator (2)
16176	3F30	Printer width
16177	3F31	Printer Head Position
16178	3F32	Significant digits on output (default 9)
16190	3F3E	Random number seed values (4)
16195	3F43	IN vector storage
16197	3F45	TAPE/DISK read vector
16199	3F47	PR vector storage
16201	3F49	TAPE/DISK write vector
16203	3F4B	Screen write vector
16205	3F4D	Token buffer length
16207	3F4F	Line number for GOTO GOSUB
16210	3F52	Null string pointer (255,0,0)
16213	3F55	PR tables (8 vectors)
16229	3F65	IN tables (8 vectors)
16245	3F75	Input buffer max length - set by input routine at 12185
16246	3F76	Input buffer used length
16247	3F77	Input buffer up to 254 bytes; may be increased
16502	4076	Length of token buffer
16503	4077	Token buffer (up to 255 bytes)
16481	4061	Cold Boot set up routine----overwritten after boot
16641	4101	Boot device for HELLO file
16763	417B	HPLOT Last coordinates used (in case next is HPLOT TO x,y)
16765	417D	SCALE value
16766	417E	Shape Tbale address
16776	4188	COLOR value
16777	4189	HCOLOR value
16778	418A	Joystick 1 Stick
16779	418B	Joystick 1 RFB
16780	418C	Joystick 1 LFB
16781	418D	Joystick 1 Keypad
16782	418E	Joystick 1 Spinner
16783	418F	Joystick 2 Stick
16784	4190	Joystick 2 RFB
16785	4191	Joystick 2 LFB
16786	4192	Joystick 2 Keypad
16787	4193	Joystick 2 Spinner
16788	4194	Binary file ID
16789	4195	Binary file header (2 bytes)
16791	4197	Binary file load address
16793	4199	Random record length
16795	419B	Length of current file (for write)
16797	419D	File 1 name
16809	41A9	File 2 name
16821	41B5	Logged in device

```
16829 41BD    MON C vector
16831 41BF    MON I vector
16833 41C1    MON O vector
16835 41C3    MON L vector
16837 41C5    File 1 attributes
16838 41C6    File 1 file number
16839 41C7    File 1 FCB address
16841 41C9    File 1 length
16847 41CF    File 2 attributes
16848 41D0    File 2 file number
16849 41D1    File 2 FCB address
16851 41D3    File 2 length
16857 41D9    File 1 buffer (name and length)
16871 41E7    File 2 buffer (name and length)
16885 41F5    Directory file entry 1 - (full entry 26 bytes)
16911 420F    Directory file entry 2 - (full entry 26 bytes)
16953 4239    CURSOR ASCII value
16954 423A    BLANK SPACE ASCII value
16955 423B    CURSOR - character under
16956 423C    Left margin
16957 423D    Right Margin
16958 423E    Top cursor margin in all modes
16959 423F    Bottom Cursor margin
16960 4240    Screen line buffer (32 characters)
16993 4261    Scroll start line
16994 4262    Scroll up to column #
16995 4263    Top column for HOME--scroll region
16996 4264    Top line for HOME----scroll region
16997 4265    VRAM name table (current)
16999 4267    VRAM pattern table (current)
17001 4269    CURSOR line #
17002 426A    CURSOR column #
17003 426B    Last character register from keyboard or device
17005 426D    CURSOR flash flag
17006 426E    FLASH offset (128 default)
17008 4270    Mode 0=TEXT 1=GR 2=HGR 3=HGR2
17009 4271    Print to screen flag
17010 4272    FLASH flag
17011 4273    FLASH trigger - goes when 0
17013 4275    FLASH VRAM pages (8 and 24 toggled)
17014 4276    ^D flag
17015 4277    Character being printed
17016 4278    ^D buffer used length
17017 4279    ^D buffer
17040 4290    ^D buffer current position
17042 4292    File name pointer
17044 4294    File buffer pointer - $$$1 or $$$2
17046 4296    TEXT Screen initialization
17053 429D    TEXT border colour
17059 42A3    TEXT background colour
17115 42DB    TEXT character colour
17126 42E6    TEXT inverse colour
17164 430C    TEXT blank fill character (default 32)
17192 4327    TEXT FLASH frequency (default 12)
17226 434A    Character 31 TEXT appearance
17234 4352    Print characer in 'A' whicle checking graphics and flash
17275 437B    Keyboard read and flash cursor return in 'A'
17291 438B    CURSOR flash speed (default 4)
17310 439E    Keyboard read and put character in 426B
17334 43B6    Screen initialize B columns C lines HL name table A and a' pattern
17389 43ED    Print to scrren restore
17396 43F4    Print cursor
17416 4408    Print control characters
17448 4428    Print non-control characters
17581 44AD    Print character - <TAB> re-entry
17611 44CB    Print character - <CR> re-entry
17625 44D9    Print character - <LF> re-entry
17658 44FA    Print character - <UP> re-entry
17674 450A    Print character - <RIGHT> re-entry
17702 4526    Print character - <LEFT re-entry
17730 4542    Print character - ^N re-entry
17786 457A    Print character - ^O re-entry
17842 45B2    Print character - <FF> re-entry
17860 45C4    Print character - <HOME> re-entry
17869 45CD    Print character - ^V re-entry
17875 45D3    Print character - ^X re-entry
17905 45F1    Print character - ^\ re-entry
17945 4619    Print character - ^G re-entry
```

```
17954 4622    Bell frequency (default 17)
17958 4626    Bell volume 144->159 (default 144)
17963 462B    Bell duration (default 7)
17978 463A    Clear B characters
17982 463E    Clear a full line
17999 464F    Clear B lines from (HL)
18047 467F    Clear rest of line
18085 46A5    Read rest of line
18112 46C0    Scroll screen
18197 4715    Cursor update
18246 4746    Read character under cursor
18266 475A    Find name table for character at (HL)
18288 4770    Find pattern position for character in 'A'
18304 4780    Print character -control character table
18321 4791    Print character -control character jump table
18355 47B3    Cursor position to DE
18360 47B8    Find distance from left margin
18379 47CB    Print to end of screen ^P
18453 4815    TEXT Execution
18492 483C    GR Execution
18536 4868    GR number of rows-1
18537 4869    GR number of columns-1
18539 486B    GR HOME row (18539+18536 must be 23)
18540 486C    GR HOME column
18564 4884    GR blocks data
18606 48AE    GR border colour
18633 48C9    GR default colour (17*colour)
18711 4917    GR character colour
18728 4928    HCOLOR transaltion routine
18735 492F    COLOR translation routine
18755 4943    SCRN translation routine
18765 494D    HCOLOR translation table
18781 495D    COLOR translation table
18797 496D    HLIN graphics
18932 49F4    VLIN graphics
19094 4A96    PLOT graphics
19187 4AF3    Read screen colour at BC
19304 4B68    HOME Execution - print a <FF>
19348 4B94    Find pattern offset for BC coordinates
19448 4BF8    ^D echo if MON C
19471 4C0F    Print to screen or ^D buffer
19538 4C52    ^D Execution
19609 4C99    ^D Parse first character of I/O commands
19629 4CAD    ^D Parse rest of I/O command
19653 4CC5    ^D End of buffer i.e. exit
19667 4CD3    ^D Get execution address for I/O command
19693 4CED    CATALOG ^D vector
19701 4CF5    DELETE ^D vector
19712 4D00    RENAME ^D vector
19719 4D07    LOCK ^D vector
19726 4D0E    UNLOCK ^D vector
19733 4D15    BSAVE ^D vector
19740 4D1C    BLOAD ^D vector
19747 4D23    BRUN ^D vector
19754 4D2A    CLOSE ^D vector
19761 4D31    MON ^D vector
19768 4D38    NOMON ^D vector
19775 4D3F    LOAD ^D vector
19782 4D46    SAVE ^D vector
19789 4D4D    OPEN ^D vector
19796 4D54    APPEND ^D vector
19803 4D5B    WRITE ^D vector
19810 4D62    READ ^D vector
19817 4D69    POSITION ^D vector
19824 4D70    PR ^D vector
19831 4D77    IN ^D vector
19839 4D7F    FP ^D vector
19848 4D88    INT ^D vector
19857 4D91    INIT ^D vector
19866 4D9A    RUN ^D vector
19875 4DA3    RECOVER ^D vector
19884 4DAC    Parse first character of I/O commands
19914 4DCA    Parse rest of I/O command
19939 4DE3    Get Execution address for I/O command
19971 4E03    CATALOG immediate mode vector
19983 4E0F    DELETE immediate mode vector
20001 4E21    RENAME immediate mode vector
```

```
20008 4E28    LOCK immediate mode vector
20015 4E2F    UNLOCK immediate mode vector
20022 4E36    BSAVE immediate mode vector
20029 4E3D    BLOAD immediate mode vector
20036 4E44    BRUN immediate mode vector
20043 4E4B    CLOSE immediate mode vector
20050 4E52    MON immediate mode vector
20057 4E59    NOMON immediate mode vector
20064 4E60    LOAD immediate mode vector
20071 4E67    SAVE immediate mode vector
20078 4E6E    FP immediate mode vector
20087 4E77    INT immediate mode vector
20096 4E80    INIT immediate mode vector
20105 4E89    RUN immediate mode vector
20114 4E92    RECOVER immediate mode vector
20123 4E9B    I/O commands first letter
20138 4EAA    I/O commands word table and offset to jump table
20303 4F4F    I/O commands immediate mode jump table
20339 4F73    I/O commands ^D jump table
20416 4FC0    INT Execution
20419 4FC3    FP Execution
20469 4FF5    RENAME Execution
20532 5034    RECOVER Execution
20619 508B    RECOVER 'H' file fix...POKE ASC("H")
20640 50A0    UNLOCK Execution
20730 50FA    Look for ,Lnumber or ,Anumber (for BSAVE, BLOAD, BRUN)
20762 511A    Decode HEX number
20797 513D    Decode decimal number
20849 5171    BSAVE Execution
20993 5201    BLOAD Execution
21019 521B    BLOAD fix for BLOAD,Dn...POKE 11
21140 5294    BRUN Execution
21144 5298    CATALOG Execution
21298 5332    CATALOG ->POKE 0 to show usable blocks left
21331 5353    Print volume name
21352 5368    Print file name and data
21370 537A    CATALOG ->POKE 2=start block, 6=reserved length, 8= used length
21413 53A5    Decode a file name
21444 53C4    CATALOG data (Directory, Volume, Blocks free
21477 53E5    APPEND Execution
21715 54D3    POSITION Execution
21839 554F    Set write to device
21843 5553    Set read from device
21847 5557    MON I routine
21869 556D    MON O routine
21874 5572    NOMON O routine
21981 55DD    NOMON I Routine
22049 5621    READ I/O
22069 5635    READ sequential
22181 56A5    READ random
22432 57A0    Read BC bytes from file
22445 57AD    Read HL bytes from file
22455 57B7    WRITE Execution
22475 57CB    WRITE sequential
22923 598B    Write random
23042 5A02    NOMON Execution
23047 5A07    MON Execution
23140 5A64    MON NOMON Vectors
23156 5A74    MON NOMON Parameters (CIOL)
23175 5A87    MON C init
23182 5A8E    MON O init
23189 5A95    MON I init
23196 5A9C    MON L init
23203 5AA3    NOMON C init
23210 5AAA    NOMON O init
23217 5AB1    NOMON I init
23224 5AB8    NOMON L init
23231 5ABF    File name legal characters
23249 5AD1    File names $$$$1 and $$$$2
23263 5ADF    Device names SVD
23273 5AE9    Device translation table
23279 5AEF    File name - legal character check
23304 5B08    File name - look for comma
23315 5B13    File name - look for second file
23324 5B1C    File name - look for first name
```

```
23364 5B44    Parse drive name
23489 5BC1    File type change - file 1 or 2
23504 5BD0    Find available temporary file
23635 5C53    Make backup file (A H to a h)
23769 5CD9    Read one byte from file
23796 5CF4    Write one byte to file
23813 5D05    SAVE Execution
23925 5D75    File type for SAVE ->POKE 2 to make system file, or 65 for 'A'
23976 5DA8    LOAD Execution
24012 5DCC    RUN Execution
24099 5E23    Write program to medium
24126 5E3E    I/O error messages
24297 5EE9    I/O error handler
24419 5F63    Close files
24497 5FB1    OPEN Execution
24612 6024    CLOSE Execution
24934 6166    PR Execution
24923 616F    IN Execution
24980 6194    Set file data buffer
25087 61FF    Close file buffer
25151 623F    Find file data buffer
25207 6277    Find Program length
25257 62A9    System file data ->BASICPGM
25267 62B3    INIT Execution
25305 62D9    INIT medium size
25308 62DC    INIT directory size
25354 630A    Set system date
25355 630B    Set system date -- Day byte
25356 630C    Set system date -- Month byte
25358 630E    Set system date -- Year byte
25362 6312    INIT Block 0 data
25370 631A    HGR2 Execution
25412 6344    HGR2 initialization data
25431 6357    HGR2 border colour
25471 637F    HGR2 default screen colour (17*colour)
25484 638C    HGR Execution
25568 63E0    HGR character colour
25573 63E5    HGR number of lines-1
25574 63E6    HGR number of columns-1
25576 63E8    HGR HOME line (25576+25573 must equal 23)
25577 63E9    HGR HOME column
25601 6401    HPLOT Graphics
25686 6456    HPLOT xy TO xy Execution
25797 64C5    HPLOT TO xy Execution
25812 64D4    HPLOT a line
26141 661D    INVERSE Execution
26151 6627    NORMAL Execution
26163 6633    FLASH Execution
26177 6641    POS calculation
26184 6648    VPOS calculation
26191 664F    HTAB cursor adjustment
26219 666B    VTAB cursor adjustment
26297 66B9    XDRAW with no xy
26317 66CD    DRAW with no xy
26333 66DD    SCALE Execution
26344 66E8    ROT graphics
26574 67CE    Shape table default shape
26588 67DC    DRAW at xy
26884 6904    XDRAW at xy
26904 6918    PDL decode
26948 6944    PDL(6)
26959 694F    PDL(7)
26965 6955    PDL(8)
26971 695B    PDL(9)
26977 6961    PDL(10)
27006 697E    PDL(11)
27012 6984    PDL(12)
27018 698A    PDL(13)
27024 6990    PDL(4)
27029 6995    PDL(5)
27034 699A    PDL(14)
27039 699F    PDL(15)
27044 69A4    PDL(0)
27053 69AD    PDL(1)
27058 69B2    PDL(2)
27063 69B7    PDL(3)
```

```
27068 69BC    PDL jump table
27100 69DC    Joystick 1 - N-S value
27101 69DD    Joystick 1 - E-W value
27102 69DE    Joystick 2 - N-S value
27103 69DF    Joystick 2 - E-W value
27104 69E0    PDL interpretation
27130 69FA    Read Joystick 1
27148 6A0C    Read Joystick 2
27162 6A1A    Joystick 1 NORTH decode
27171 6A23    Joystick 1 SOUTH decode
27180 6A2C    Joystick 1 EAST  decode
27189 6A35    Joystick 1 WEST  decode
27198 6A3E    Joystick 1 N-E   decode
27211 6A4B    Joystick 1 N-W   decode
27224 6A58    Joystick 1 S-E   decode
27237 6A65    Joystick 1 S-W   decode
27250 6A72    Joystick 2 NORTH decode
27259 6A7B    Joystick 2 SOUTH decode
27268 6A84    Joystick 2 EAST  decode
27277 6A8D    Joystick 2 WEST  decode
27286 6A96    Joystick 2 N-E   decode
27299 6AA3    Joystick 2 N-W   decode
27312 6AB0    Joystick 2 S-E   decode
27325 6ABD    Joystick 2 S-W   decode
27388 6ACA    PDL increment
27347 6AD3    PDL decrement
27355 6ADB    Joystick 1 execution table
27381 6AF5    Joystick 2 execution table
```

```
16132 3F04    & Address
10164 27B4    & Execution

04479 117F    ABS of FPA1 i.e. make it positive
02276 08E4    ABS() Execution - reset sign of FPA1
05417 1529    ADD (+) Execution
05496 1578    AND Execution
21477 53E5    APPEND Execution
19796 4D54    APPEND ^D vector
10351 286F    ASC() Execution
04180 1054    ATN() Execution
01569 0621    Add (HL) to FPA1
14549 38D5    Add C + DE
01583 062F    Add FPA1 and FPA2

16954 423A    BLANK SPACE ASCII value
20993 5201    BLOAD Execution
19740 4D1C    BLOAD ^D vector
21019 521B    BLOAD fix for BLOAD,Dn...POKE 11
20029 4E3D    BLOAD immediate mode vector
16134 3F06    BREAK ASCII value ^C
06346 18CA    BREAK Execution
21140 5294    BRUN Execution
19747 4D23    BRUN ^D vector
20036 4E44    BRUN immediate mode vector
20849 5171    BSAVE Execution
19733 4D15    BSAVE ^D vector
20022 4E36    BSAVE immediate mode vector
17963 462B    Bell duration (default 7)
17954 4622    Bell frequency (default 17)
17958 4626    Bell volume 144->159 (default 144)
16788 4194    Binary file ID
16789 4195    Binary file header (2 bytes)
16791 4197    Binary file load address
16641 4101    Boot device for HELLO file
16959 423F    Bottom Cursor margin

10042 273A    CALL Execution
21298 5332    CATALOG ->POKE 0 to show usable blocks left
21370 537A    CATALOG ->POKE 2=start block, 6=reserved length, 8= used length
21144 5298    CATALOG Execution
19693 4CED    CATALOG ^D vector
21444 53C4    CATALOG data (Directory, Volume, Blocks free
19971 4E03    CATALOG immediate mode vector
10371 2883    CHR$() Execution
08141 1FCD    CLEAR Execution
24612 6024    CLOSE Execution
19754 4D2A    CLOSE ^D vector
20043 4E4B    CLOSE immediate mode vector
08109 1FAD    CLRERR Execution
11099 2B5B    COLOR Execution
18735 492F    COLOR translation routine
18781 495D    COLOR translation table
16776 4188    COLOR value
06387 18F3    CONT Execution
16122 3EFA    CONTinue storage for DE
16124 3EFC    CONTinue storage for HL'
03946 0F6A    COS() Execution - SIN(x+pi/2)
16955 423B    CURSOR - character under
16953 4239    CURSOR ASCII value
17002 426A    CURSOR column #
17005 426D    CURSOR flash flag
17291 438B    CURSOR flash speed (default 4)
17001 4269    CURSOR line #
17226 434A    Character 31 TEXT appearance
17015 4277    Character being printed
11864 2E58    Check (DE) for letter
11884 2E6C    Check (DE) for number
13826 3602    Check both token tables
13659 355B    Check character type in (DE)
07173 1C05    Check length of DATA or DIM
13766 35C6    Check primary token table
13774 35CE    Check secondary token table
04929 1341    Check token of incoming number to decide return address
06555 199B    Check variable type
15307 3BCB    Check variable type FP INT STR or array
```

```
17978 463A    Clear B characters
17999 464F    Clear B lines from (HL)
17982 463E    Clear a full line
18047 467F    Clear rest of line
25087 61FF    Close file buffer
24419 5F63    Close files
16481 4061    Cold Boot set up routine----overwritten after boot
00256 0100    Cold Start address
16029 3E9D    Cold Start routine
01083 0436    Coleco SmartBasic message
06423 1917    Command Execution jump table by token
06112 17E0    Command Execution routine
00272 0110    Command Parse Vector pointers
00938 03AA    Command Parsing Vectors
01148 047C    Command Separator (the COLON) - count, character
02488 09B8    Compare FPA1 and FPA2 carry set if FPA1 bigger
01851 073B    Compare signs of FPA1 and FPA2
11786 2E0A    Compare strings at (HL) and (DE)
11851 2E4B    Compare strings of known length
11838 2E3E    Compare strings of unknown length
05636 1604    Compare strings pointed to by FPA1 and FPA2
04497 1191    Constants FP and INT used by power series calculators
12451 30A3    Convert ASCII in (DE) to integer in HL, aborts if >FFFF
18355 47B3    Cursor position to DE
18197 4715    Cursor update

08419 20E3    DATA Execution
16121 3EF9    DATA bytes remaining in line
16117 3EF5    DATA line number
16119 3EF7    DATA token buffer
08244 2034    DEF Execution
07555 1D83    DEL Execution
19701 4CF5    DELETE ^D vector
19983 4E0F    DELETE immediate mode vector
06942 1B1E    DIM Execution
06729 1A49    DIM for Floating Point
07097 1BB9    DIM for strings
05482 156A    DIVIDE (/) Execution
11351 2C57    DRAW Execution
26588 67DC    DRAW at xy
26317 66CD    DRAW with no xy
20762 511A    Decode HEX number
21413 53A5    Decode a file name
20797 513D    Decode decimal number
07264 1C60    Define string
12779 31EB    Delete line at HL
23263 5ADF    Device names SVD
23273 5AE9    Device translation table
16885 41F5    Directory file entry 1 - (full entry 26 bytes)
16911 420F    Directory file entry 2 - (full entry 26 bytes)
03580 0DFC    Divide FPA1 by 10 only if greater than 10
02018 07E2    Divide FPA1 by FPA2
02140 085C    Divide FPA2 by 2

06047 179F    END Execution
05617 15F1    EQUAL (=) Execution
10814 2A3E    ERRNUM() Execution
03816 0EE8    EXP() Execution
05489 1571    EXPONENT (^) Execution
16109 3EED    End of numeric variables
16115 3EF3    End of string space - remember strings allocated downward
16103 3EE7    End of token buffer
16099 3EE3    End of variable commands
16097 3EE1    End of variables
01463 05B7    Errnum Offsets to error messages
07900 1EDC    Error handler for runtime errors
01152 0480    Error messages - Length, message
16128 3F00    Error number
05885 16FD    Evaluate equation
03696 0E70    Exponent FPA1^FPA2
```

```
26163 6633    FLASH Execution
11050 2B2A    FLASH Execution vector
17013 4275    FLASH VRAM pages (8 and 24 toggled)
17010 4272    FLASH flag
17006 426E    FLASH offset (128 default)
00159 009F    FLASH reset value
17011 4273    FLASH trigger - goes when 0
08557 216D    FOR Execution
20419 4FC3    FP Execution
19839 4D7F    FP ^D vector
20078 4E6E    FP immediate mode vector
10192 27D0    FRE() Execution
01064 0428    Fatal System Error
16839 41C7    File 1 FCB address
16837 41C5    File 1 attributes
16857 41D9    File 1 buffer (name and length)
16838 41C6    File 1 file number
16841 41C9    File 1 length
16797 419D    File 1 name
16849 41D1    File 2 FCB address
16847 41CF    File 2 attributes
16871 41E7    File 2 buffer (name and length)
16848 41D0    File 2 file number
16851 41D3    File 2 length
16809 41A9    File 2 name
17044 4294    File buffer pointer - $$$1 or $$$2
23279 5AEF    File name - legal character check
23304 5B08    File name - look for comma
23324 5B1C    File name - look for first name
23315 5B13    File name - look for second file
23231 5ABF    File name legal characters
17042 4292    File name pointer
23249 5AD1    File names $$$$1 and $$$$2
23489 5BC1    File type change - file 1 or 2
23925 5D75    File type for SAVE ->POKE 2 to make system file, or 65 for 'A'
14530 38C2    Fill token buffer
25207 6277    Find Program length
23504 5BD0    Find available temporary file
18360 47B8    Find distance from left margin
25151 623F    Find file data buffer
18266 475A    Find name table for character at (HL)
05760 1680    Find operator in token line
19348 4B94    Find pattern offset for BC coordinates
18288 4770    Find pattern position for character in 'A'
14082 3702    Find variable in table
13379 3443    Find word for token in A in primary word table
13389 344D    Find word for token in A in secondary word table
16162 3F22    Floating Point accumulator (1)
16171 3F2B    Floating Point accumulator (2)
16151 3F17    Floating Point workspace
02875 0B3B    Floating point number from 1E-38 to 1E38

09378 24A2    GET Execution
09395 24B3    GET number byte
09464 24F8    GET string byte
08427 20EB    GOSUB Execution
08342 2096    GOTO Execution
18492 483C    GR Execution
11070 2B3E    GR Execution vector
18540 486C    GR HOME column
18539 486B    GR HOME row (18539+18536 must be 23)
18564 4884    GR blocks data
18606 48AE    GR border colour
18711 4917    GR character colour
18633 48C9    GR default colour (17*colour)
18537 4869    GR number of columns-1
18536 4868    GR number of rows-1
05573 15C5    GREATER EQUAL (>=) Execution
05541 15A5    GREATER THAN (>) Execution
```

```
19939 4DE3    Get Execution address for I/O command
12137 2F69    Get a byte using IN
05837 16CD    Get a number (int, FP, STRING, etc) from token line to FPA1
12528 30F0    Get address for line number
05939 1733    Get equation from token line
05008 1390    Get integer >255 from token line to FPA1 or FPA2
04982 1376    Get integer from 0-9 from token line to FPA1 or FPA2
04995 1383    Get integer from 10-255 from token line to FPA1 or FPA2
09987 2703    Get integer from token line
05994 176A    Get line number for RUN
06016 1780    Get next line number
06190 182E    Get next program line
11776 2E00    Get non-control character from (DE)
05861 16E5    Get number from token line to FPA2

11119 2B6F    HCOLOR Execution
18728 4928    HCOLOR transaltion routine
18765 494D    HCOLOR translation table
16777 4189    HCOLOR value
25484 638C    HGR Execution
11075 2B43    HGR Execution vector
25577 63E9    HGR HOME column
25576 63E8    HGR HOME line (25576+25573 must equal 23)
25568 63E0    HGR character colour
25574 63E6    HGR number of columns-1
25573 63E5    HGR number of lines-1
25370 631A    HGR2 Execution
11080 2B48    HGR2 Execution vector
25431 6357    HGR2 border colour
25471 637F    HGR2 default screen colour (17*colour)
25412 6344    HGR2 initialization data
11010 2B02    HIMEM Execution
16089 3ED9    HIMEM setting
11170 2BA2    HLIN Execution
18797 496D    HLIN graphics
19304 4B68    HOME Execution - print a <FF>
11090 2B52    HOME Execution vector
11487 2CDF    HPLOT Execution
25601 6401    HPLOT Graphics
16763 417B    HPLOT Last coordinates used ( in case next is HPLOT TO x,y)
25797 64C5    HPLOT TO xy Execution
25812 64D4    HPLOT a line
25686 6456    HPLOT xy TO xy Execution
11320 2C38    HTAB Execution
26191 664F    HTAB cursor adjustment
01055 041F    Hi Cathy

20339 4F73    I/O commands ^D jump table
20123 4E9B    I/O commands first letter
20303 4F4F    I/O commands immediate mode jump table
20138 4EAA    I/O commands word table and offset to jump table
24297 5EE9    I/O error handler
24126 5E3E    I/O error messages
07705 1E19    IF Execution
24923 616F    IN Execution
19831 4D77    IN ^D vector
12084 2F34    IN initialization
16229 3F65    IN tables (8 vectors)
16195 3F43    IN vector storage
25362 6312    INIT Block 0 data
25267 62B3    INIT Execution
19857 4D91    INIT ^D vector
25308 62DC    INIT directory size
20096 4E80    INIT immediate mode vector
25305 62D9    INIT medium size
08957 22FD    INPUT Execution
09068 236C    INPUT floating Point
09336 2478    INPUT integer
09003 232B    INPUT prompt character
09229 240D    INPUT string
20416 4FC0    INT Execution
19848 4D88    INT ^D vector
20087 4E77    INT immediate mode vector
10672 29B0    INT() Execution
26141 661D    INVERSE Execution
11055 2B2F    INVERSE Execution vector
```

```
16148 3F14    Indenting code (usually space)
12185 2F99    Input buffer - set length - can safely increase to about 239 from 128
16245 3F75    Input buffer max length - set by input routine at 12185
16247 3F77    Input buffer up to 254 bytes; may be increased
16246 3F76    Input buffer used length
12618 314A    Insert line into table
01485 05CD    Integers 1 through 15??
00262 0106    Integers 10000,1000,100,10,1
03900 0F3C    Invert FPA1 (1/FPA1)

27101 69DD    Joystick 1 - E-W value
27100 69DC    Joystick 1 - N-S value
27180 6A2C    Joystick 1 EAST  decode
16781 418D    Joystick 1 Keypad
16780 418C    Joystick 1 LFB
27198 6A3E    Joystick 1 N-E   decode
27211 6A4B    Joystick 1 N-W   decode
27162 6A1A    Joystick 1 NORTH decode
16779 418B    Joystick 1 RFB
27224 6A58    Joystick 1 S-E   decode
27237 6A65    Joystick 1 S-W   decode
27171 6A23    Joystick 1 SOUTH decode
16782 418E    Joystick 1 Spinner
16778 418A    Joystick 1 Stick
27189 6A35    Joystick 1 WEST  decode
27355 6ADB    Joystick 1 execution table
27103 69DF    Joystick 2 - E-W value
27102 69DE    Joystick 2 - N-S value
27268 6A84    Joystick 2 EAST  decode
16786 4192    Joystick 2 Keypad
16785 4191    Joystick 2 LFB
27286 6A96    Joystick 2 N-E   decode
27299 6AA3    Joystick 2 N-W   decode
27250 6A72    Joystick 2 NORTH decode
16784 4190    Joystick 2 RFB
27312 6AB0    Joystick 2 S-E   decode
27325 6ABD    Joystick 2 S-W   decode
27259 6A7B    Joystick 2 SOUTH decode
16787 4193    Joystick 2 Spinner
16783 418F    Joystick 2 Stick
27277 6A8D    Joystick 2 WEST  decode
27381 6AF5    Joystick 2 execution table

17275 437B    Keyboard read and flash cursor return in 'A'
17310 439E    Keyboard read and put character in 426B

10508 290C    LEFT$() Execution
10454 28D6    LEN() Execution
05557 15B5    LESS EQUAL (<=) Execution
05527 1597    LESS THAN (<) Execution
06247 1867    LET Execution
07407 1CEF    LIST Execution
07357 1CBD    LIST a range of lines
23976 5DA8    LOAD Execution
19775 4D3F    LOAD ^D vector
20064 4E60    LOAD immediate mode vector
19719 4D07    LOCK ^D vector
20008 4E28    LOCK immediate mode vector
03604 0E14    LOG() Execution
16095 3EDF    LOMEM
10870 2A76    LOMEM Execution
17003 426B    Last character register from keyboard or device
16956 423C    Left margin
16795 419B    Length of current file (for write)
16093 3EDD    Length of line number table
16502 4076    Length of token buffer
16207 3F4F    Line number for GOTO GOSUB
16089 3ED9    Line number table
01519 05EF    Load BC with number for GOTO
02407 0967    Load FPA1 or FPA2 (depends on carry) with HL (integer)
04348 1120    Load FPA1 with 1
02576 0A10    Load FPA1 with ASCII number from (DE)
02354 0932    Load HL with INT(FPA1)
01500 05DC    Load HL with a line number
05023 139F    Load value from token line to FPA1
05041 13B1    Load value from token line to FPA2
```

```
16821 41B5    Logged in device
15902 3E1E    Look for '=' token
14628 3924    Look for + or -
20730 50FA    Look for ,Lnumber or ,Anumber (for BSAVE, BLOAD, BRUN)
14605 390D    Look for NOT + or -
14555 38DB    Look for selected token , ; + - not ( )
14581 38F5    Look for token in A

10553 2939    MID$() Execution
23175 5A87    MON C init
16829 41BD    MON C vector
23047 5A07    MON Execution
23189 5A95    MON I init
21847 5557    MON I routine
16831 41BF    MON I vector
23196 5A9C    MON L init
16835 41C3    MON L vector
23156 5A74    MON NOMON Parameters (CIOL)
23140 5A64    MON NOMON Vectors
23182 5A8E    MON O init
21869 556D    MON O routine
16833 41C1    MON O vector
19761 4D31    MON ^D vector
20050 4E52    MON immediate mode vector
05475 1563    MULTIPLY (*) Execution
16035 3EA3    Main Processing Loop
23635 5C53    Make backup file (A H to a h)
13592 3518    Make new string
14483 3893    Make new variables
14218 378A    Make one new variable
13513 34C9    Make room for new string
17008 4270    Mode 0=TEXT 1=GR 2=HGR 3=HGR2
04357 1117    Move (HL) to FPA2
05059 13C3    Move FP number from (BC) to (HL)
04356 1104    Move FPA1 to FPA2
04869 1305    Move FPA1 to FPA2 while checking for string pointer
01832 0728    Move FPA2 to FPA1
04899 1323    Move FPA2 to FPA1 while checking for string pointer
05078 13D6    Move integer from (BC) to (HL)
15488 3C80    Move long variable name into memory (more than 2 letters)
05374 14FE    Move number from token line (DE) to FPA1 or FPA2
05407 151F    Move string from (HL) to (DE)
05095 13E7    Move string number from (BC) to (HL)
05167 142F    Move variable to FPA1
05135 140F    Move variable value from token line to FPA1
05141 1415    Move variable value from token line to FPA2
03420 0D5C    Multiply DEBC by 10
02708 0A94    Multiply DEBC by 2
01885 075D    Multiply FPA1 by (HL)
01875 0753    Multiply FPA1 by 2
01888 0760    Multiply FPA1 by FPA2
07066 1B9A    Multiply HL by DE

06356 18D4    NEW Execution - POKE 201 to enable MERGE, POKE 205 to restore
11892 2E74    NEW initialization
08749 222D    NEXT Execution
06351 18CF    NOBREAK Execution
23203 5AA3    NOMON C init
23042 5A02    NOMON Execution
21981 55DD    NOMON I Routine
23217 5AB1    NOMON I init
23224 5AB8    NOMON L init
23210 5AAA    NOMON O init
21874 5572    NOMON O routine
19768 4D38    NOMON ^D vector
20057 4E59    NOMON immediate mode vector
26151 6627    NORMAL Execution
11060 2B34    NORMAL Execution vector
05587 15D3    NOT EQUAL (<>) Execution
05654 1616    NOT Execution
06341 18C5    NOTRACE Execution
00102 0066    Non Maskable Interrupt used for FLASH
16210 3F52    Null string pointer (255,0,0)
16091 3EDB    Number of program lines
16107 3EEB    Number of variables
```

```
08381 20BD    ON Execution
08516 2144    ON GOSUB Execution
08144 1FB2    ONERR Execution
16126 3EFE    ONERR goto line number
24497 5FB1    OPEN Execution
19789 4D4D    OPEN ^D vector
05513 1589    OR Execution
00818 0332    Operator Token table
05797 16A5    Operator order of precedence and jump table for execute
14742 3996    Order of precedence for operators for PARSING

26904 6918    PDL decode
27347 6AD3    PDL decrement
27388 6ACA    PDL increment
27104 69E0    PDL interpretation
27068 69BC    PDL jump table
11619 2D63    PDL() Execution
27044 69A4    PDL(0)
27053 69AD    PDL(1)
26977 6961    PDL(10)
27006 697E    PDL(11)
27012 6984    PDL(12)
27018 698A    PDL(13)
27034 699A    PDL(14)
27039 699F    PDL(15)
27058 69B2    PDL(2)
27063 69B7    PDL(3)
27024 6990    PDL(4)
27029 6995    PDL(5)
26948 6944    PDL(6)
26959 694F    PDL(7)
26965 6955    PDL(8)
26971 695B    PDL(9)
10091 276B    PEEK Execution
11139 2B83    PLOT Execution
19094 4A96    PLOT graphics
10104 2778    POKE Execution
16149 3F15    POKE limit
08493 212D    POP Execution
26177 6641    POS calculation
10844 2A5C    POS() Execution
21715 54D3    POSITION Execution
19817 4D69    POSITION ^D vector
24934 6166    PR Execution
19824 4D70    PR ^D vector
12058 2F1A    PR initialization
16213 3F55    PR tables (8 vectors)
16199 3F47    PR vector storage
07740 1E3C    PRINT Execution
15814 3DC6    Parse &(1) - ignore rest of command
15653 3D25    Parse ASCII number into FP
14875 3A1B    Parse CALL(1) - look for equation
15926 3E36    Parse COLOR(1) - look for '='
15574 3CD6    Parse DATA(1) - look for variable, comma, and more if any
15125 3B15    Parse DEF(1) - look for FN etc.
15247 3B8F    Parse DEL(1) - lokk for line or range of lines
15574 3CD6    Parse DIM(1) - look for variable,comma, and mre if any
14875 3A1B    Parse DRAW(1) - look for equation
14976 3A80    Parse DRAW(2) - look for AT x,y
14991 3A8F    Parse FOR(1) - look for FP variable and '='
15963 3E5B    Parse FOR(2) - look for TO
15093 3AF5    Parse FOR(3) - look for STEP and equation
15364 3C04    Parse GET(1) - look for variable name
15756 3D8C    Parse GOSUB(1) - look for line number
15756 3D8C    Parse GOTO(1) - look for line number
15926 3E36    Parse HCOLOR(1) - look for '='
14875 3A1B    Parse HCOLOR(2) - look for equation
15911 3E27    Parse HIMEM(1) - look for ':'
14875 3A1B    Parse HIMEM(1) - look for equation
14875 3A1B    Parse HLIN(1) - look for equation
15939 3E43    Parse HLIN(2) - look for comma
14875 3A1B    Parse HLIN(3) - look for equation
15977 3E69    Parse HLIN(4) - look for AT
14875 3A1B    Parse HLIN(5) - look for equation
15102 3AFE    Parse HPLOT(1) - look for "x,y" then TO and more "x,y" if any
14875 3A1B    Parse HTAB(1) - look for equation
```

```
14947 3A63    Parse IF(1) - look for equation
15035 3ABB    Parse IF(2) - look for THEN or GOTO
15950 3E4E    Parse IN(1) - look for '#'
14875 3A1B    Parse IN(2) - look for equation
15543 3CB7    Parse INPUT(1) - look for string prompt and/or variable
15020 3AAC    Parse LET(1) - look for variable '=' and equation/string
15243 3B8B    Parse LIST(1) - look for line range if any
15911 3E27    Parse LOMEM(1) - look for ':'
14875 3A1B    Parse LOMEM(1) - look for equation
15567 3CCF    Parse NEXT(1) - look for variable name, comma, and more if any
14875 3A1B    Parse ON(1) - look for equation
15209 3B69    Parse ON(2) - look for GOTO or GOSUB followed by line numbers
15991 3E77    Parse ONERR(1) - look for GOTO
15756 3D8C    Parse ONERR(2) - look for line number
14875 3A1B    Parse PLOT(1) - look for equation
15939 3E43    Parse PLOT(2) - look for comma
14875 3A1B    Parse PLOT(3) - look for equation
15950 3E4E    Parse PR(1) - look for '#'
14875 3A1B    Parse PR(2) - look for equation
15580 3CDC    Parse PRINT(1) - look for equations, commas, semicolons, strings, etc
15574 3CD6    Parse Quoted string - look for variable, comma, and more if any
15574 3CD6    Parse READ(1) - look for variable, comma, and more if any
15364 3C04    Parse RECALL(1) - look for variable name
15817 3DC9    Parse REM(1) - ignore rest of command
15926 3E36    Parse ROT(1) - look for '='
15232 3B80    Parse RUN(1) - look for line number if any
15926 3E36    Parse SCALE(1) - look for '='
15364 3C04    Parse STORE(1) - look for variable name
14875 3A1B    Parse VLIN(1) - look for equation
15939 3E43    Parse VLIN(2) - look for comma
14875 3A1B    Parse VLIN(3) - look for equation
15977 3E69    Parse VLIN(4) - look for AT
14875 3A1B    Parse VLIN(5) - look for equation
14875 3A1B    Parse VTAB(1) - look for equation
14875 3A1B    Parse WAIT(1) - look for equation
15939 3E43    Parse WAIT(2) - look for comma
14875 3A1B    Parse WAIT(3) - look for equation
14969 3A79    Parse WAIT(4) - look for comma followed by equation
14875 3A1B    Parse XDRAW(1) - look for equation
14976 3A80    Parse XDRAW(2) - look for AT x,y
23364 5B44    Parse drive name
14777 39B9    Parse equation in token line
19884 4DAC    Parse first character of I/O commands
13833 3609    Parse immediate command
13992 36A8    Parse input line
14651 393B    Parse line with DATA
19914 4DCA    Parse rest of I/O command
14716 397C    Parse secondary tokens
16135 3F07    Pause ASCII value ^S
16136 3F08    Pause on flag - POKE 0 to force a pause in a program
04411 113B    Pop FPA1 from Stack
04835 12E3    Pop FPA1 from Stack while checking for string pointer
04441 1159    Pop FPA2 from Stack
04156 103C    Power series calculator for 2 squares algorithms
04270 10AE    Power series calculator for multiplication algorithms
13264 33D0    Print 'FN'
12128 2F60    Print <CR> as defined at 047E using SPEED
12435 3093    Print <CR> to screen as defined in 047E
13273 33D9    Print REM or DATA line from (DE)
13349 3425    Print Separator after comma (use 00 for none)
13357 342D    Print Separator after semicolon (use 00 for none)
17234 4352    Print characer in 'A' whicle checking graphics and flash
17611 44CB    Print character - <CR> re-entry
17842 45B2    Print character - <FF> re-entry
17860 45C4    Print character - <HOME> re-entry
17702 4526    Print character - <LEFT re-entry
17625 44D9    Print character - <LF> re-entry
17674 450A    Print character - <RIGHT> re-entry
17581 44AD    Print character - <TAB> re-entry
17658 44FA    Print character - <UP> re-entry
17945 4619    Print character - ^G re-entry
17730 4542    Print character - ^N re-entry
17786 457A    Print character - ^O re-entry
17869 45CD    Print character - ^V re-entry
17875 45D3    Print character - ^X re-entry
17905 45F1    Print character - ^\ re-entry
```

```
18321 4791    Print character -control character jump table
18304 4780    Print character -control character table
11994 2EDA    Print character in 'A' to current PR routine
12420 3084    Print character in 'A' to screen
12010 2EEA    Print character in 'A' to screen and printer using SPEED
12043 2F0B    Print character in 'A' to screen only using SPEED
17416 4408    Print control characters
17396 43F4    Print cursor
21352 5368    Print file name and data
01553 0611    Print floating Point Number
13477 34A5    Print floating point in (DE)
12110 2F4E    Print length encoded message in (HL)
13459 3493    Print line number and line
17448 4428    Print non-control characters
13153 3361    Print number in (DE) grater than 255
13142 3356    Print number in (DE) less than 256
13134 334E    Print number in A less than 10
11920 2E90    Print parse errors
12150 2F76    Print prompt at 0479
13285 33E5    Print string from (DE)
18379 47CB    Print to end of screen ^P
17009 4271    Print to screen flag
19471 4C0F    Print to screen or ^D buffer
17389 43ED    Print to scrren restore
13413 3465    Print token line to ":"
13436 347C    Print token line to end of line
13075 3313    Print token number from (DE)
13038 32EE    Print token word in (DE)
12959 329F    Print value in (HL) or DE
13164 336C    Print variable name from (DE)
21331 5353    Print volume name
13312 3400    Print word from token line
16177 3F31    Printer Head Position
16176 3F30    Printer width
01145 0479    Prompt pointer - Count, prompt1, prompt2
04396 112C    Push FPA1 on Stack
04789 12B5    Push FPA1 to Stack while checking for string pointer
04426 114A    Push FPA2 on Stack
14417 3851    Put new variable in temporary storage

09499 251B    READ Execution (read data)
22049 5621    READ I/O
19810 4D62    READ ^D vector
09778 2632    READ floating point
09591 2577    READ integer
09705 25E9    READ quoted string
22181 56A5    READ random
22069 5635    READ sequential
09602 2582    READ string
11764 2DF4    RECALL Execution
20619 508B    RECOVER 'H' file fix...POKE ASC("H")
20532 5034    RECOVER Execution
19875 4DA3    RECOVER ^D vector
20114 4E92    RECOVER immediate mode vector
08419 20E3    REM Execution
20469 4FF5    RENAME Execution
19712 4D00    RENAME ^D vector
20001 4E21    RENAME immediate mode vector
09482 250A    RESTORE Execution
08313 2079    RESUME Execution
08477 211D    RETURN Execution
10529 2921    RIGHT$() Execution
04696 1258    RND() Execution
11459 2CC3    ROT Execution
26344 66E8    ROT graphics
06159 180F    RUN Execution
24012 5DCC    RUN Execution
19866 4D9A    RUN ^D vector
20105 4E89    RUN immediate mode vector
16190 3F3E    Random number seed values (4)
16793 4199    Random record length
```

```
22432 57A0    Read BC bytes from file
22445 57AD    Read HL bytes from file
27130 69FA    Read Joystick 1
27148 6A0C    Read Joystick 2
18246 4746    Read character under cursor
12295 3007    Read command - <BS> re entry
12221 2FBD    Read command - <CR> re entry
12324 3024    Read command - <TAB> re entry
12310 3016    Read command - <right> re entry
12273 2FF1    Read command - ^N re entry
12227 2FF5    Read command - ^O re entry
12356 3044    Read command - ^X re entry
12196 2FA4    Read command - null re entry
12283 2FFB    Read command - up/down arrow and HOME re entry
12247 2FD7    Read command -check control character table
12386 3062    Read command -control character jump table
12369 3051    Read command -control characters
12159 2F7F    Read command into input buffer
23769 5CD9    Read one byte from file
18085 46A5    Read rest of line
19187 4AF3    Read screen colour at BC
01150 047E    Return pointer - count and character <CR>
16957 423D    Right Margin

23813 5D05    SAVE Execution
19782 4D46    SAVE ^D vector
20071 4E67    SAVE immediate mode vector
26333 66DD    SCALE Execution
11473 2CD1    SCALE Execution
16765 417D    SCALE value
18755 4943    SCRN translation routine
11268 2C04    SCRN() Execution
02285 08ED    SGN() Execution - get sign of FPA1
11085 2B4D    SHLOAD Execution vector (points to a return)
03954 0F72    SIN() Execution
10832 2A50    SPEED Execution
16129 3F01    SPEED setting
03678 0E5E    SQR() Execution
06378 18EA    STOP Execution
11756 2DEC    STORE Execution
10411 28AB    STR$() Execution
05468 155C    SUBTRACT (-) Execution
17334 43B6    Screen initialize B columns C lines HL name table A and a' pattern
16960 4240    Screen line buffer (32 characters)
16203 3F4B    Screen write vector
18112 46C0    Scroll screen
16993 4261    Scroll start line
16994 4262    Scroll up to column #
24980 6194    Set file data buffer
21843 5553    Set read from device
25354 630A    Set system date
25355 630B    Set system date -- Day byte
25356 630C    Set system date -- Month byte
25358 630E    Set system date -- Year byte
21839 554F    Set write to device
16766 417E    Shape Tbale address
26574 67CE    Shape table default shape
16178 3F32    Significant digits on output (default 9)
07299 1C83    Stack check for underflow
05975 1757    Stack set up
16099 3EE3    Start of numeric variables
16111 3EEF    Start of string space
16101 3EE5    Start of token buffer
16095 3EDF    Start of variables
10616 2978    String cut routine
10464 28E0    String length determination (for LEFT RIGHT MID)
07311 1C8F    String space check
01739 06CB    Subtract DEBC from FPA1
01575 0627    Subtract FPA2 from FPA1
25257 62A9    System file data ->BASICPGM
```

```
03912 0F48    TAN() Execution - SIN/COS
16197 3F45    TAPE/DISK read vector
16201 3F49    TAPE/DISK write vector
18453 4815    TEXT Execution
11065 2B39    TEXT Execution vector
17192 4327    TEXT FLASH frequency (default 12)
17046 4296    TEXT Screen initialization
17059 42A3    TEXT background colour
17164 430C    TEXT blank fill character (default 32)
17053 429D    TEXT border colour
17115 42DB    TEXT character colour
17126 42E6    TEXT inverse colour
06336 18C0    TRACE Execution
06069 17B5    TRACE check during RUN
16105 3EE9    Temporary variables used while parsing line
02326 0916    Toggle FPA1 or FPA2 - if zero make non-zero
16503 4077    Token buffer (up to 255 bytes)
16205 3F4D    Token buffer length
16995 4263    Top column for HOME--scroll region
16958 423E    Top cursor margin in all modes
16996 4264    Top line for HOME----scroll region

20640 50A0    UNLOCK Execution
19726 4D0E    UNLOCK ^D vector
20015 4E2F    UNLOCK immediate mode vector
16130 3F02    USR Address
10073 2759    USR Execution

10309 2845    VAL() Execution
11219 2BD3    VLIN Execution
18932 49F4    VLIN graphics
26184 6648    VPOS calculation
10857 2A69    VPOS() Execution
16997 4265    VRAM name table (current)
16999 4267    VRAM pattern table (current)
11330 2C42    VTAB Execution
26219 666B    VTAB cursor adjustment
00260 0104    Version of Basic

10126 278E    WAIT Execution
22455 57B7    WRITE Execution
19803 4D5B    WRITE ^D vector
22475 57CB    WRITE sequential
03260 0CBC    Write FPA1 in ASCII to input buffer (3F77)
23796 5CF4    Write one byte to file
24099 5E23    Write program to medium
22923 598B    Write random

11405 2C8D    XDRAW Execution
26884 6904    XDRAW at xy
26297 66B9    XDRAW with no xy

19653 4CC5    ^D End of buffer i.e. exit
19538 4C52    ^D Execution
19667 4CD3    ^D Get execution address for I/O command
19609 4C99    ^D Parse first character of I/O commands
19629 4CAD    ^D Parse rest of I/O command
17017 4279    ^D buffer
17040 4290    ^D buffer current position
17016 4278    ^D buffer used length
19448 4BF8    ^D echo if MON C
17014 4276    ^D flag
```

```
BASIC RST ALL RETURN  No apparent use is made of them anywhere
0:0000 RET                  .     201
1:0001 NOP                  .       0
2:0002 RST  $38             .     255
3:0003 NOP                  .       0
4:0004 RST  $38             .     255
5:0005 NOP                  .       0
6:0006 RST  $38             .     255
7:0007 NOP                  .       0
8:0008 RET                  .     201
9:0009 NOP                  .       0
10:000A RST  $38            .     255
11:000B NOP                 .       0
12:000C RST  $38            .     255
13:000D NOP                 .       0
14:000E RST  $38            .     255
15:000F NOP                 .       0
16:0010 RET                 .     201
17:0011 NOP                 .       0
18:0012 RST  $38            .     255
19:0013 NOP                 .       0
20:0014 RST  $38            .     255
21:0015 NOP                 .       0
22:0016 RST  $38            .     255
23:0017 NOP                 .       0
24:0018 RET                 .     201
25:0019 NOP                 .       0
26:001A RST  $38            .     255
27:001B NOP                 .       0
28:001C RST  $38            .     255
29:001D NOP                 .       0
30:001E RST  $38            .     255
31:001F NOP                 .       0
32:0020 RET                 .     201
33:0021 NOP                 .       0
34:0022 RST  $38            .     255
35:0023 NOP                 .       0
36:0024 RST  $38            .     255
37:0025 NOP                 .       0
38:0026 RST  $38            .     255
39:0027 NOP                 .       0
40:0028 RET                 .     201
41:0029 NOP                 .       0
42:002A RST  $38            .     255
43:002B NOP                 .       0
44:002C RST  $38            .     255
45:002D NOP                 .       0
46:002E RST  $38            .     255
47:002F NOP                 .       0
48:0030 RET                 .     201
49:0031 NOP                 .       0
50:0032 RST  $38            .     255
51:0033 NOP                 .       0
52:0034 RST  $38            .     255
53:0035 NOP                 .       0
54:0036 RST  $38            .     255
55:0037 NOP                 .       0
56:0038 RET                 .     201
57:0039 NOP                 .       0
58:003A RST  $38            .     255
59:003B NOP                 .       0
60:003C RST  $38            .     255
61:003D NOP                 .       0
62:003E RST  $38            .     255
63:003F NOP                 .       0
```

```
END OF RST  TABLE more FILL which can be reused
64:0040 RST  $38              .    255
65:0041 NOP                   .      0
66:0042 RST  $38              .    255
67:0043 NOP                   .      0
68:0044 RST  $38              .    255
69:0045 NOP                   .      0
70:0046 RST  $38              .    255
71:0047 NOP                   .      0
72:0048 RST  $38              .    255
73:0049 NOP                   .      0
74:004A RST  $38              .    255
75:004B NOP                   .      0
76:004C RST  $38              .    255
77:004D NOP                   .      0
78:004E RST  $38              .    255
79:004F NOP                   .      0
80:0050 RST  $38              .    255
81:0051 NOP                   .      0
82:0052 RST  $38              .    255
83:0053 NOP                   .      0
84:0054 RST  $38              .    255
85:0055 NOP                   .      0
86:0056 RST  $38              .    255
87:0057 NOP                   .      0
88:0058 RST  $38              .    255
89:0059 NOP                   .      0
90:005A RST  $38              .    255
91:005B NOP                   .      0
92:005C RST  $38              .    255
93:005D NOP                   .      0
94:005E RST  $38              .    255
95:005F NOP                   .      0
96:0060 RST  $38              .    255
97:0061 NOP                   .      0
98:0062 RST  $38              .    255
99:0063 LD   A,(HL)           .    126
100:0064 NOP                  .      0
101:0065 NOP                  .      0

NMI starts at 66H..used for FLASH
102:0066 PUSH AF              .    245
103:0067 LD   A,(4270:17008) :pB    58 112  66     graphics mode
106:006A OR   A         183
107:006B JR   NZ,0078:120     .     32  11        if not TEXT mode
109:006D DEC  A               =     61
110:006E LD   (4272:17010),A 2rB    50 114  66     flash flag
113:0071 LD   A,(4271:17009) :qB    58 113  66     print to screen flag
116:0074 OR   A         183
117:0075 CALL NZ 007B:123     .{.   196 123   0
120:0078 POP  AF              .    241
121:0079 RET NMI              .E   237  69
123:007B PUSH BC              .    197
124:007C LD   A,(4273:17011) :sB    58 115  66     flash decrement
127:007F DEC  A               =     61
128:0080 JP   NZ,416A:16746   .jA  194 106  65     not ready to FLASH
131:0083 PUSH DE              .    213
132:0084 PUSH HL              .    229
133:0085 PUSH IX              ..   221 229
135:0087 PUSH IY              ..   253 229
137:0089 LD   HL,(4274:17012) *tB    42 116  66     name table for flash
140:008C LD   A,0010:16       >.    62  16
142:008E XOR  H         172
143:008F LD   H,A             g    103
144:0090 LD   (4274:17012),HL"tB    34 116  66     change VRAM FLASH page
145:0091 LD   A,0002:2        >.    62   2
149:0095 CALL FD29:64809      .).   205  41 253     init VRAM table
152:0098 POP  IY              ..   253 225
154:009A POP  IX              ..   221 225
156:009C POP  HL              .    225
157:009D POP  DE              .    209
158:009E LD   A,000C:12       >.    62  12
160:00A0 LD   (4273:17011),A 2sB    50 115  66     reset FLASH cycle
163:00A3 XOR  A         175
164:00A4 LD   (4272:17010),A 2rB    50 114  66     flash flag
167:00A7 CALL FD23:64803      .#.   205  35 253     read VDP register
170:00AA POP  BC              .    193
171:00AB RET                  .    201
```

```
UNUSED-- all 0,255,0,255
172:00AC NOP                    .      0
                                 to
254:00FE NOP                    .      0
255:00FF RST  $38               .    255

COLD START INITIALLY JUMPS TO 3ED9
256:0100 JP   3EA3:16035        ..>  195 163  62     main loop
259:0103 195                    ???
260:0104  79   1                version of basic
262:0106  16  39                10000 in binary
264:0108  232  3                1000
266:010A  100  0                100
268:010C   10  0                10
270:010E    1  0                1

ADDRESS TOKEN ROUTINE SIZE WORD -- BASIC KEY WORDS
  272     1   938:03AA  0  ----> same as LET if no word found
  276     2   941:03AD  5  GOSUB
  285     3   941:03AD  4  GOTO
  293     4   983:03D7  5  INPUT
  302     5   938:03AA  3  LET
  309     6   977:03D1  4  NEXT
  317     7   996:03E4  5  PRINT
  326     8   980:03D4  4  READ
  334     9   953:03B9  3  REM
  341    10   959:03BF  3  FOR
  348    11   968:03C8  2  IF
  354    12   956:03BC  4  DATA
  362    13   980:03D4  3  DIM
  369    14   986:03DA  2  ON
  375    15   991:03DF  5  ONERR
  384    16   976:03D0  4  STOP
  392    17   976:03D0  6  RETURN
  402    18   976:03D0  3  END
  409    19   973:03CD  3  DEF
  416    20   976:03D0  5  CLEAR
  425    21   976:03D0  6  RESUME
  435    22   976:03D0  3  NEW
  442    23   976:03D0  3  POP
  449    24   944:03B0  3  RUN
  456    25   950:03B6  4  LIST
  464    26   976:03D0  5  TRACE
  473    27   976:03D0  7  NOTRACE
  484    28   947:03B3  3  DEL
  491    29   999:03E7  4  CALL
  499    30   976:03D0  4  CONT
  507    31   976:03D0  6  CLRERR
  517    32  1002:03EA  3  GET
  524    33  1017:03F9  4  POKE
  532    34   976:03D0  7  RESTORE
  543    35   976:03D0  4  HOME
  551    36  1024:0400  4  DRAW
  559    37  1024:0400  5  XDRAW
  568    38   976:03D0  5  FLASH
  577    39   976:03D0  7  INVERSE
  588    40   976:03D0  6  NORMAL
  598    41   976:03D0  4  TEXT
  606    42   976:03D0  2  GR
  612    43   976:03D0  3  HGR
  619    44   976:03D0  4  HGR2
  627    45  1029:0405  4  HLIN
  635    46  1029:0405  4  VLIN
  643    47  1014:03F6  5  HPLOT
  652    48  1017:03F9  4  PLOT
  660    49   999:03E7  4  HTAB
  668    50   999:03E7  4  VTAB
  676    51   976:03D0  6  SHLOAD
  686    52  1002:03EA  6  RECALL
  696    53  1002:03EA  5  STORE
  705    54  1005:03ED  4  WAIT
  713    55  1040:0410  5  SPEED
  722    56  1040:0410  3  ROT
  729    57  1040:0410  5  SCALE
  738    58  1040:0410  5  COLOR
  747    59  1040:0410  6  HCOLOR
  757    60  1045:0415  2  IN
  763    61  1045:0415  2  PR
```

```
  769     62 1050:041A  5  HIMEM
  778     63 1050:041A  5  LOMEM
  787     64  976:03D0  5  BREAK
  796     65  976:03D0  7  NOBREAK
  807      7  996:03E4  1  ?
  812     66  953:03B9  1  &
  817  0                          end of table

OPERATORS TABLE
ADDRESS SIZE WORD TOKEN
  818     1  +       160
  821     1  -       161
  824     1  *       162
  827     1  /       163
  830     1  ^       164
  833     1  <       165
  836     1  >       166
  839     2  <=      167
  843     2  >=      168
  847     2  <>      169
  851     1  =       170
  854     3  AND     171
  859     2  OR      172
  863     3  NOT     173
  868     4  GOTO    174
  874     5  GOSUB   175
  881     4  STEP    176
  887     2  AT      177
  891     4  THEN    178
  897     4  THEN    179
  903     2  TO      180
  907     1  :       181
  910     1  #       182
  913     1  (       183
  916     1  )       184
  919     1  ,       185
  922     1  ;       166
  925     2  =<      167
  929     2  =>      168
  933     2  ><      169
  937     0                  end of table

BASIC WORDS PARSING TABLE --> first is number of routines to do
LET
938           1  3AAC:15020          1 172  58
GOSUB GOTO
941:03AD      1  3D8C:15756  ..=     1 140  61
RUN
944:03B0      1  3B80:15232  ..;     1 128  59
DELETE
947:03B3      1  3B8F:15247  ..;     1 143  59
LIST
950:03B6      1  3B8B:15243  ..;     1 139  59
REM &
953:03B9      1  3DC9:15817  ..=     1 201  61
DATA
956:03BC      1  3DC6:15814  ..=     1 198  61
FOR
959:03BF      4  3A99:15001  .       4 153  58
961:03C1         3E5B:15963) :[>    91  62
964:03C4         3AF5:15093  .     245  58
965:03C5         3AF5:15093) :.:   245  58
IF
969:03C9      2  3A63:14947  c       2  99  58
970:03CA         3ABB:15035) :.:   187  58
DEF
973:03CD      1  3B15:15125  ..;     1  21  59
STOP RETURN END CLEAR RESUME NEW POP TRACE NOTRACE CONT CLRERR RESTORE HOME
FLASH INVERSE NORMAL TEXT GR HGR HGR2 SHLOAD BREAK NOBREAK
976:03D0 NOP                  .       0
NEXT
977:03D1      1  3CCF:15567  ..<     1 207  60
READ DIM
980:03D4      1  3CD6:15574  ..<     1 214  60
INPUT
983:03D7      1  3CB7:15543  ..<     1 183  60
```

```
ON
987:03DB      2  3A16:14875  .      2  27  58
988:03DC         3B69:15209) :i;  105  59
ONERR
991:03DF      2  3E77:15991  .      2 119  62
993:03E1         3D8C:15756  >.   140  61
PRINT ?
996:03E4      1  3CDC:15580  ..<    1 220  60
CALL HTAB VTAB
999:03E7      1  3A1B:14875  ..:    1  27  58
GET RECALL STORE
1002:03EA     1  3C04:15364  ..<    1   4  60
WAIT
1006:03EE     4  3A16:14875  .      4  27  58
1007:03EF        3E43:15939) :C>   67  62
1010:03F2        3A16:14875  .     27  58
1011:03F3        3A79:14969) :y:  121  58
HPLOT
1014:03F6     1  3AFE:15102  ..:    1 254  58
PLOT
1018:03FA     3  3A16:14875  .      3  27  58
1019:03FB        3E43:15939) :C>   67  62
1022:03FE        3A16:14875   .     27  58
DRAW XDRAW
1023:03FF     2  3A16:14875   :..    2  27  58
1026:0402        3A80:14976) :.:  128  58
HLIN VLIN
1030:0406     5  3A16:14875  .      5  27  58
1031:0407        3E43:15939) :C>   58  67  62
1034:040A        3A16:14875  .     27  58
1035:040B        3E69:15977) :i>   105  62
1038:040E        3A16:14875   .     27  58
SPEED ROT SCALE COLOR HCOLOR
1040:0410     2  3436:13366           2  54  62
1042:0412        3A16:14875   >.     27  58
IN PR
1045:0415     2  3E4E:15950           2  78  62
1047:0417        3A16:14875   >.     27  58
1050 HIMEM LOMEM
1050:041A     2  3E27:15911           2  39  62
1052:041C        3A16:14875   >.     27  58

 Error messages      Format:length, message in ASCII
1048:0418         :                                58   4
1056:0420  Hi Cathy  72 105  32  67  97 116 104 121
1064:0428   FATAL S  18  70  65  84  65  76  32  83
1072:0430  YSTEM ER  89  83  84  69  77  32  69  82
1080:0438  ROR       82  79  82  28  12  32  32  32
1088:0440    Coleco  32  32  67 111 108 101  99 111
1096:0448   SmartBA  32  83 109  97 114 116  66  65
1104:0450  SIC V1.0  83  73  67  32  86  49  46  48
1112:0458   (c) 198  32  40  99  41  32  49  57  56
1120:0460  3, Lazer  51  44  32  76  97 122 101 114
1128:0468   MicroSy  32  77 105  99 114 111  83 121
1136:0470  stems In 115 116 101 109 115  32  73 110
1144:0478  c ]  :    99   1  93   0   1  58   1  13
1145                  1       number of prompts
1146       ]         93       ASCII of first prompt
1147                  0       ASCII of second prompt
1149       :         53       ASCII of statement separator
```

```
1152:0480   NEXT wi  16  78  69  88  84  32 119 105
1160:0488   thout FO 116 104 111 117 116  32  70  79
1168:0490   R Syntax  82   6  83 121 110 116  97 120
1176:0498    RETURN   20  82  69  84  85  82  78  32
1184:04A0   without  119 105 116 104 111 117 116  32
1192:04A8   GOSUB Ou  71  79  83  85  66  11  79 117
1200:04B0   t of DAT 116  32 111 102  32  68  65  84
1208:04B8   A Illega  65  16  73 108 108 101 103  97
1216:04C0   l Quanti 108  32  81 117  97 110 116 105
1224:04C8   ty Overf 116 121   8  79 118 101 114 102
1232:04D0   low Out  108 111 119  13  79 117 116  32
1240:04D8   of Memor 111 102  32  77 101 109 111 114
1248:04E0   y Stack  121  14  83 116  97  99 107  32
1256:04E8   Overflow  79 118 101 114 102 108 111 119
1264:04F0    Undefin  19  85 110 100 101 102 105 110
1272:04F8   ed State 101 100  32  83 116  97 116 101
1280:0500   ment Bad 109 101 110 116  13  66  97 100
1288:0508    Subscri  32  83 117  98 115  99 114 105
1296:0510   pt Redim 112 116  19  82 101 100 105 109
1304:0518   ensioned 101 110 115 105 111 110 101 100
1312:0520    Array D  32  65 114 114  97 121  14  68
1320:0528   ivide by 105 118 105 100 101  32  98 121
1328:0530    Zero Ty  32  90 101 114 111  13  84 121
1336:0538   pe Misma 112 101  32  77 105 115 109  97
1344:0540   tch Stri 116  99 104  15  83 116 114 105
1352:0548   ng Too L 110 103  32  84 111 111  32  76
1360:0550   ong Form 111 110 103  19  70 111 114 109
1368:0558   ula Too  117 108  97  32  84 111 111  32
1376:0560   Complex   67 111 109 112 108 101 120  18
1384:0568   Undefine  85 110 100 101 102 105 110 101
1392:0570   d Functi 100  32  70 117 110  99 116 105
1400:0578   on Incor 111 110  24  73 110  99 111 114
1408:0580   rect Fun 114 101  99 116  32  70 117 110
1416:0588   ction Us  99 116 105 111 110  32  85 115
1424:0590   age Ille  97 103 101  12  73 108 108 101
1432:0598   gal Mode 103  97 108  32  77 111 100 101
1440:05A0      Break   0   0   5  66 114 101  97 107
1448:05A8    Can't C  14  67  97 110  39 116  32  67
1456:05B0   ontinue  111 110 116 105 110 117 101   0
```

```
1463        0       add offset to message number from 1152
1464       16       Syntax
1465       22       RETURN without GOSUB
1466       42       Out of DATA
1467       53       Illegal Quantity
1468       69       Overflow
1469       77       Out of Memory
1470       77       Out of Memory
1471       90       Undefined Statement
1472      107       Bad Subscript
1473      120       Redimensioned array
1474      133       Divide by Zero
1475      163       Type Mismatch
1476      176       String too Long
1477      191       Formula too Complex
1478      224       Bad FN
```

```
DON'T KNOW WHAT THESE ARE
1479       16
1480      149
1481      254
1482      255
1483      255
1484      210
1485        1
1486        2
1487        4
1488        5
1489        6
1490        7
1491        8
1492        9
1493       10
1494       11
1495       12
1496       13
1497       14
1498       15
1499       13
```

```
Load HL with Line Number
1500:05DC CALL 1733:5939       .3.  205  51  23    get equation from token line

1503:05DF OR   A                .    183
1504:05E0 JP   NZ,1F03:7939     ...  194   3  31    Type mismatch error
1507:05E3 CALL 0932:2354        .2.  205  50   9    load HL with FPA1
1510:05E6 JP   C,1F00:7936      ...  218   0  31    Illegal quantity
1513:05E9 LD   A,H              |    124
1514:05EA OR   H                .    180
1515:05EB RET  Z                .    200
1516:05EC JP   1F00:7936        ...  195   0  31    Illegal quantity

Load BC with number for GOTO
1519:05EF CP   008A:138         ..   254 138
1521:05F1 JP   C,0602:1538      ...  218   2   6
1524:05F4 JP   Z,0608:1544      ...  202   8   6
1527:05F7 EXX                   .    217
1528:05F8 DEC  C                .     13
1529:05F9 DEC  C                .     13
1530:05FA EXX                   .    217
1531:05FB INC  DE               .     19
1532:05FC LD   A,(DE)           .     26
1533:05FD LD   C,A              O     79
1534:05FE INC  DE               .     19
1535:05FF LD   A,(DE)           .     26
1536:0600 LD   B,A              G     71
1537:0601 RET                   .    201

1538:0602 AND  000F:15          ..   230  15
1540:0604 LD   C,A              O     79
1541:0605 LD   B,0000:0         ..     6   0
1543:0607 RET                   .    201

1544:0608 EXX                   .    217
1545:0609 DEC  C                .     13
1546:060A EXX                   .    217
1547:060B INC  DE               .     19
1548:060C LD   A,(DE)           .     26
1549:060D LD   C,A              O     79
1550:060E LD   B,0000:0         ..     6   0
1552:0610 RET                   .    201

Print Floating Point Number
1553:0611 PUSH DE               .    213
1554:0612 PUSH HL               .    229
1555:0613 PUSH BC               .    197
1556:0614 CALL 0CBC:3260        ...  205 188  12    change to ASCII
1559:0617 LD   HL,3F76:16246    !v?   33 118  63    start of buffer
1562:061A CALL 2F4E:12110       .N/  205  78  47    print it
1565:061D POP  BC               .    193
1566:061E POP  HL               .    225
1567:061F POP  DE               .    209
1568:0620 RET                   .    201

Add (HL) to Floating Point Number
1569:0621 CALL 1117:4375        ...  205  23  17    move (HL) to FPA2
1572:0624 JP   062F:1583        ./.  195  47   6    add FPA2 to FPA1
1575:0627 LD   A,(3F2E:16174)   :.?   58  46  63    most significant digit 2
1578:062A XOR  0080:128         ..   238 128        reverse sign
1580:062C LD   (3F2E:16174),A   2.?   50  46  63
1583:062F LD   A,(3F26:16166)   :&?   58  38  63    exponent 1
1586:0632 LD   HL,(3F2E:16174)  *.?   42  46  63    exponent 2
1589:0635 SUB   H               .    148
1590:0636 JR   C,0646:1606      8.    56  14        if FPA2 is bigger
1592:0638 LD   BC,(3F2B:16171)       237  75  43  63 load FPA2
1596:063C LD   DE,(3F2D:16173)       237  91  45  63
1600:0640 LD   HL,(3F25:16165)        42  37  63
1603:0643 JP   0653:1619             195  83   6
1606:0646 LD   BC,(3F22:16162)       237  75  34  63 load FPA1
1610:064A LD   DE,(3F24:16164)       237  91  36  63
1614:064E CALL 0728:1832        .(.  205  40   7     move FPA2 to FPA1
1617:0651 NEG  A                .D   237  68
1619:0653 CP   H                .    188
1620:0654 RET  Z                .    200
```

```
1621:0655 EX   AF,AF'              .      8
1622:0656 LD   A,(3F25:16165) :%?    58  37  63    most significant byte
1625:0659 LD   (3F17:16151),A 2.?    50  23  63    save it
1628:065C XOR  D                   .    170
1629:065D EX   AF,AF'              .      8
1630:065E LD   HL,3F25:16165  !%?    33  37  63
1633:0661 SET  7,(HL)              ..   203 254        set MSB
1635:0663 SET  7,D                 ..   203 250
1637:0665 LD   L,0000:0            ..    46   0
1639:0667 LD   H,A                 g    103
1640:0668 SRL  H                   .<   203  60
1642:066A SRL  H                   .<   203  60
1644:066C SRL  H                   .<   203  60
1646:066E JR   Z,0679:1657           (.     40          9
1648:0670 LD   L,C                 i    105
1649:0671 LD   C,B                 H     72
1650:0672 LD   B,E                 C     67
1651:0673 LD   E,D                 Z     90
1652:0674 LD   D,0000:0            ..    22   0
1654:0676 DEC  H                   %     37
1655:0677 JR   NZ,0670:1648              .     32 247
1657:0679 AND  0007:7              ..   230   7
1659:067B JR   Z,068A:1674           (.     40         13
1661:067D SRL  D                   .:   203  58
1663:067F RR   E                   ..   203  27
1665:0681 RR   B                   ..   203  24
1667:0683 RR   C                   ..   203  25
1669:0685 RR   L                   ..   203  29
1671:0687 DEC  A                   =     61
1672:0688 JR   NZ,067D:1661              .     32 243
1674:068A EX   AF,AF'              .      8
1675:068B LD   A,L                 }    125
1676:068C JP   M,06CB:1739               250 203   6
1679:068F LD   HL,(3F22:16162)            42  34  63   least significant byte 1
1682:0692 ADD  HL,BC                       9
1683:0693 LD   B,H                        68
1684:0694 LD   C,L                        77
1685:0695 LD   HL,(3F24:16164)            42  36  63   next 2 bytes
1688:0698 ADC  HL,DE                     237  90
1690:069A JR   NC,06A8:1704            0.     48        12
1692:069C RR   H                   ..   203  28
1694:069E RR   L                   ..   203  29
1696:06A0 RR   B                   ..   203  24
1698:06A2 RR   C                   ..   203  25
1700:06A4 RR   A                   .     31
1701:06A5 CALL 0753:1875           .S.  205  83   7    multiply FPA1 by 2
1704:06A8 RL   A                   .     23           increment all digits
1705:06A9 JR   NC,06BB:1723          0.     48        16         as required
1707:06AB INC  C                   .     12
1708:06AC JP   NZ,06BB:1723              ...   194 187   6
1711:06AF INC  B                   .      4
1712:06B0 JP   NZ,06BB:1723              ...   194 187   6
1715:06B3 INC  L                   ,     44
1716:06B4 JP   NZ,06BB:1723              ...   194 187   6
1719:06B7 INC  H                   $     36
1720:06B8 CALL Z  0753:1875        .S.  204  83   7    multiply FPA1 by 2
1723:06BB LD   A,(3F17:16151) :.?    58  23  63    saved control byte
1726:06BE RL   H                   ..   203  20
1728:06C0 RL   A                   .     23
1729:06C1 RR   H                         203  28
1731:06C3 LD   (3F24:16164),HL           34  36  63   write result to FPA1
1734:06C6 LD   (3F22:16162),BC           237  67  34  63
1738:06CA RET                      .     201
```

```
Subtract BC from LSB and DE from MSB
1739:06CB NEG  A                        237  68
1741:06CD LD   HL,(3F22:16162)           42  34  63
1744:06D0 SBC  HL,BC                    237  66
1746:06D2 LD   B,H                       68
1747:06D3 LD   C,L                       77
1748:06D4 LD   HL,(3F24:16164)           42  36  63
1751:06D7 SBC  HL,DE             .R     237  82
1753:06D9 LD   E,A                       95
1754:06DA JR   NC,06F4:1780      -  0.       48        24
1756:06DC LD   A,(3F17:16151)  :.?       58  23  63    stored exponent byte
1759:06DF XOR  0080:128          ..     238 128
1761:06E1 LD   (3F17:16151),A  2.?       50  23  63
1764:06E4 XOR  A                 .      175
1765:06E5 SUB   E                .      147
1766:06E6 EX   DE,HL             .      235
1767:06E7 LD   HL,0000:0         !..     33   0   0
1770:06EA SBC  HL,BC             .B     237  66
1772:06EC LD   B,H               D       68
1773:06ED LD   C,L               M       77
1774:06EE LD   HL,0000:0         !..     33   0   0
1777:06F1 SBC  HL,DE             .R     237  82
1779:06F3 LD   E,A                       95
1780:06F4 LD   A,(3F26:16166)  :&?       58  38  63    exponent byte
1783:06F7 LD   D,A               W       87
1784:06F8 LD   A,H               |      124
1785:06F9 OR   A                 .      183
1786:06FA JR   NZ,071C:1820              32  32
1788:06FC LD   H,L               e      101
1789:06FD LD   L,B               h      104
1790:06FE LD   B,C               A       65
1791:06FF LD   C,E               K       75
1792:0700 LD   E,0000:0          ..      30   0
1794:0702 LD   A,D               z      122
1795:0703 SUB   0008:8           ..     214   8
1797:0705 LD   D,A               W       87
1798:0706 JR   Z,070A:1802         (.     40          2
1800:0708 JR   NC,06F8:1784       0.      48 238
1802:070A XOR  A                 .      175
1803:070B LD   (3F26:16166),A  2&?       50  38  63    exponent
1806:070E RET                    .      201

routine enters at 1820:071C
1807:070F DEC  D                 .       21
1808:0710 JR   Z,070A:1802         (.     40 248
1810:0712 SLA  E                .#      203  35
1812:0714 RL   C                 ..      203  17
1814:0716 RL   B                 ..      203  16
1816:0718 RL   L                 ..      203  21
1818:071A RL   H                 ..      203  20
1820:071C JP   P,070F:1807         ...   242        15   7
1823:071F LD   A,D               z      122
1824:0720 LD   (3F26:16166),A  2&?       50  38  63    exponent
1827:0723 RL   E                         203  19
1829:0725 JP   06A9:1705                 195 169   6

Move FPA2 to FPA1  no LDIR to save stack activity
1832:0728 LD   HL,(3F2B:16171)           42  43  63
1835:072B LD   (3F22:16162),HL           34  34  63
1838:072E LD   HL,(3F2D:16173)           42  45  63
1841:0731 LD   (3F24:16164),HL           34  36  63
1844:0734 LD   HL,(3F2E:16174)           42  46  63
1847:0737 LD   (3F25:16165),HL           34  37  63
1850:073A RET                    .      201

Compare signs of FPA1 and FPA2
1851:073B LD   DE,(3F25:16165)           237  91  37  63 MSB and exponent 1
1855:073F LD   HL,(3F2E:16174)            42  46  63    MSB and exponent 2
1858:0742 LD   A,L                       125
1859:0743 XOR  E                         171            hi bit on if signs diff
1860:0744 LD   (3F17:16151),A             50  23  63    save it
1863:0747 SET  7,E                       203 251        turn on high bits for
1865:0749 SET  7,L                       203 253        calculations
1867:074B LD   (3F2E:16174),HL           34  46  63
1870:074E LD   (3F25:16165),DE           237  83  37  63
1874:0752 RET                    .      201
```

```
Multiply FPA1 by 2
1875:0753 EX   DE,HL              .     235
1876:0754 LD   HL,3F26:16166      !&?    33  38  63    exponent
1879:0757 INC  (HL)               4      52
1880:0758 JP   Z,1EF1:7921          ...  202 241  30          if overflow
1883:075B EX   DE,HL              .     235
1884:075C RET                     .     201

multiply FPA1 by (HL)
1885:075D CALL 1117:4375          ...   205  23  17    put (HL) in FPA2

Multiply FPA2 by FPA1
1888:0760 CALL 073B:1851          .;.   205  59   7    compare signs
1891:0763 LD   A,D                z     122
1892:0764 OR   A                  .     183
1893:0765 RET  Z                  .     200
1894:0766 LD   A,H                |     124
1895:0767 OR   A                  .     183
1896:0768 JR   Z,070B:1803        (.     40 161
1898:076A ADD  A,D                .     130
1899:076B JR   NC,0779:1913       0.     48  12
1901:076D SUB   0081:129          ..    214 129
1903:076F JP   M,077D:1917        .}.   250 125   7
1906:0772 CP   007F:127           ..    254 127
1908:0774 JR   Z,077D:1917        .      40   7
1910:0776 JP   1EF1:7921          ...   195 241  30    overflow
1913:0779 SUB   0081:129          ..    214 129
1915:077B JR   C,070A:1802        8.     56 141
1917:077D LD   (3F26:16166),A     2&?    50  38  63    exponent
1920:0780 LD   BC,0000:0          ...     1   0   0
1923:0783 LD   D,B                P      80
1924:0784 LD   E,C                Y      89
1925:0785 LD   A,(3F22:16162)     :"?    58  34  63    LSB
1928:0788 CALL 07B0:1968          ...   205 176   7    A*FPA2
1931:078B LD   A,(3F23:16163)     :#?    58  35  63
1934:078E CALL 07B0:1968          ...   205 176   7
1937:0791 LD   A,(3F24:16164)     :$?    58  36  63
1940:0794 CALL 07B0:1968          ...   205 176   7
1943:0797 LD   A,(3F25:16165)     :%?    58  37  63    MSB
1946:079A CALL 07B0:1968          ...   205 176   7
1949:079D EX   AF,AF'             .       8
1950:079E EX   DE,HL              .     235
1951:079F BIT  7,H                .|    203 124
1953:07A1 JP   NZ,06A5:1701         ...  194 165   6
1956:07A4 RL   A                  .      23
1957:07A5 RL   C                  ..    203  17
1959:07A7 RL   B                  ..    203  16
1961:07A9 RL   L                  ..    203  21
1963:07AB RL   H                  ..    203  20
1965:07AD JP   06A8:1704          ...   195 168   6

do multiplication A*FPA2
1968:07B0 OR   A                  .     183
1969:07B1 JR   Z,07D9:2009          (&    40        38
1971:07B3 SET  CF                 7      55
1972:07B4 RR   A                  .      31
1973:07B5 JR   NC,07C9:1993         0.    48        18
1975:07B7 EX   AF,AF'             .       8
1976:07B8 LD   HL,(3F2B:16171)    *+?    42  43  63   LSB FPA2
1979:07BB ADD  HL,BC              .       9
1980:07BC LD   B,H                D      68
1981:07BD LD   C,L                M      77
1982:07BE LD   HL,(3F2D:16173)    *-?    42  45  63   MSB FPA2
1985:07C1 ADC  HL,DE              .Z    237  90
1987:07C3 EX   DE,HL              .     235
1988:07C4 RR   D                  ..    203  26
1990:07C6 JP   07CC:1996          ...   195 204   7
1993:07C9 EX   AF,AF'             .       8
1994:07CA SRL  D                  .:    203  58
1996:07CC RR   E                  ..    203  27
1998:07CE RR   B                  ..    203  24
2000:07D0 RR   C                  ..    203  25
2002:07D2 RR   A                  .      31
2003:07D3 EX   AF,AF'             .       8
2004:07D4 SRL  A                  .?    203  63
2006:07D6 JR   NZ,07B5:1973         .    32 221
2008:07D8 RET                     .     201
```

```
?????
2009:07D9 EX   AF,AF'          .      8
2010:07DA LD   A,C             y    121
2011:07DB EX   AF,AF'          .      8
2012:07DC LD   C,B             H     72
2013:07DD LD   B,E             C     67
2014:07DE LD   E,D             Z     90
2015:07DF LD   D,0000:0        ..    22   0
2017:07E1 RET                  .    201

FPA1=FPA1/FPA2
2018:07E2 CALL 073B:1851      .;.   205  59   7    compare signs
2021:07E5 LD   A,H            |     124
2022:07E6 OR   A              .     183
2023:07E7 JP   Z,1EEE:7918      ...   202 238  30          division by zero
2026:07EA LD   A,D            z     122
2027:07EB OR   A              .     183
2028:07EC JP   Z,070B:1803      ...   202        11   7
2031:07EF SUB   H             .     148
2032:07F0 JP   NC,07FF:2047      ...   210 255   7
2035:07F3 ADD  A,0080:128      ..    198 128
2037:07F5 JP   P,0804:2052      ...   242         4   8
2040:07F8 CP   0080:128       ..    254 128
2042:07FA JR   Z,0804:2052     (.     40         8
2044:07FC JP   070A:1802             195  10   7
2047:07FF ADD  A,0080:128            198 128
2049:0801 JP   P,1EF1:7921           242 241  30    overflow
2052:0804 LD   (3F26:16166),A         50  38  63
2055:0807 LD   HL,0000:0              33   0   0    null all control bytes
2058:080A LD   (3F18:16152),HL        34  24  63
2061:080D LD   (3F1A:16154),HL        34  26  63
2064:0810 LD   (3F27:16167),HL        34  39  63
2067:0813 LD   (3F29:16169),HL        34  41  63
2070:0816 LD   (3F1E:16158),HL        34  30  63
2073:0819 LD   (3F20:16160),HL        34  32  63
2076:081C LD   D,0020:32       .      22  32       4 numbers of 8 digits

do shift through 8 bytes  work space leads to least significant byte
of FPA1 and FPA2.  Repeat 32 times for each bit of the number
2078:081E LD   HL,3F27:16167          33  39  63    temporary work space
2081:0821 LD   BC,3F1E:16158           1  30  63    temporary work space
2084:0824 LD   A,(BC)          .      10
2085:0825 SUB   (HL)           .     150
2086:0826 LD   (BC),A          .       2
2087:0827 INC  BC              .       3
2088:0828 INC  HL              #      35
2089:0829 LD   A,(BC)          .      10
2090:082A SBC  A,(HL)          .     158
2091:082B LD   (BC),A          .       2
2092:082C INC  BC              .       3
2093:082D INC  HL              #      35
2094:082E LD   A,(BC)          .      10
2095:082F SBC  A,(HL)          .     158
2096:0830 LD   (BC),A          .       2
2097:0831 INC  BC              .       3
2098:0832 INC  HL              #      35
2099:0833 LD   A,(BC)          .      10
2100:0834 SBC  A,(HL)          .     158
2101:0835 LD   (BC),A          .       2
2102:0836 INC  BC              .       3
2103:0837 INC  HL              #      35
2104:0838 LD   A,(BC)          .      10
2105:0839 SBC  A,(HL)          .     158
2106:083A LD   (BC),A          .       2
2107:083B INC  BC              .       3
2108:083C INC  HL              #      35
2109:083D LD   A,(BC)          .      10
2110:083E SBC  A,(HL)          .     158
2111:083F LD   (BC),A          .       2
2112:0840 INC  BC              .       3
2113:0841 INC  HL              #      35
2114:0842 LD   A,(BC)          .      10
2115:0843 SBC  A,(HL)          .     158
2116:0844 LD   (BC),A          .       2
```

```
2117:0845 INC  BC                .      3
2118:0846 INC  HL                #     35
2119:0847 LD   A,(BC)            .     10
2120:0848 SBC  A,(HL)            .    158
2121:0849 LD   (BC),A            .      2
2122:084A JP   C,08B4:2228         ...  218 180   8          add back if overflow
2125:084D SET  CF                7     55

Double result
2126:084E LD   HL,3F18:16152     !.?    33  24  63
2129:0851 RL   (HL)              ..    203  22
2131:0853 INC  HL                #     35
2132:0854 RL   (HL)              ..    203  22
2134:0856 INC  HL                #     35
2135:0857 RL   (HL)              ..    203  22
2137:0859 INC  HL                #     35
2138:085A RL   (HL)              ..    203  22

Halve divisor
2140:085C LD   HL,3F2E:16174     !.?    33  46  63
2143:085F SRL  (HL)              .>    203  62
2145:0861 DEC  HL                +      43
2146:0862 RR   (HL)              ..    203  30
2148:0864 DEC  HL                +      43
2149:0865 RR   (HL)              ..    203  30
2151:0867 DEC  HL                +      43
2152:0868 RR   (HL)              ..    203  30
2154:086A DEC  HL                +      43
2155:086B RR   (HL)              ..    203  30
2157:086D DEC  HL                +      43
2158:086E RR   (HL)              ..    203  30
2160:0870 DEC  HL                +      43
2161:0871 RR   (HL)              ..    203  30
2163:0873 DEC  HL                +      43
2164:0874 RR   (HL)              ..    203  30
2166:0876 DEC  D                 .      21
2167:0877 JP   NZ,081E:2078             194  30   8
2170:087A LD   A,(3F1B:16155)           58  27  63
2173:087D OR   A                        183
2174:087E JP   M,089E:2206              250 158   8
2177:0881 LD   HL,(3F18:16152)          42  24  63
2180:0884 ADD  HL,HL                    41
2181:0885 LD   (3F22:16162),HL          34  34  63
2184:0888 LD   HL,(3F1A:16154)          42  26  63
2187:088B ADC  HL,HL                   237 106
2189:088D LD   (3F24:16164),HL          34  36  63
2192:0890 LD   A,(3F17:16151)           58  23  63
2195:0893 OR   A                 .     183
2196:0894 RET  M                 .     248
2197:0895 LD   A,(3F25:16165)           58  37  63
2200:0898 AND  007F:127    230 127
2202:089A LD   (3F25:16165),A           50  37  63
2205:089D RET                          201
2206:089E LD   A,(3F26:16166)           58  38  63
2209:08A1 INC  A                        60
2210:08A2 LD   (3F26:16166),A           50  38  63
2213:08A5 LD   HL,(3F18:16152)          42  24  63
2216:08A8 LD   (3F22:16162),HL          34  34  63
2219:08AB LD   HL,(3F1A:16154)          42  26  63
2222:08AE LD   (3F24:16164),HL          34  36  63
2225:08B1 JP   0890:2192               195 144   8
```

```
Add bits back on negative result in MSB
2228:08B4 LD   HL,3F27:16167          33  39  63
2231:08B7 LD   BC,3F1E:16158           1  30  63
2234:08BA LD   A,(BC)            .     10
2235:08BB ADD  A,(HL)            .    134
2236:08BC LD   (BC),A            .      2
2237:08BD INC  HL                #     35
2238:08BE INC  BC                .      3
2239:08BF LD   A,(BC)            .     10
2240:08C0 ADC A,(HL)             .    142
2241:08C1 LD   (BC),A            .      2
2242:08C2 INC  HL                #     35
2243:08C3 INC  BC                .      3
2244:08C4 LD   A,(BC)            .     10
2245:08C5 ADC A,(HL)             .    142
2246:08C6 LD   (BC),A            .      2
2247:08C7 INC  HL                #     35
2248:08C8 INC  BC                .      3
2249:08C9 LD   A,(BC)            .     10
2250:08CA ADC A,(HL)             .    142
2251:08CB LD   (BC),A            .      2
2252:08CC INC  HL                #     35
2253:08CD INC  BC                .      3
2254:08CE LD   A,(BC)            .     10
2255:08CF ADC A,(HL)             .    142
2256:08D0 LD   (BC),A            .      2
2257:08D1 INC  HL                #     35
2258:08D2 INC  BC                .      3
2259:08D3 LD   A,(BC)            .     10
2260:08D4 ADC A,(HL)             .    142
2261:08D5 LD   (BC),A            .      2
2262:08D6 INC  HL                #     35
2263:08D7 INC  BC                .      3
2264:08D8 LD   A,(BC)            .     10
2265:08D9 ADC A,(HL)             .    142
2266:08DA LD   (BC),A            .      2
2267:08DB INC  HL                #     35
2268:08DC INC  BC                .      3
2269:08DD LD   A,(BC)            .     10
2270:08DE ADC A,(HL)             .    142
2271:08DF LD   (BC),A            .      2
2272:08E0 OR   A                 .    183
2273:08E1 JP   084E:2126         .N.  195  78   8   result*2  divisor/2

ABS
2276:08E4 JP   NZ,1F03:7939           ...  194   3  31
2279:08E7 LD   HL,3F25:16165     !%?   33  37  63   most significant byte
2282:08EA RES  7,(HL)            ..   203 190       make positive
2284:08EC RET                    .    201

SGN
2285:08ED JP   NZ,1F03:7939           194   3  31
2288:08F0 LD   A,(3F26:16166)          58  38  63   exponent
2291:08F3 OR   A                      183
2292:08F4 RET  Z                      200
2293:08F5 LD   A,(3F25:16165)          58  37  63   MSB
2296:08F8 OR   A                      183
2297:08F9 LD   HL,00:0                 33   0   0
2300:08FC LD   (3F22:16162),HL         34  34  63   make FPA1 0 to display
2303:08FF LD   (3F24:16164),HL         34  36  63   result of function
2306:0902 LD   A,81:129                62 129       +1
2308:0904 LD   (3F26:16166),A          50  38  63   exponent byte
2311:0907 RET  P                      240
2312:0908 LD   HL,3F2E:16174           33  46  63
2315:090B JP   C,0911:2321            218  17   9
2318:090E LD   HL,3F25:16165           33  37  63
2321:0911 LD   A,80:128          >.    62 128       make it -1
2323:0913 XOR  (HL)              .    174
2324:0914 LD   (HL),A            w    119
2325:0915 RET                    .    201
```

```
Toggle FPA1 or FPA2 -- depends on Carry on entry
2326:0916 LD   HL,3F2F:16175   !/?   33  47  63   FPA2
2329:0919 JP   C,091F:2335       ...  218  31   9
2332:091C LD   HL,3F26:16166   !&?   33  38  63   FPA1
2335:091F LD   A,(HL)          .     126
2336:0920 OR   A               .     183
2337:0921 JP   NZ,092F:2351      ./.  194  47   9   if not 0 make it 0
2340:0924 LD   (HL),81:129     6.     54 129        make it 1
2342:0926 DEC  HL              +      43
2343:0927 LD   (HL),A          w     119
2344:0928 DEC  HL              +      43
2345:0929 LD   (HL),A          w     119
2346:092A DEC  HL              +      43
2347:092B LD   (HL),A          w     119
2348:092C DEC  HL              +      43
2349:092D LD   (HL),A          w     119
2350:092E RET                  .     201
2351:092F XOR  A               .     175
2352:0930 LD   (HL),A          w     119
2353:0931 RET                  .     201

load HL with INT(FPA1)
2354:0932 LD   A,(3F26:16166)  :&?   58  38  63   exponent
2357:0935 CP   81:129          ..    254 129       check sign
2359:0937 JP   C,0961:2401       .a.  218  97   9
2362:093A SUB   90:144         ..    214 144       2^16
2364:093C JP   NC,0965:2405      .e.  210 101   9   overflow
2367:093F NEG  A               .D    237  68
2369:0941 EX   AF,AF'          .       8
2370:0942 LD   HL,(3F24:16164) *$?   42  36  63
2373:0945 BIT  7,H             .|    203 124
2375:0947 SET  7,H             ..    203 252
2377:0949 EX   AF,AF'          .       8
2378:094A SRL  H               .<    203  60
2380:094C RR   L               ..    203  29
2382:094E DEC  A               =      61
2383:094F JP   NZ,094A:2378      .J.  194  74   9
2386:0952 EX   AF,AF'          .       8
2387:0953 JP   NZ,0958:2392      .X.  194  88   9
2390:0956 OR   A               .     183
2391:0957 RET                  .     201
2392:0958 XOR  A               .     175
2393:0959 SUB   L              .     149
2394:095A LD   L,A             o     111
2395:095B LD   A,00:0          >.     62   0
2397:095D SBC  A,H             .     156
2398:095E LD   H,A             g     103
2399:095F OR   A               .     183
2400:0960 RET                  .     201
2401:0961 LD   HL,00:0         !..    33   0   0
2404:0964 AND  37:55           .7    230  55       if 0965 SET CF
2406:0966 RET                  .     201

2407:0967 PUSH DE              .     213
2408:0968 EX   DE,HL           .     235
2409:0969 LD   HL,3F2F:16175   !/?   33  47  63   FPA2 exponent
2412:096C JP   C,0972:2418       .r.  218 114   9
2415:096F LD   HL,3F26:16166   !&?   33  38  63   FPA1 exponent
2418:0972 LD   A,D             z     122
2419:0973 OR   E               .     179
2420:0974 JP   Z,09A7:2471       ...  202 167   9
2423:0977 AND  D               .     162
2424:0978 JP   Z,09AA:2474       ...  202 170   9
2427:097B JP   P,098D:2445       ...  242 141   9
2430:097E EX   AF,AF'          .       8
2431:097F LD   A,E             {     123
2432:0980 NEG  A               .D    237  68
2434:0982 LD   E,A                    95
2435:0983 LD   A,D             z     122
2436:0984 CP   L,A             /      47
2437:0985 CPL   CF             ?      63
2438:0986 ADC A,00:0           ..    206   0
2440:0988 LD   D,A             W      87
2441:0989 JP   Z,09AB:2475       ...  202 171   9
```

```
2444:098C LD   A,08:8          >.    62   8
2446:098E LD   A,90:144        >.    62 144
2448:0990 DEC  A               =     61
2449:0991 SLA  E               .#   203  35
2451:0993 RL   D               ..   203  18
2453:0995 JP   P,0990:2448       ...  242 144   9
2456:0998 LD   (HL),A          w    119
2457:0999 DEC  HL              +     43
2458:099A EX   AF,AF'          .      8
2459:099B JP   M,09A0:2464       ...  250 160   9
2462:099E RES  7,D             ..   203 186
2464:09A0 LD   (HL),D          r    114
2465:09A1 DEC  HL              +     43
2466:09A2 LD   (HL),E          s    115
2467:09A3 DEC  HL              +     43
2468:09A4 XOR  A               .    175
2469:09A5 LD   (HL),A          w    119
2470:09A6 DEC  HL              +     43
2471:09A7 LD   (HL),A          w    119
2472:09A8 POP  DE              .    209
2473:09A9 RET                  .    201
2474:09AA EX   AF,AF'          .      8
2475:09AB LD   D,E             S     83
2476:09AC LD   E,00:0          ..    30   0
2478:09AE LD   A,D             z    122
2479:09AF OR   A               .    183
2480:09B0 LD   A,88:136        >.    62 136
2482:09B2 JP   P,0990:2448       ...  242 144   9
2485:09B5 JP   0998:2456       ...  195 152   9

Compare FPA1 and FPA2
2488:09B8 PUSH DE              .    213
2489:09B9 PUSH HL              .    229
2490:09BA LD   DE,(3F25:16165)  .[%?237  91  37  63 FPA1 MSB
2494:09BE LD   HL,(3F2E:16174)  *.?  42  46  63     FPA2 MSB
2497:09C1 LD   A,D             z    122            check exponents
2498:09C2 OR   A               .    183
2499:09C3 JP   Z,09FF:2559       ...  202 255   9
2502:09C6 LD   A,H             |    124
2503:09C7 OR   A               .    183
2504:09C8 JP   Z,0A0A:2570       ...  202  10  10
2507:09CB LD   A,L             }    125
2508:09CC XOR  E               .    171            check sign bits
2509:09CD JP   M,0A0A:2570       ...  250  10  10
2512:09D0 LD   A,D             z    122
2513:09D1 CP   H               .    188
2514:09D2 JP   NZ,09F3:2547      ...  194 243   9
2517:09D5 LD   A,E             {    123
2518:09D6 CP   L               .    189
2519:09D7 JP   NZ,09F3:2547      ...  194 243   9
2522:09DA LD   DE,3F24:16164   .$?   17  36  63   compare the 4 mantissa
2525:09DD LD   HL,3F2D:16173   !-?   33  45  63   bits from MSB to LSB
2528:09E0 LD   A,(DE)          .     26
2529:09E1 CP   (HL)            .    190
2530:09E2 JP   NZ,09F3:2547      ...  194 243   9
2533:09E5 DEC  DE              .     27
2534:09E6 DEC  HL              +     43
2535:09E7 LD   A,(DE)          .     26
2536:09E8 CP   (HL)            .    190
2537:09E9 JP   NZ,09F3:2547      ...  194 243   9
2540:09EC DEC  DE              .     27
2541:09ED DEC  HL              +     43
2542:09EE LD   A,(DE)          .     26
2543:09EF CP   (HL)            .    190
2544:09F0 JP   Z,0A0D:2573       ...  202  13  10
2547:09F3 POP  HL              .    225
2548:09F4 POP  DE              .    209
2549:09F5 LD   A,(3F25:16165)  :%?   58  37  63   FP operand
2552:09F8 RL   A               .     23
2553:09F9 ADC  A,00:0          ..   206   0
2555:09FB RR   A               .     31
2556:09FC RET  NZ              .    192
2557:09FD INC  A               <     60
2558:09FE RET                  .    201
```

```
2559:09FF OR   H                 .    180
2560:0A00 JP   Z,0A0D:2573         ...  202  13  10
2563:0A03 LD   A,L               }    125
2564:0A04 OR   A                 .    183
2565:0A05 JP   M,0A0D:2573         ...  250  13  10
2568:0A08 SET  7,E               ..   203 251
2570:0A0A SET  CF                7     55
2571:0A0B RL   E                 ..   203  19
2573:0A0D POP  HL                .    225
2574:0A0E POP  DE                .    209
2575:0A0F RET                    .    201

load FPA1 with ASCII number
2576:0A10 EXX                    .    217
2577:0A11 PUSH BC                .    197
2578:0A12 PUSH DE                .    213
2579:0A13 PUSH HL                .    229
2580:0A14 XOR  A                 .    175
2581:0A15 LD   (3F17:16151),A    2.?   50  23  63   set work space to 0
2584:0A18 LD   B,A               G     71
2585:0A19 LD   C,A               O     79
2586:0A1A LD   D,A               W     87
2587:0A1B LD   E,A               _     95
2588:0A1C EXX                    .    217
2589:0A1D LD   L,A               o    111
2590:0A1E LD   C,A               O     79
2591:0A1F LD   B,01:1            ..     6   1
2593:0A21 EX   AF,AF'            .      8
2594:0A22 LD   A,(DE)            .     26
2595:0A23 CP   2B:43             .+   254  43       check for leading sign
2597:0A25 JR   Z,0A2F:2607         (.    40   8
2599:0A27 SUB  2D:45             .-   214  45       check for minus sign
2601:0A29 JR   NZ,0A30:2608      .     32   5
2603:0A2B DEC  A                 =     61
2604:0A2C LD   (3F17:16151),A    2.?   50  23  63
2607:0A2F INC  DE                .     19
2608:0A30 LD   A,(DE)            .     26
2609:0A31 CP   45:69             .E   254  69       check for Exponent
2611:0A33 JR   Z,0A9F:2719         (j    40 106
2613:0A35 CP   65:101            .e   254 101       check for exponent
2615:0A37 JR   Z,0A9F:2719         (f    40 102
2617:0A39 CP   2E:46             ..   254  46       check for decimal
2619:0A3B JR   NZ,0A44:2628      .     32   7
2621:0A3D DEC  C                 .     13
2622:0A3E DEC  B                 .      5
2623:0A3F JR   Z,0A2F:2607         (.    40 238
2625:0A41 JP   0AD2:2770         ...  195 210  10
2628:0A44 SUB  30:48             .0   214  48
2630:0A46 CP   0A:10             ..   254  10       check for digit
2632:0A48 JP   NC,0AD2:2770        ...  210 210  10
2635:0A4B EX   AF,AF'            .      8
2636:0A4C ADD  A,C               .    129
2637:0A4D EX   AF,AF'            .      8
2638:0A4E LD   H,A               g    103
2639:0A4F LD   A,L               }    125
2640:0A50 AND  F0:240            ..   230 240
2642:0A52 JR   NZ,0A2F:2607      .     32 219
2644:0A54 EXX                    .    217
2645:0A55 CALL 0A94:2708         ...  205 148  10
2648:0A58 PUSH DE                .    213
2649:0A59 PUSH BC                .    197
2650:0A5A EXX                    .    217
2651:0A5B LD   A,L               }    125
2652:0A5C ADC A,A                .    143
2653:0A5D LD   L,A               o    111
2654:0A5E EXX                    .    217
2655:0A5F CALL 0A94:2708         ...  205 148  10
2658:0A62 EXX                    .    217
2659:0A63 ADC A,A                .    143
2660:0A64 EXX                    .    217
2661:0A65 CALL 0A94:2708         ...  205 148  10
2664:0A68 EXX                    .    217
2665:0A69 ADC A,A                .    143
2666:0A6A EXX                    .    217
2667:0A6B POP  HL                .    225
```

```
2668:0A6C ADD  HL,BC            .      9
2669:0A6D LD   B,H              D     68
2670:0A6E LD   C,L              M     77
2671:0A6F POP  HL               .    225
2672:0A70 ADC  HL,DE            .Z   237  90
2674:0A72 EX   DE,HL            .    235
2675:0A73 EXX                   .    217
2676:0A74 ADC A,L               .    141
2677:0A75 LD   L,A              o    111
2678:0A76 LD   A,H              |    124
2679:0A77 EXX                   .    217
2680:0A78 ADD  A,C              .    129
2681:0A79 LD   C,A              O     79
2682:0A7A JR   NC,0A87:2695       0.     48  11
2684:0A7C XOR  A                .    175
2685:0A7D INC  B                .      4
2686:0A7E JR   NZ,0A87:2695      .     32   7
2688:0A80 INC  E                .     28
2689:0A81 JR   NZ,0A87:2695      .     32   4
2691:0A83 INC  D                .     20
2692:0A84 JR   NZ,0A87:2695      .     32   1
2694:0A86 SET  CF               7     55
2695:0A87 EXX                   .    217
2696:0A88 LD   A,L              }    125
2697:0A89 ADC A,00:0            ..   206   0

Multiply BC and DE by 2
2699:0A8B LD   L,A              o    111
2700:0A8C AND  F0:240           ..   230 240
2702:0A8E JR   Z,0A2F:2607        (.     40 159
2704:0A90 INC  C                .     12
2705:0A91 JP   0A2F:2607        ./.   195  47  10
2708:0A94 LD   H,B              `     96
2709:0A95 LD   L,C              i    105
2710:0A96 ADD  HL,HL            )     41
2711:0A97 LD   B,H              D     68
2712:0A98 LD   C,L              M     77
2713:0A99 LD   H,D              b     98
2714:0A9A LD   L,E              k    107
2715:0A9B ADC  HL,HL            .j   237 106
2717:0A9D EX   DE,HL            .    235
2718:0A9E RET                   .    201

?????
2719:0A9F XOR  A                .    175
2720:0AA0 LD   B,A              G     71
2721:0AA1 LD   C,A              O     79
2722:0AA2 INC  DE               .     19
2723:0AA3 LD   A,(DE)           .     26
2724:0AA4 CP   2B:43            .+   254  43        positive?
2726:0AA6 JR   Z,0AAD:2733        (.     40   5
2728:0AA8 CP   2D:45            .-   254  45        negative?
2730:0AAA JR   NZ,0AAF:2735      .     32   3
2732:0AAC DEC  C                .     13
2733:0AAD INC  DE               .     19
2734:0AAE LD   A,(DE)           .     26
2735:0AAF SUB   30:48           .0   214  48
2737:0AB1 CP   0A:10            ..   254  10        check for digit
2739:0AB3 JR   NC,0AC5:2757       0.     48  16
2741:0AB5 LD   H,A              g    103
2742:0AB6 LD   A,B              x    120
2743:0AB7 CP   0C:12            ..   254  12
2745:0AB9 JR   NC,0B11:2833       0V     48  86
2747:0ABB RLC  A                .      7
2748:0ABC LD   B,A              G     71
2749:0ABD RLC  A                .      7
2750:0ABE RLC  A                .      7
2751:0ABF LD   B,B              .    128
2752:0AC0 ADD  A,H              .    132
2753:0AC1 LD   B,A              G     71
2754:0AC2 JP   0AAD:2733        ...   195 173  10
2757:0AC5 INC  C                .     12
2758:0AC6 JR   NZ,0ACC:2764      .     32   4
2760:0AC8 LD   A,B              x    120
2761:0AC9 NEG  A                .D   237  68
2763:0ACB LD   B,A              G     71
2764:0ACC EX   AF,AF'           .      8
2765:0ACD LD   B,B              .    128
2766:0ACE JP   PE,0B1A:2842       ...   234  26  11
```

```
2769:0AD1 EX   AF,AF'              .      8
2770:0AD2 LD   H,L                 e    101
2771:0AD3 EXX                      .    217
2772:0AD4 PUSH BC                  .    197
2773:0AD5 PUSH DE                  .    213
2774:0AD6 EXX                      .    217
2775:0AD7 POP  BC                  .    193
2776:0AD8 LD   L,B                 h    104
2777:0AD9 LD   B,C                 A     65
2778:0ADA LD   (3F22:16162),DE     .S"?237  83  34  63 FPA1 LSB
2782:0ADE POP  DE                  .    209
2783:0ADF LD   C,D                 J     74
2784:0AE0 LD   D,A8:168            ..    22 168
2786:0AE2 LD   (3F24:16164),HL     "$?  34  36  63   FPA1 MSB
2789:0AE5 CALL 0B0A:2826           ...  205  10  11
2792:0AE8 LD   HL,(3F22:16162)     *"?  42  34  63   FPA1 LSB
2795:0AEB PUSH HL                  .    229
2796:0AEC LD   HL,(3F24:16164)     *$?  42  36  63   FPA1 MSB
2799:0AEF CALL 06F8:1784           ...  205 248   6
2802:0AF2 EX   AF,AF'              .      8
2803:0AF3 ADD  A,26:38             .&   198  38
2805:0AF5 CP   4D:77               .M   254  77
2807:0AF7 JR   NC,0B22:2850          0)     48  41
2809:0AF9 LD   B,00:0              ..     6   0
2811:0AFB LD   C,A                 O     79
2812:0AFC LD   HL,0B3B:2875        !;.   33  59  11   start of table
2815:0AFF ADD  HL,BC               .      9
2816:0B00 ADD  HL,BC               .      9           5 bytes per exponent
2817:0B01 ADD  HL,BC               .      9
2818:0B02 ADD  HL,BC               .      9
2819:0B03 ADD  HL,BC               .      9
2820:0B04 CALL 075D:1885           .].  205  93   7   (HL)*FPA1
2823:0B07 POP  DE                  .    209
2824:0B08 OR   A                   .    183
2825:0B09 RET                      .    201
2826:0B0A POP  HL                  .    225
2827:0B0B EXX                      .    217
2828:0B0C POP  HL                  .    225
2829:0B0D POP  DE                  .    209
2830:0B0E POP  BC                  .    193
2831:0B0F EXX                      .    217
2832:0B10 JP   (HL)                .    233
2833:0B11 CALL 0B0A:2826           ...  205  10  11
2836:0B14 INC  C                   .     12
2837:0B15 JP   Z,070A:1802           ...  202  10   7
2840:0B18 SET  CF                  7     55
2841:0B19 RET                      .    201
2842:0B1A CALL 0B0A:2826           ...  205  10  11
2845:0B1D JP   P,070A:1802           ...  242  10   7
2848:0B20 JR   0B18:2840           ..    24 246
2850:0B22 CP   F5:245              ..   254 245
2852:0B24 JR   C,0B33:2867           8.     56  13
2854:0B26 ADD  A,0B:11             ..   198  11
2856:0B28 PUSH AF                  .    245
2857:0B29 LD   HL,0BC2:3010        !..   33 194  11   1 E -11
2860:0B2C CALL 075D:1885           .].  205  93   7   (HL)*FPA1
2863:0B2F POP  AF                  .    241
2864:0B30 JP   0AF9:2809           ...  195 249  10
2867:0B33 POP  DE                  .    209
2868:0B34 SUB   26:38              .&   214  38
2870:0B36 JP   M,070A:1802           ...  250  10   7
2873:0B39 JR   0B18:2840           ..    24 221
```

Stored in floating point, exponent numbers to speed up math routines

2875:0B3B	59	11	199	89	2	1E-38
2880:0B40	64	0	28	8	6	1E-37
2885:0B45	69	5	36	42	9	1E-36
2890:0B4A	74	10	173	84	12	1E-35
2895:0B4F	79	15	236	4	16	1E-34
2900:0B54	84	4	39	38	19	1E-33
2905:0B59	89	9	177	79	22	1E-32
2910:0B5E	94	14	206	1	26	1E-31
2915:0B63	99	3	66	34	29	1E-30
2920:0B68	104	8	210	74	32	1E-29
2925:0B6D	109	13	135	125	35	1E-28
2930:0B72	114	2	116	30	39	1E-27
2935:0B77	119	7	18	70	42	1E-26
2940:0B7C	124	12	150	119	45	1E-25
2945:0B81	129	1	190	26	49	1E-24
2950:0B86	134	6	109	65	52	1E-23
2955:0B8B	139	11	201	113	55	1E-22
2960:0B90	144	0	29	23	59	1E-21
2965:0B95	149	5	229	60	62	1E-20
2970:0B9A	154	10	30	108	65	1E-19
2975:0B9F	159	15	146	19	69	1E-18
2980:0BA4	164	4	119	56	72	1E-17
2985:0BA9	169	9	149	102	75	1E-16
2990:0BAE	174	14	29	16	79	1E-15
2995:0BB3	179	3	36	52	82	1E-14
3000:0BB8	184	8	46	97	85	1E-13
3005:0BBD	189	13	188	12	89	1E-12
3010:0BC2	194	2	235	47	92	1E-11
3015:0BC7	199	7	230	91	95	1E-10
3020:0BCC	204	12	112	9	99	1E-9
3025:0BD1	209	1	204	43	102	1E-8
3030:0BD6	214	6	191	86	105	1E-7
3035:0BDB	219	11	55	6	109	1E-6
3040:0BE0	224	0	197	39	112	1E-5
3045:0BE5	229	5	183	81	115	1E-4
3050:0BEA	234	10	18	3	119	1E-3
3055:0BEF	239	15	215	35	122	1E-2
3060:0BF4	244	4	204	76	125	1E-1
3065:0BF9	249	9	0	0	129	1E0
3070:0BFE	254	14	0	32	132	1E1
3075:0C03	3	3	0	72	135	1E2
3080:0C08	8	8	0	122	138	1E3
3085:0C0D	13	13	64	28	142	1E4
3090:0C12	18	2	80	67	145	1E5
3095:0C17	23	7	36	116	148	1E6
3100:0C1C	28	12	150	24	152	1E7
3105:0C21	33	1	188	62	155	1E8
3110:0C26	38	6	107	110	158	1E9
3115:0C2B	43	11	2	21	162	1E10
3120:0C30	48	0	67	58	165	1E11
3125:0C35	53	5	212	104	168	1E12
3130:0C3A	58	10	132	17	172	1E13
3135:0C3F	63	15	230	53	175	1E14
3140:0C44	68	4	95	99	178	1E15
3145:0C49	73	9	27	14	182	1E16
3150:0C4E	78	14	162	49	185	1E17
3155:0C53	83	3	11	94	188	1E18
3160:0C58	88	8	199	10	192	1E19
3165:0C5D	93	13	120	45	195	1E20
3170:0C62	98	2	215	88	198	1E21
3175:0C67	103	7	134	7	202	1E22
3180:0C6C	108	12	104	41	205	1E23
3185:0C71	113	1	194	83	208	1E24
3190:0C76	118	6	89	4	212	1E25
3195:0C7B	123	11	111	37	215	1E26
3200:0C80	128	0	203	78	218	1E27
3205:0C85	133	5	63	1	222	1E28
3210:0C8A	138	10	143	33	225	1E29
3215:0C8F	143	15	242	73	228	1E30
3220:0C94	148	4	111	124	231	1E31
3225:0C99	153	9	197	29	235	1E32
3230:0C9E	158	14	55	69	238	1E33
3235:0CA3	163	3	132	118	241	1E34
3240:0CA8	168	8	19	26	245	1E35
3245:0CAD	173	13	151	64	248	1E36
3250:0CB2	178	2	189	112	251	1E37
3255:0CB7	183	7	118	22	255	1E38

```
3260:0CBC LD   HL,3F77:16247  !w?   33 119  63    buffer
3263:0CBF LD   A,(3F26:16166) :&?   58  38  63    exponent
3266:0CC2 OR   A              .     183
3267:0CC3 JR   NZ,0CCB:3275         32        6
3269:0CC5 LD   (HL),0030:48   60    54  48        write 0 to buffer
3271:0CC7 DEC  HL             +     43
3272:0CC8 LD   (HL),0001:1    6.    54   1        length to print
3274:0CCA RET                 .     201
3275:0CCB LD   A,(3F25:16165) :%?   58  37  63    most significant byte
3278:0CCE OR   A              .     183
3279:0CCF JP   P,0CDA:3290          242 218  12
3282:0CD2 LD   (HL),002D:45   6-    54  45        write minus sign
3284:0CD4 INC  HL             #     35
3285:0CD5 AND  007F:127       ..    230 127       strip high bit
3287:0CD7 LD   (3F25:16165),A 2%?   50  37  63    write it back
3290:0CDA PUSH HL             .     229
3291:0CDB LD   E,0026:38      .&    30  38
3293:0CDD LD   A,(3F26:16166) :&?   58  38  63    exponent
3296:0CE0 SUB   0003:3        ..    214   3
3298:0CE2 INC  A              <     60
3299:0CE3 JR   NC,0CEF:3311     0.      48        10
3301:0CE5 LD   HL,0BFE:3070   !..   33 254  11    1 E +1
3304:0CE8 CALL 075D:1885      .].   205  93   7    (HL)*FPA1
3307:0CEB LD   E,0025:37      .%    30  37
3309:0CED JR   0CDD:3293      ..    24 238
3311:0CEF LD   D,A            W     87
3312:0CF0 LD   BC,0005:5      ...    1   5   0    length of entry
3315:0CF3 LD   HL,0B3F:2879   !?.   33  63  11    start at 1 E -38
3318:0CF6 ADD  HL,BC          .      9
3319:0CF7 DEC  E              .     29
3320:0CF8 LD   A,(HL)         .     126
3321:0CF9 ADD  A,D            .     130
3322:0CFA JR   NC,0CF6:3318     0.    48 250      until match found
3324:0CFC LD   A,E            {     123
3325:0CFD LD   (3F1C:16156),A 2.?   50  28  63    work space
3328:0D00 LD   DE,3F2F:16175  ./?   17  47  63    FPA2
3331:0D03 LDDR                .;    237 184       move number in
3333:0D05 CALL 0760:1888      .'.   205  96   7    FPA1*FPA2
3336:0D08 CALL 0DFC:3580      ...   205 252  13    FPA1/10
3339:0D0B LD   A,(3F32:16178) :2?   58  50  63    significant digits output
3342:0D0E LD   E,A            _     95
3343:0D0F SLA  A              .'    203  39       *2
3345:0D11 SLA  A              .'    203  39       *4
3347:0D13 ADD  A,E            .     131           *5
3348:0D14 LD   E,A            _      95
3349:0D15 LD   D,0000:0       ..     22   0
3351:0D17 LD   HL,0BFE:3070   !..   33 254  11    1 E +1
3354:0D1A SBC  HL,DE          .R    237  82
3356:0D1C CALL 1117:4375      ...   205  23  17    move (HL) to FPA2
3359:0D1F DEC  DE             .      27
3360:0D20 EX   DE,HL          .     235
3361:0D21 DEC  (HL)           5      53
3362:0D22 CALL 062F:1583      ./.   205  47   6    FPA1+FPA2
3365:0D25 CALL 0DFC:3580      ...   205 252  13    FPA1/10
3368:0D28 LD   A,(3F26:16166)        58  38  63    FPA1
3371:0D2B LD   BC,(3F22:16162)       237  75  34  63
3375:0D2F LD   DE,(3F24:16164)       237  91  36  63
3379:0D33 SET  7,D            ..    203 250       make sure bit is on
3381:0D35 SUB   0084:132      ..    214 132
3383:0D37 JR   Z,0D44:3396      (.     40        11
3385:0D39 SRL  D              .:    203  58
3387:0D3B RR   E              ..    203  27
3389:0D3D RR   B              ..    203  24
3391:0D3F RR   C              ..    203  25
3393:0D41 INC  A              <      60           place decimal
3394:0D42 JR   NZ,0D39:3385         32 245
3396:0D44 LD   A,(3F32:16178) :2?   58  50  63    significant digits output
3399:0D47 POP  HL             .     225
3400:0D48 PUSH AF             .     245
3401:0D49 LD   A,D            z     122
3402:0D4A RR   A              .      31
3403:0D4B RR   A              .      31
3404:0D4C RR   A              .      31
3405:0D4D RR   A              .      31
3406:0D4E AND  000F:15        ..    230   15       get digit
3408:0D50 ADD  A,0030:48       .0   198  48       ADD  ASCII offset
3410:0D52 LD   (HL),A         w     119
```

```
3411:0D53 INC  HL              #     35
3412:0D54 LD   A,D             z    122
3413:0D55 AND  000F:15        ..    230   15          get second half
3415:0D57 LD   D,A             W     87
3416:0D58 POP  AF              .    241               significant digits
3417:0D59 DEC  A               =     61
3418:0D5A JR   Z,0D7B:3451       (.     40         31          zero fill rest

Multiply BC and DE by 10
3420:0D5C PUSH HL              .    229
3421:0D5D LD   H,B             ;     96
3422:0D5E LD   L,C             i    105
3423:0D5F ADD  HL,HL           )     41
3424:0D60 LD   B,H             D     68                BC*2
3425:0D61 LD   C,L             M     77
3426:0D62 EX   DE,HL           .    235
3427:0D63 ADC  HL,HL          .j    237 106
3429:0D65 PUSH HL              .    229                DE*2
3430:0D66 EX   DE,HL           .    235
3431:0D67 ADD  HL,HL           )     41                BC*4
3432:0D68 EX   DE,HL           .    235
3433:0D69 ADC  HL,HL          .j    237 106           DE*4
3435:0D6B EX   DE,HL           .    235
3436:0D6C ADD  HL,HL           )     41                BC*8
3437:0D6D EX   DE,HL           .    235
3438:0D6E ADC  HL,HL          .j    237 106           DE*8
3440:0D70 EX   DE,HL           .    235
3441:0D71 ADD  HL,BC           .      9               DE*10
3442:0D72 LD   B,H             D     68
3443:0D73 LD   C,L             M     77
3444:0D74 POP  HL              .    225
3445:0D75 ADC  HL,DE          .Z    237  90           BC*10
3447:0D77 EX   DE,HL           .    235
3448:0D78 JP   0D47:3399      .G.   195  71  13       write next digit
3451:0D7B PUSH AF              .    245
3452:0D7C LD   A,(3F32:16178) :2?    58  50  63       significant digits output
3455:0D7F LD   B,A             G     71               zero fill rest
3456:0D80 DEC  HL              +     43
3457:0D81 LD   A,(HL)          .    126
3458:0D82 CP   0030:48        .0    254   48
3460:0D84 JR   NZ,0D8A:3466      .     32          4
3462:0D86 DJNZ 0D80:3456       ..     16 248
3464:0D88 DEC  HL              +     43
3465:0D89 LD   A,(HL)          .    126
3466:0D8A PUSH AF              .    245
3467:0D8B DJNZ 0D88:3464       ..     16 251
3469:0D8D LD   A,(3F1C:16156) :.?    58  28  63       ?????
3472:0D90 LD   C,0000:0        ..     14   0
3474:0D92 ADD  A,0002:2        ..    198   2
3476:0D94 CP   000B:11        ..    254  11
3478:0D96 JR   C,0D9B:3483       8.     56          3
3480:0D98 INC  C               .     12
3481:0D99 LD   A,0002:2        >.     62   2
3483:0D9B SUB   0002:2         ..    214   2
3485:0D9D JP   P,0DB4:3508       ...   242 180  13
3488:0DA0 LD   (HL),002E:46    6.     54  46
3490:0DA2 NEG  A               .D    237  68
3492:0DA4 LD   B,A             G     71
3493:0DA5 JR   0DAA:3498       ..     24   3
3495:0DA7 INC  HL              #     35
3496:0DA8 LD   (HL),0030:48    60     54  48          Write a zero
3498:0DAA DJNZ 0DA7:3495       ..     16 251
3500:0DAC POP  AF              .    241
3501:0DAD INC  HL              #     35
3502:0DAE LD   (HL),A          w    119
3503:0DAF JR   NZ,0DAC:3500      .     32 251
3505:0DB1 JP   0DCA:3530       ...   195 202  13
3508:0DB4 LD   B,A             G     71
3509:0DB5 INC  B               .      4
3510:0DB6 POP  AF              .    241
3511:0DB7 JR   Z,0DC5:3525       (.     40         12
3513:0DB9 LD   (HL),A          w    119
3514:0DBA INC  HL              #     35
3515:0DBB DJNZ 0DB6:3510       ..     16 249
3517:0DBD LD   (HL),002E:46    6.     54  46
3519:0DBF POP  AF              .    241
3520:0DC0 JR   NZ,0DAD:3501      .     32 235
3522:0DC2 JP   0DCA:3530       ...   195 202  13
```

```
3525:0DC5 LD   (HL),0030:48   60     54  48        Write a zero
3527:0DC7 INC  HL             #      35
3528:0DC8 DJNZ 0DC5:3525      ..     16 251
3530:0DCA DEC  C              .      13
3531:0DCB JR   NZ,0DF2:3570       %     32       37
3533:0DCD LD   (HL),0045:69   6E     54  69
3535:0DCF INC  HL             #      35
3536:0DD0 LD   A,(3F1C:16156) :.?    58  28  63
3539:0DD3 OR   A              .     183
3540:0DD4 LD   B,002B:43      .+      6  43
3542:0DD6 JP   P,0DDD:3549      ...   242 221  13
3545:0DD9 LD   B,002D:45      .-      6  45
3547:0DDB NEG  A              .D    237  68
3549:0DDD LD   (HL),B         p     112
3550:0DDE INC  HL             #      35
3551:0DDF LD   B,002F:47      ./      6  47
3553:0DE1 INC  B              .       4
3554:0DE2 SUB   000A:10       ..    214  10
3556:0DE4 JP   P,0DE1:3553      ...   242 225  13
3559:0DE7 LD   (HL),B         p     112
3560:0DE8 INC  HL             #      35
3561:0DE9 LD   B,003A:58      .:      6  58
3563:0DEB DEC  B              .       5
3564:0DEC INC  A              <      60
3565:0DED JP   M,0DEB:3563      ...   250 235  13
3568:0DF0 LD   (HL),B         p     112
3569:0DF1 INC  HL             #      35
3570:0DF2 DEC  HL             +      43
3571:0DF3 LD   DE,3F76:16246  .v?    17 118  63
3574:0DF6 OR   A              .     183
3575:0DF7 SBC  HL,DE          .R    237  82
3577:0DF9 LD   A,L            }     125
3578:0DFA LD   (DE),A         .      18
3579:0DFB RET                 .     201

Divide FPA1 by 10 if > 10
3580:0DFC LD   A,(3F26:16166) :&?    58  38  63    exponent
3583:0DFF CP   0084:132       ..    254 132
3585:0E01 RET  C              .     216
3586:0E02 JR   NZ,0E0A:3594       .     32       6
3588:0E04 LD   A,(3F25:16165) :%?    58  37  63    MSB
3591:0E07 CP   0020:32        .     254  32
3593:0E09 RET  C              .     216
3594:0E0A LD   HL,3F1C:16156  !.?    33  28  63
3597:0E0D INC  (HL)           4      52
3598:0E0E LD   HL,0BF4:3060   !..    33 244  11    1 E -1
3601:0E11 JP   075D:1885      .].    195  93   7    (HL)*FPA1

LOG
3604:0E14 JP   NZ,1F03:7939       ...   194       3  31    overflow error
3607:0E17 LD   HL,(3F25:16165) *%?    42  37  63    most significant byte
3610:0E1A LD   A,H            |      124
3611:0E1B OR   A              .     183
3612:0E1C JP   Z,1F00:7936      ...   202       0  31    if zero illegal quantity
3615:0E1F BIT  7,L            .}    203 125
3617:0E21 JP   NZ,1F00:7936     ..    194       0  31    if negative -illegal
3620:0E24 PUSH AF             .     245
3621:0E25 LD   A,0080:128     >.     62 128
3623:0E27 LD   (3F26:16166),A 2&?    50  38  63    exponent
3626:0E2A LD   HL,11BF:4543   !..    33 191  17    location of constant
3629:0E2D CALL 0621:1569      .!.   205  33   6    add (HL) to FPA1
3632:0E30 CALL 112C:4396      .,.   205  44  17    push FPA1
3635:0E33 LD   HL,11BA:4538   !..    33 186  17    constant
3638:0E36 CALL 0621:1569      .!.   205  33   6    add it
3641:0E39 CALL 1159:4441      .Y.   205  89  17    pop FPA2
3644:0E3C CALL 07E2:2018      ...   205 226   7    FPA1/FPA2
3647:0E3F LD   HL,1243:4675   !C.    33  67  18    3 constants for LOG
3650:0E42 CALL 103C:4156      .<.   205  60  16    do power series
3653:0E45 LD   HL,11B5:4533   !..    33 181  17    constant
3656:0E48 CALL 0621:1569      .!.   205  33   6    add (HL to FPA1
3659:0E4B POP  AF             .     241
3660:0E4C SUB   0080:128      ..    214 128
3662:0E4E LD   L,A            o     111
3663:0E4F SBC  A,A            .     159
3664:0E50 LD   H,A            g     103
3665:0E51 SET  CF             7      55
```

```
3666:0E52 CALL 0967:2407      .g.  205 103   9    ?????
3669:0E55 CALL 062F:1583      ./.  205  47   6    FPA1+FPA2
3672:0E58 LD   HL,1192:4498   !..   33 146  17    LOG(2)
3675:0E5B JP   075D:1885      .].  195  93   7    (HL)*FPA1

SQR done by e^(log(x)/2)
3678:0E5E JP   NZ,1F03:7939       ...  194        3  31    type mismatch
3681:0E61 LD   A,(3F26:16166) :&?   58  38  63    exponent
3684:0E64 OR   A              .    183
3685:0E65 RET  Z              .    200
3686:0E66 CALL 0E17:3607      ...  205  23  14    get LOG
3689:0E69 LD   HL,3F26:16166  !&?   33  38  63    divide by 2
3692:0E6C DEC  (HL)           5    53
3693:0E6D JP   0EEB:3819      ...  195 235  14    do exponent

FPA1 ^ FPA2
3696:0E70 LD   A,(3F2F:16175) :/?   58  47  63    FPA2 exponent
3699:0E73 OR   A              .    183
3700:0E74 JP   Z,1120:4384        . .  202       32  17    load FPA1 with 1 (x^0=1)
3703:0E77 LD   A,(3F26:16166) :&?   58  38  63    FPA1 exponent
3706:0E7A OR   A              .    183
3707:0E7B RET  Z              .    200
3708:0E7C LD   DE,0EDC:3804   ...   17 220  14
3711:0E7F CALL 1182:4482      ...  205 130  17    make FPA1 positive
3714:0E82 LD   HL,3F2E:16174  !.?   33  46  63
3717:0E85 BIT  7,(HL)         ..   203 126        FPA2 sign bit
3719:0E87 JR   Z,0E8F:3727      (.    40         6
3721:0E89 RES  7,(HL)         ..   203 190
3723:0E8B LD   HL,0F3C:3900   !<.   33  60  15
3726:0E8E PUSH HL             .    229
3727:0E8F CALL 112C:4396      .,.  205  44  17    push FPA1
3730:0E92 CALL 0728:1832      .(.  205  40   7    move FPA2 to FPA1
3733:0E95 CALL 29B3:10675     ..)  205 179  41    do INT
3736:0E98 AND  0001:1         ..   230   1
3738:0E9A LD   (3F42:16194),A 2B?   50  66  63    storage
3741:0E9D CALL 09B8:2488      ...  205 184   9    compare FPA1 and FPA2
3744:0EA0 JR   Z,0EAA:3754      (.    40         8
3746:0EA2 LD   A,00FF:255     >.    62 255
3748:0EA4 LD   (3F42:16194),A 2B?   50  66  63
3751:0EA7 JP   0ECA:3786      ...  195 202  14
3754:0EAA CALL 1168:4456      .h.  205 104  17
3757:0EAD JR   C,0ECA:3786      8.    56        27
3759:0EAF CP   001A:26        ..   254  26
3761:0EB1 JR   NC,0ECA:3786       0.    48       23
3763:0EB3 CALL 113B:4411      .;.  205  59  17    pop FPA1
3766:0EB6 CALL 1104:4356      ...  205   4  17    move FPA1 to FPA2
3769:0EB9 PUSH AF             .    245
3770:0EBA JP   0EC6:3782      ...  195 198  14
3773:0EBD PUSH BC             .    197
3774:0EBE LD   HL,3F2E:16174  !.?   33  46  63    FPA2 MSB
3777:0EC1 RES  7,(HL)         ..   203 190        sign bit
3779:0EC3 CALL 0760:1888      .'.  205  96   7    FPA1*FPA2
3782:0EC6 POP  BC             .    193
3783:0EC7 DJNZ 0EBD:3773      ..    16 244        do power
3785:0EC9 RET                 .    201
3786:0ECA CALL 113B:4411      .;.  205  59  17    pop FPA1
3789:0ECD CALL 114A:4426      .J.  205  74  17    push FPA2
3792:0ED0 CALL 0E17:3607      ...  205  23  14    do LOG
3795:0ED3 CALL 1159:4441      .Y.  205  89  17    pop FPA2
3798:0ED6 CALL 0760:1888      .'.  205  96   7    FPA1*FPA2
3801:0ED9 JP   0EEB:3819      ...  195 235  14    do EXPONENT
3804:0EDC LD   A,(3F42:16194) :B?   58  66  63    from storage
3807:0EDF SRA  A              ./   203  47
3809:0EE1 JP   M,1F00:7936    ...  250   0  31    illegal quantity
3812:0EE4 RET  NC             .    208
3813:0EE5 JP   090E:2318      ...  195  14   9    exit
3816:0EE8 JP   NZ,1F03:7939       ...  194        3  31
3819:0EEB LD   HL,1197:4503   !..   33 151  17    1/log(2)
3822:0EEE CALL 075D:1885      .].  205  93   7    (HL)*FPA1
3825:0EF1 CALL 1104:4356      ...  205   4  17    move FPA1 to FPA2
3828:0EF4 CALL 1168:4456      .h.  205 104  17    check range
3831:0EF7 JR   C,0F32:3890      89    56        57
3833:0EF9 OR   A              .    183
3834:0EFA JP   P,0F08:3848    ...  242   8  15
3837:0EFD NEG  A              .D   237  68
3839:0EFF LD   HL,0F3C:3900   !<.   33  60  15
```

```
3842:0F02 PUSH HL                .     229                save return address?
3843:0F03 LD   HL,3F2E:16174     !.?    33  46  63       FPA2 MSB
3846:0F06 RES  7,(HL)            ..    203 190           set top mantissa bit
3848:0F08 PUSH AF                .     245
3849:0F09 NEG  A                 .D    237  68
3851:0F0B LD   L,A               o     111
3852:0F0C SBC  A,A               .     159
3853:0F0D LD   H,A               g     103
3854:0F0E XOR  A                 .     175
3855:0F0F CALL 0967:2407         .g.   205 103   9       convert integer to FPA
3858:0F12 CALL 062F:1583         ./.   205  47   6       FPA1+FPA2
3861:0F15 LD   HL,11EB:4587      !..    33 235  17       8 constants for EXP
3864:0F18 CALL 10B6:4278         ...   205 182  16       do power series
3867:0F1B LD   A,(3F26:16166)    :&?    58  38  63       exponent
3870:0F1E POP  BC                .     193
3871:0F1F LD   B,B               .     128
3872:0F20 JP   NC,070B:1803        ...  210        11   7    exit
3875:0F23 POP  HL                .     225
3876:0F24 LD   DE,0F3C:3900      .<.    17  60  15
3879:0F27 PUSH HL                .     229
3880:0F28 OR   A                 .     183
3881:0F29 SBC  HL,DE             .R    237  82           compare return address?
3883:0F2B POP  HL                .     225
3884:0F2C JP   NZ,1EF1:7921        ...  194 241  30          overflow
3887:0F2F JP   070A:1802         ...   195  10   7    exit

error checks
3890:0F32 LD   A,(3F2E:16174)    :.?    58  46  63       FPA2 MSB
3893:0F35 OR   A                 .     183
3894:0F36 JP   M,070A:1802         ...  250        10   7    exit
3897:0F39 JP   1F00:7936         ...   195   0  31       illegal quantity

TAKE INVERSE
3900:0F3C CALL 1104:4356         ...   205   4  17       move FPA1 to FPA2
3903:0F3F LD   HL,0BF9:3065      !..    33 249  11       FP one....not required!!!
3906:0F42 CALL 1123:4387         .#.   205  35  17       load FPA1 with one
3909:0F45 JP   07E2:2018         ...   195 226   7       FPA1/FPA2

TAN = SIN/COS
3912:0F48 JP   NZ,1F03:7939        ...  194          3  31
3915:0F4B LD   A,(3F25:16165)    :%?    58  37  63       FPA1 MSB
3918:0F4E RL   A                 .      23
3919:0F4F LD   A,0080:128        >.     62 128
3921:0F51 RR   A                 .      31
3922:0F52 CALL 0F7C:3964         .|.   205 124  15       do SIN(x+pi/2)
3925:0F55 CALL 112C:4396         .,.   205  44  17       push FPA1
3928:0F58 LD   BC,(3F33:16179)   .K3?  237  75  51  63
3932:0F5C LD   DE,(3F35:16181)   .[5?  237  91  53  63
3936:0F60 XOR  A                 .     175
3937:0F61 CALL 0FD3:4051         ...   205 211  15       do SIN
3940:0F64 CALL 1159:4441         .Y.   205  89  17       pop FPA2
3943:0F67 JP   07E2:2018         ...   195 226   7       FPA1/FPA2

COS = SIN(x+pi/2)
3946:0F6A JP   NZ,1F03:7939        ...  194          3  31
3949:0F6D LD   A,0040:64         >@     62  64
3951:0F6F JP   0F7C:3964         .|.   195 124  15       so SIN
SIN  this is the big one
3954:0F72 JP   NZ,1F03:7939        ...  194          3  31
3957:0F75 LD   A,(3F25:16165)    :%?    58  37  63       FPA1 MSB
3960:0F78 RL   A                 .      23
3961:0F79 LD   A,0000:0          >.     62   0          drop pi multiples
3963:0F7B RR   A                 .      31
3964:0F7C LD   HL,3F25:16165     !%?    33  37  63
3967:0F7F RES  7,(HL)            ..    203 190           put top bit on..WASTE!!!!
3969:0F81 LD   (3F42:16194),A    2B?    50  66  63       control byte
3972:0F84 LD   HL,11A1:4513      !..    33 161  17       2/pi
3975:0F87 CALL 075D:1885         .].   205  93   7       (HL)*FPA1
3978:0F8A LD   BC,(3F22:16162)         237  75  34  63 FPA1 mantissa
3982:0F8E LD   DE,(3F24:16164)         237  91  36  63
3986:0F92 SET  7,D               ..    203 250           set top bit
3988:0F94 LD   A,(3F26:16166)    :&?    58  38  63       exponent
3991:0F97 SUB   0082:130         ..    214 130
3993:0F99 JR   C,0FAA:4010         8.     56          15
3995:0F9B JR   Z,0FC8:4040         (+     40          43
3997:0F9D CP   0020:32           .    254  32
3999:0F9F JR   C,0FBD:4029         8.     56          28
```

```
4001:0FA1 LD   BC,0000:0       ...    1   0   0
4004:0FA4 LD   DE,0000:0       ...   17   0   0
4007:0FA7 JP   0FC8:4040       ...  195 200  15
4010:0FAA CP   00E1:225        ..   254 225
4012:0FAC JR   C,0FA1:4001       8.     56 243
4014:0FAE SRL  D                .:    203  58
4016:0FB0 RR   E                ..    203  27
4018:0FB2 RR   B                ..    203  24
4020:0FB4 RR   C                ..    203  25
4022:0FB6 INC  A                <      60
4023:0FB7 JP   NZ,0FAE:4014        ...  194 174  15
4026:0FBA JP   0FC8:4040       ...  195 200  15
4029:0FBD SLA  C                .!    203  33
4031:0FBF RL   B                ..    203  16
4033:0FC1 RL   E                ..    203  19
4035:0FC3 RL   D                ..    203  18
4037:0FC5 DEC  A                =      61
4038:0FC6 JR   NZ,0FBD:4029       .     32 245
4040:0FC8 LD   (3F33:16179),BC     237  67  51  63 save in workspace
4044:0FCC LD   (3F35:16181),DE     237  83  53  63
4048:0FD0 LD   A,(3F42:16194) :B?   58  66  63     control byte
4051:0FD3 ADD  A,D              .    130
4052:0FD4 LD   D,A              W     87
4053:0FD5 JP   P,0FDE:4062         ...  242 222  15
4056:0FD8 LD   HL,090E:2318    !..   33  14   9
4059:0FDB PUSH HL               .    229              save return address?
4060:0FDC RES  7,D              ..   203 186
4062:0FDE BIT  6,D              .r   203 114
4064:0FE0 JR   Z,0FF0:4080       (.     40         14
4066:0FE2 LD   HL,0000:0       !..   33   0   0
4069:0FE5 XOR  A                .    175
4070:0FE6 SBC  HL,BC            .B   237  66
4072:0FE8 LD   B,H              D     68
4073:0FE9 LD   C,L              M     77
4074:0FEA LD   HL,8000:32768   !..   33   0 128
4077:0FED SBC  HL,DE            .R   237  82
4079:0FEF EX   DE,HL            .    235
4080:0FF0 LD   HL,(11C6:4550) *..   42 198  17    constant
4083:0FF3 OR   A                .    183
4084:0FF4 SBC  HL,DE            .R   237  82
4086:0FF6 JR   C,1006:4102       8.     56         14
4088:0FF8 JR   NZ,0FFF:4095       .     32          5
4090:0FFA LD   HL,(11C4:4548) *..   42 196  17    constant
4093:0FFD SBC  HL,BC            .B   237  66
4095:0FFF LD   HL,1214:4628    !..   33  20  18    4 constants for SIN
4098:1002 PUSH AF               .    245
4099:1003 JR   NC,1018:4120      0.     48         19
4101:1005 LD   H,00F5:245      &.     38 245
4102:1006 PUSH AF only if JP  1006      245
4103:1007 LD   HL,0000:0       !..   33   0   0
4106:100A OR   A                .    183
4107:100B SBC  HL,BC            .B   237  66
4109:100D LD   B,H              D     68
4110:100E LD   C,L              M     77
4111:100F LD   HL,4000:16384   !.@   33   0  64
4114:1012 SBC  HL,DE            .R   237  82
4116:1014 EX   DE,HL            .    235
4117:1015 LD   HL,1229:4649    !).   33  41  18    5 constants for SIN
4120:1018 LD   A,B              x    120
4121:1019 OR   C                .    177
4122:101A OR   D                .    178
4123:101B OR   E                .    179
4124:101C JR   Z,102E:4142       (.     40         16
4126:101E LD   A,0082:130      >.     62 130
4128:1020 DEC  A                =      61
4129:1021 SLA  C                .!    203  33
4131:1023 RL   B                ..    203  16
4133:1025 RL   E                ..    203  19
4135:1027 RL   D                ..    203  18
4137:1029 JP   P,1020:4128       . .   242         32  16
4140:102C RES  7,D              ..   203 186
4142:102E LD   (3F26:16166),A 2&?   50  38  63    store in FPA1
4145:1031 LD   (3F22:16162),BC     237  67  34  63
4149:1035 LD   (3F24:16164),DE     237  83  36  63
4153:1039 POP  AF               .    241
4154:103A JR   C,10AE:4270       8r     56 114      do power series
```

```
POWER SERIES CALCULATOR
4156:103C LD   (3F37:16183),HL "7?   34  55  63   save table address
4159:103F CALL 1104:4356       ...  205   4  17   move FPA1 to FPA2
4162:1042 CALL 112C:4396       .¡.  205  44  17   push FPA1
4165:1045 CALL 0760:1888       .'.  205  96   7   (HL)*FPA1
4168:1048 LD   HL,(3F37:16183) *7?   42  55  63   get address
4171:104B CALL 10B6:4278       ...  205 182  16   move next number in
4174:104E CALL 1159:4441       .Y.  205  89  17   pop FPA2
4177:1051 JP   0760:1888       .'.  195  96   7   (HL)*FPA1

ATN
4180:1054 JP   NZ,1F03:7939        ...  194          3  31
4183:1057 CALL 117F:4479       ...  205 127  17   make FPA1 positive
4186:105A LD   HL,11AB:4523    !..   33 171  17   FP constant
4189:105D CALL 1117:4375       ...  205  23  17   load FPA2, (HL)
4192:1060 CALL 09B8:2488       ...  205 184   9   compare FPA1 and FPA2
4195:1063 JR   C,1099:4249       84     56         52
4197:1065 LD   HL,11B0:4528    !..   33 176  17   FP constant
4200:1068 CALL 1117:4375       ...  205  23  17   move it
4203:106B CALL 09B8:2488       ...  205 184   9   compare
4206:106E JR   C,107A:4218       8.     56         10
4208:1070 CALL 0F3C:3900       .<.  205  60  15   take inverse
4211:1073 LD   HL,109F:4255    !..   33 159  16
4214:1076 PUSH HL              .    229           save return address?
4215:1077 JP   1099:4249       ...  195 153  16
4218:107A LD   HL,0BF9:3065    !..   33 249  11
4221:107D CALL 0621:1569       .!.  205  33   6
4224:1080 CALL 0F3C:3900       .<.  205  60  15
4227:1083 LD   HL,(3F25:16165) *%?   42  37  63
4230:1086 SET  7,L             ..   203 253
4232:1088 INC  H               $     36
4233:1089 LD   (3F25:16165),HL "%?   34  37  63
4236:108C LD   HL,0BF9:3065    !..   33 249  11   FP number 1
4239:108F CALL 0621:1569       .!.  205  33   6   FPA1+(HL)
4242:1092 LD   HL,10A8:4264    !..   33 168  16
4245:1095 PUSH HL              .    229           save return address?
4246:1096 CALL 117F:4479       ...  205 127  17   make FPA1 positive
4249:1099 LD   HL,11CC:4556    !..   33 204  17   6 constants for ATAN
4252:109C JP   103C:4156       .<.  195  60  16   do power series

ADJUSTMENTS
4255:109F CALL 090E:2318       ...  205  14   9   negate FPA1
4258:10A2 LD   HL,119C:4508    !..   33 156  17   pi/2
4261:10A5 JP   0621:1569       .!.  195  33   6   FPA1+(HL)
4264:10A8 LD   HL,11A6:4518    !..   33 166  17   pi/4
4267:10AB JP   0621:1569       .!.  195  33   6   FPA1+(HL)

POWER SERIES CALCULATOR
4270:10AE PUSH HL              .    229
4271:10AF CALL 1104:4356       ...  205   4  17   move FPA1 to FPA2
4274:10B2 CALL 0760:1888       .'.  205  96   7   FPA1*FPA2
4277:10B5 POP  HL              .    225
4278:10B6 LD   A,(HL)          .    126
4279:10B7 PUSH AF              .    245           save count, pop into BC
4280:10B8 INC  HL              #     35
4281:10B9 LD   BC,0005:5       ...    1   5   0
4284:10BC LD   DE,3F2B:16171   .+?   17  43  63
4287:10BF LDIR                 ..   237 176       move next number in
4289:10C1 LD   (3F37:16183),HL "7?   34  55  63   save address of table
4292:10C4 LD   HL,(3F22:16162) *"?   42  34  63   move FPA1 to workspace
4295:10C7 LD   (3F39:16185),HL "9    34  57  63
4298:10CA LD   HL,(3F24:16164) *$?   42  36  63
4301:10CD LD   (3F3B:16187),HL ";?   34  59  63
4304:10D0 LD   A,(3F26:16166) :&?    58  38  63
4307:10D3 LD   (3F3D:16189),A 2=?    50  61  63
4310:10D6 JP   10EC:4332       ...  195 236  16
4313:10D9 PUSH BC              .    197
4314:10DA LD   HL,(3F39:16185) *9?   42  57  63   move workspace to FPA2
4317:10DD LD   (3F2B:16171),HL "+?   34  43  63
4320:10E0 LD   HL,(3F3B:16187) *;?   42  59  63
4323:10E3 LD   (3F2D:16173),HL "-?   34  45  63
4326:10E6 LD   A,(3F3D:16189) :=?    58  61  63
4329:10E9 LD   (3F2F:16175),A 2/?    50  47  63
4332:10EC CALL 0760:1888       .'.  205  96   7   FPA1*FPA2
4335:10EF LD   HL,(3F37:16183) *7?   42  55  63   get table address back
4338:10F2 LD   BC,0005:5       ...    1   5   0
4341:10F5 LD   DE,3F2B:16171   .+?   17  43  63
4344:10F8 LDIR                 ..   237 176       move next number in
4346:10FA LD   (3F37:16183),HL "7?   34  55  63   save table
```

```
4349:10FD CALL 062F:1583       ./.   205  47   6    FPA1+FPA2
4352:1100 POP  BC              .     193              count byte
4353:1101 DJNZ 10D9:4313       ..     16 214
4355:1103 RET                  .     201

Move FPA1 to FPA2
4356:1104 LD   HL,(3F22:16162) *"?    42  34  63
4359:1107 LD   (3F2B:16171),HL "+?    34  43  63
4362:110A LD   HL,(3F24:16164) *$?    42  36  63
4365:110D LD   (3F2D:16173),HL "-?    34  45  63
4368:1110 LD   HL,(3F25:16165) *%?    42  37  63
4371:1113 LD   (3F2E:16174),HL ".?    34  46  63
4374:1116 RET                  .     201

Move (HL) to FPA2
4375:1117 LD   DE,3F2B:16171   .+?    17  43  63
4378:111A LD   BC,0005:5       ...     1   5   0
4381:111D LDIR                 ..    237 176          (HL)->(DE) BC times
4383:111F RET                  .     201

load FPA1 with 1
4384:1120 LD   HL,0BF9:3065    !..    33 249  11    10^0
4387:1123 LD   DE,3F22:16162   ."?    17  34  63
4390:1126 LD   BC,0005:5       ...     1   5   0
4393:1129 LDIR                 ..    237 176          (HL)->(DE) BC times
4395:112B RET                  .     201

Push FPA1 to stack
4396:112C POP  DE              .     209              return address
4397:112D LD   HL,(3F25:16165) *%?    42  37  63
4400:1130 PUSH HL              .     229
4401:1131 LD   HL,(3F23:16163) *#?    42  35  63
4404:1134 PUSH HL              .     229
4405:1135 LD   HL,(3F21:16161) *!?    42  33  63
4408:1138 PUSH HL              .     229
4409:1139 EX   DE,HL           .     235
4410:113A JP   (HL)            .     233

Pop FPA1 from stack
4411:113B POP  DE              .     209
4412:113C POP  HL              .     225
4413:113D LD   (3F21:16161),HL "!?    34  33  63
4416:1140 POP  HL              .     225
4417:1141 LD   (3F23:16163),HL "#?    34  35  63
4420:1144 POP  HL              .     225
4421:1145 LD   (3F25:16165),HL "%?    34  37  63
4424:1148 EX   DE,HL           .     235
4425:1149 JP   (HL)            .     233

Push FPA2 to stack
4426:114A POP  DE              .     209
4427:114B LD   HL,(3F2E:16174) *.?    42  46  63
4430:114E PUSH HL              .     229
4431:114F LD   HL,(3F2C:16172) *,?    42  44  63
4434:1152 PUSH HL              .     229
4435:1153 LD   HL,(3F2A:16170) **?    42  42  63
4438:1156 PUSH HL              .     229
4439:1157 EX   DE,HL           .     235
4440:1158 JP   (HL)            .     233

Pop FPA2 from Stack
4441:1159 POP  DE              .     209
4442:115A POP  HL              .     225
4443:115B LD   (3F2A:16170),HL "*?    34  42  63
4446:115E POP  HL              .     225
4447:115F LD   (3F2C:16172),HL ",?    34  44  63
4450:1162 POP  HL              .     225
4451:1163 LD   (3F2E:16174),HL ".?    34  46  63
4454:1166 EX   DE,HL           .     235
4455:1167 JP   (HL)            .     233
```

```
Check range of FPA1
4456:1168 CALL 0932:2354      .2.  205  50   9    get integer
4459:116B RET  C              .    216
4460:116C LD   A,H            |    124
4461:116D OR   A              .    183
4462:116E JP   P,1178:4472      .x.  242 120  17
4465:1171 CP   00FF:255       ..   254 255
4467:1173 RET  C              .    216
4468:1174 LD   A,L            }    125
4469:1175 CP   0081:129       ..   254 129
4471:1177 RET                 .    201
4472:1178 JR   NZ,117D:4477        .    32        3
4474:117A LD   A,L            }    125
4475:117B CP   0080:128       ..   254 128
4477:117D CPL   CF            ?     63
4478:117E RET                 .    201

Make FPA1 positive
4479:117F LD   DE,090E:2318   ...   17  14   9    return address if negativ
4482:1182 LD   HL,3F25:16165  !%?   33  37  63    MSB
4485:1185 BIT  7,(HL)               203 126       ok if zero
4487:1187 RET  Z              .    200
4488:1188 LD   A,(3F26:16166) :&?   58  38  63    exponent
4491:118B OR   A              .    183
4492:118C RET  Z              .    200            ok if zero
4493:118D RES  7,(HL)         ..   203 190
4495:118F EX   DE,HL          .    235
4496:1190 EX   (SP),HL        .    227
4497:1191 JP   (HL)           .    233            set flag

Floating point constants for math routines
                      EXP   mantissa top bit assumed on-decimal shift on EXP
4497  248 23 114 49 128    00110001011100100001011111111000
        LOG(2)              .69314718
4502  41 59 170 56 129     00111000101010100011101100101001
        1/LOG(2)           1.44269504
4507  162 218 15 73 129     01001001000011111101101010100010
        pi/2               1.57079632
4512  110 131 249 34 128   00100010111110011000001101101110
        2/pi                .636619772
4517  162 218 15 73 128     01001001000011111101101010100010
        pi/4                .785398164
4522  208 204 19 84 127     01010100000100111100110011010000
                            .414213562
4527  154 121 130 26 130   00011010100000100111100110011010
                           2.41421356
4532  0 0 0 128 128        10000000000000000000000000000000
                           -.5
4537  52 243 4 181 129     10110101000001001111001100110100
                          -1.41421356
4542  52 243 4 53 128      00110101000001001111001100110100
                            .707106781

INTEGER CONSTANT USED BY POWER SERIES CALCULATIONS
4548  26  180   46106
4550  236  24   6380
4552  45  230   58925
4554  64  187   47936

4556     5                      6 constants used by ATN
4557  14 75 46 247 124     11110111001011100100101100001110
        not 1/11           -.060346883
4562  79 71 139 88 125     01011000100010110100011101001111
        not 1/9             .105734403
4567  102 130 209 145 126 10010001110100011000001001100110
        almost 1/7         -.142400777
4572  6 32 200 76 126      01001100110010000010000000000110
        1/5                 .199982167
4577  10 162 170 170 127   10101010101010101010001000001010
        1/3                -.333333076
4582  253 255 255 127 128 01111111111111111111111111111101
        1/1                 .999999999
```

```
4587     7                      8 constants for exponent
4588  13 108 93 45 113    00101101010111010110110000001101
                          2.06667101E-05
4593  124 111 101 25 116   00011001011001010110111101111100
                          1.46290047E-04
4598  246 248 118 47 119   00101111011101101111100011110110
                          1.3386897E-03
4603  118 143 141 29 122   00011101100011011000111101110110
                          9.61627016E-03
4608  178 152 88 99 124    01100011010110001001100010110010
                          .0555044141
4613  188 238 253 117 126 01110101111111011110111010111100
                          .240226488
4618  249 23 114 49 128    00110001011100100001011111111001
                          .693147181
4623  0 0 0 0 129          00000000000000000000000000000000
                          1.0

4628     3                      4 constants used by SIN
4629  132 136 209 151 121 10010111110100011000100010000100
                         -4.63313261E-03
4634  74 118 51 35 125     00100011001100110111011001001010
                          .0796879998
4639  210 228 93 165 128   10100101010111011111001001101010
                         -.645963956
4644  161 218 15 73 129    01001001000011111101101010100001
                          1.57079632

4649     4                      5 constants used by SIN
4650  255 251 185 106 118 01101010101110011111101111111111
                          8.95410543E-04
4655  54 18 217 170 123    10101010110110010001001000110110
                         -.0208554607
4660  74 218 224 1 127     00000001111000001101101001001010
                          .253668615
4665  215 229 233 157 129 10011101111010011110010111010111
                         -1.23370049
4670  254 255 255 127 128 01111111111111111111111111111110
                          1.0

4675     3                      4 constants used by LOG
4676  121 203 86 94 127    01011110010101101100101101111001
                          .434255942
4681  100 11 155 19 128    00010011100110110000101101100100
                          .576584541
4686  21 147 56 118 128    01110110001110001001001100010101
                          .961800759
4691  29 59 170 56 130     00111000101010100011101100011101
                          2.88539007

4696:1258 JP   NZ,1F03:7939       ...  194          3  31
4699:125B LD   DE,(3F25:16165)     237  91  37  63 FPA1 MSB
4703:125F LD   BC,(3F3E:16190)     237  75  62  63 random seed
4707:1263 LD   HL,(3F40:16192) *@?   42  64  63    random seed
4710:1266 LD   A,D                z    122
4711:1267 OR   A                  .    183
4712:1268 JR   Z,12A2:4770         (8    40         56        is it RND(0)
4714:126A BIT  7,E                .{   203 123        is it RND(-) for seed
4716:126C JR   Z,1273:4723         (.    40          5
4718:126E EX   DE,HL              .    235
4719:126F LD   BC,(3F23:16163)     237  75  35  63 use seed supplied
4723:1273 LD   (3F2B:16171),BC     237  67  43  63 write to FPA2
4727:1277 LD   (3F2D:16173),HL "-?   34  45  63
4730:127A LD   BC,0001:1          ...    1   1   0
4733:127D LD   DE,0000:0          ...   17   0   0
4736:1280 LD   HL,11C8:4552       !..   33 200  17    4 constants for RND
4739:1283 CALL 12AC:4780          ...  205 172  18    do it
4742:1286 LD   (3F3E:16190),A 2>?   50  62  63    write back to seed
4745:1289 CALL 12AC:4780          ...  205 172  18
4748:128C LD   (3F3F:16191),A 2??   50  63  63
4751:128F CALL 12AC:4780          ...  205 172  18
4754:1292 LD   (3F40:16192),A 2@?   50  64  63
4757:1295 CALL 12AC:4780          ...  205 172  18
4760:1298 LD   (3F41:16193),A 2A?   50  65  63
4763:129B LD   BC,(3F3E:16190)     237  75  62  63 get number back
4767:129F LD   HL,(3F40:16192)      42  64  63
4770:12A2 XOR  A                  .    175
4771:12A3 LD   (3F17:16151),A 2.?   50  23  63    control byte
```

```
4774:12A6 LD   D,0080:128      ..    22 128
4776:12A8 LD   E,A             .     95
4777:12A9 JP   06F8:1784       ...   195 248   6    compute number and exit

randomize
4780:12AC LD   A,(HL)          .     126              get value
4781:12AD PUSH HL              .     229
4782:12AE CALL 07B0:1968       ...   205 176   7    A*FPA2
4785:12B1 POP  HL              .     225
4786:12B2 EX   AF,AF'          .       8
4787:12B3 INC  HL              #      35              advance to next byte
4788:12B4 RET                  .     201

PUSH FPA1 to STACK and ???
4789:12B5 POP  HL              .     225              save HL in A and A'
4790:12B6 LD   A,L             }     125
4791:12B7 EX   AF,AF'          .       8
4792:12B8 LD   A,H             |     124
4793:12B9 LD   HL,(3F25:16165) *%?    42  37  63
4796:12BC PUSH HL              .     229              push FPA1
4797:12BD LD   HL,(3F23:16163) *#?    42  35  63
4800:12C0 PUSH HL              .     229
4801:12C1 LD   HL,(3F21:16161) *!?    42  33  63   LSB and control byte
4804:12C4 PUSH HL              .     229
4805:12C5 INC  L               ,      44
4806:12C6 DEC  L               -      45
4807:12C7 JR   NZ,12CD:4813          .    32        4
4809:12C9 LD   H,A             g     103
4810:12CA EX   AF,AF'          .       8
4811:12CB LD   L,A             o     111
4812:12CC JP   (HL)            .     233              if control byte 0
4813:12CD PUSH AF              .     245
4814:12CE LD   HL,(3F22:16162) *"?    42  34  63    FPA1 LSB
4817:12D1 PUSH HL              .     229
4818:12D2 LD   HL,0004:4       !..    33   4   0
4821:12D5 ADD  HL,SP           9      57              raise stack by 4
4822:12D6 LD   A,L             }     125
4823:12D7 EX   (SP),HL         .     227
4824:12D8 LD   (HL),A          w     119              save current L
4825:12D9 INC  HL              #      35
4826:12DA EX   (SP),HL         .     227
4827:12DB LD   A,H             |     124
4828:12DC EX   (SP),HL         .     227              save current H
4829:12DD LD   (HL),A          w     119
4830:12DE POP  HL              .     225
4831:12DF POP  HL              .     225              restore stack
4832:12E0 EX   AF,AF'          .       8
4833:12E1 LD   L,A             o     111
4834:12E2 JP   (HL)            .     233

POP FPA1 from STACK and ???
4835:12E3 POP  HL              .     225              save HL in A and A'
4836:12E4 LD   A,L             }     125
4837:12E5 EX   AF,AF'          .       8
4838:12E6 LD   A,H             |     124
4839:12E7 POP  HL              .     225
4840:12E8 INC  L               ,      44
4841:12E9 DEC  L               -      45
4842:12EA LD   (3F21:16161),HL "!?    34  33  63    FPA1
4845:12ED POP  HL              .     225
4846:12EE LD   (3F23:16163),HL "#?    34  35  63
4849:12F1 POP  HL              .     225
4850:12F2 LD   (3F25:16165),HL "%?    34  37  63
4853:12F5 JR   NZ,12FB:4859          .    32        4
4855:12F7 LD   H,A             g     103
4856:12F8 EX   AF,AF'          .       8
4857:12F9 LD   L,A             o     111
4858:12FA JP   (HL)            .     233
4859:12FB LD   HL,(3F22:16162) *"?    42  34  63   address of pointer
4862:12FE LD   (HL),0022:34    6"     54  34        write 3F22 to pointer
4864:1300 INC  HL              #      35
4865:1301 LD   (HL),003F:63    6?     54  63
4867:1303 JR   12F7:4855       ..     24 242        same exit
```

```
MOVE FPA1 TO FPA2
4869:1305 LD   HL,(3F25:16165) *%?   42  37  63
4872:1308 LD   (3F2E:16174),HL ".?   34  46  63
4875:130B LD   HL,(3F23:16163) *#?   42  35  63
4878:130E LD   (3F2C:16172),HL ",?   34  44  63
4881:1311 LD   HL,(3F21:16161) *!?   42  33  63
4884:1314 LD   (3F2A:16170),HL "*?   34  42  63
4887:1317 INC  L               ,     44
4888:1318 DEC  L               -     45
4889:1319 RET  Z               .    200
4890:131A LD   HL,(3F2B:16171) *+?   42  43  63   if pointer, write address

4893:131D LD   (HL),002B:43    6+    54  43
4895:131F INC  HL              #     35
4896:1320 LD   (HL),003F:63    6?    54  63
4898:1322 RET                  .    201

MOVE FPA2 TO FPA1
4899:1323 LD   HL,(3F2E:16174) *.?   42  46  63
4902:1326 LD   (3F25:16165),HL "%?   34  37  63
4905:1329 LD   HL,(3F2C:16172) *,?   42  44  63
4908:132C LD   (3F23:16163),HL "#?   34  35  63
4911:132F LD   HL,(3F2A:16170) **?   42  42  63
4914:1332 LD   (3F21:16161),HL "!?   34  33  63
4917:1335 INC  L               ,     44
4918:1336 DEC  L               -     45
4919:1337 RET  Z               .    200
4920:1338 LD   HL,(3F22:16162) *"?   42  34  63   if pointer, write address

4923:133B LD   (HL),0022:34    6"    54  34
4925:133D INC  HL              #     35
4926:133E LD   (HL),003F:63    6?    54  63
4928:1340 RET                  .    201

Get number from a token line
4929:1341 CP   0093:147        ..   254 147
4931:1343 JP   NC,135E:4958       .^.  210      94  19
4934:1346 AND  001F:31         ..   230   31
4936:1348 LD   HL,1363:4963    !c.   33  99  19    offset table
4939:134B ADD  A,L             .    133
4940:134C LD   L,A             o    111
4941:134D JP   NC,1351:4945       .Q.  210      81  19
4944:1350 INC  H               $     36
4945:1351 LD   A,(HL)          .    126           get offset
4946:1352 POP  HL              .    225           base address from CALLing
4947:1353 ADD  A,L             .    133
4948:1354 LD   L,A             o    111           add offset
4949:1355 JP   NC,1359:4953       .Y.  210      89  19
4952:1358 INC  H               $     36
4953:1359 LD   A,(HL)          .    126           get address
4954:135A INC  HL              #     35
4955:135B LD   H,(HL)          f    102
4956:135C LD   L,A             o    111
4957:135D JP   (HL)            .    233           go to it
4958:135E LD   A,000E:14       >.    62  14       if no token use 14
4960:1360 JP   1352:4946       .R.  195  82  19

offset table
4963:1363           0  0  0  0  0
4968:1368  0  0  0  0  0  2  4  6
4976:1370  6  6  6  8 10 12

Get an Integer Digit from Token Line
4982:1376 AND  0037:55         .7   230   55       SET CF if 1377
4984:1378 LD   H,0000:0        &.    38    0
4986:137A LD   A,(DE)          .     26
4987:137B RL   A               .     23
4988:137C AND  001F:31         ..   230   31
4990:137E RR   A               .     31
4991:137F LD   L,A             o    111
4992:1380 JP   0967:2407       .g.  195 103   9    put HL to FPA1 or 2
```

```
get 2nd half of integer digit ie <255
4995:1383 AND  0037:55        .7   230   55        SET CF if 1384
4997:1385 EXX                 .    217
4998:1386 DEC  C              .     13             decrement counter
4999:1387 EXX                 .    217
5000:1388 INC  DE             .     19
5001:1389 LD   A,(DE)         .     26
5002:138A LD   H,0000:0       &.    38    0
5004:138C LD   L,A            o    111
5005:138D JP   0967:2407      .g.  195 103   9    put HL to FPA1 or 2

Get integer >255 from Token line
5008:1390 AND  0037:55        .7   230   55        SET CF if 1391
5010:1392 EXX                 .    217
5011:1393 DEC  C              .     13             decrement counter
5012:1394 DEC  C              .     13
5013:1395 EXX                 .    217
5014:1396 INC  DE             .     19
5015:1397 LD   A,(DE)         .     26
5016:1398 LD   L,A            o    111
5017:1399 INC  DE             .     19
5018:139A LD   A,(DE)         .     26
5019:139B LD   H,A            g    103
5020:139C JP   0967:2407      .g.  195 103   9    put HL in FPA1 or 2

Load variable to FPA1
5023:139F PUSH BC             .    197
5024:13A0 LD   HL,3F22:16162  !"?   33  34  63    FPA1 LSB
5027:13A3 PUSH HL             .    229
5028:13A4 CALL 199F:6559      ...  205 159  25    check variable type

PC points here and is popped to calculate offset
5031:13A7   195  19           13C3:5059
5033:13A9   214  19           13D6:5078
5035:13AB   231  19           13E7:5095
5037:13AD    15  20           140F:5135
5039:13AF    15  20           140F:5135

LOAD variable to FPA2
5041:13B1 PUSH BC             .    197
5042:13B2 LD   HL,3F2B:16171  !+?   33  43  63    FPA2 LSB
5045:13B5 PUSH HL             .    229
5046:13B6 CALL 19A0:6560      ...  205 160  25    check variable type

PC points here and is popped to calculate offset
5049:13B9   195  19           13C3:5059
5051:13BB   214  19           13D6:5078
5053:13BD   231  19           13E7:5095
5055:13BF    21  20           1415:5141
5057:13C1    21  20           1415:5141

Move FP number from (BC) to (HL)
5059:13C3 EX   DE,HL          .    235
5060:13C4 EX   (SP),HL        .    227
5061:13C5 LD   D,B            P     80
5062:13C6 LD   E,C            Y     89
5063:13C7 EX   DE,HL          .    235
5064:13C8 LDI                 ..   237 160        move 5 bytes
5066:13CA LDI                 ..   237 160
5068:13CC LDI                 ..   237 160
5070:13CE LDI                 ..   237 160
5072:13D0 LDI                 ..   237 160
5074:13D2 POP  DE             .    209
5075:13D3 POP  BC             .    193
5076:13D4 XOR  A              .    175
5077:13D5 RET                 .    201
```

```
Move integer from (BC) to (HL)
5078:13D6 LD   A,(BC)           .     10              save it in A & A'
5079:13D7 EX   AF,AF'           .      8
5080:13D8 INC  BC               .      3
5081:13D9 LD   A,(BC)           .     10
5082:13DA POP  HL               .    225
5083:13DB LD   BC,0004:4        ...    1   4   0
5086:13DE ADD  HL,BC            .      9
5087:13DF POP  BC               .    193
5088:13E0 PUSH DE               .    213
5089:13E1 LD   D,A              W     87              put it in DE
5090:13E2 EX   AF,AF'           .      8
5091:13E3 LD   E,A              _     95
5092:13E4 JP   0972:2418        _r.   195 114   9     move it?

string number from (BC) to (HL)
5095:13E7 LD   A,(BC)           .     10
5096:13E8 LD   L,A              o     111             get pointer in HL
5097:13E9 INC  BC               .      3
5098:13EA LD   A,(BC)           .     10
5099:13EB LD   H,A              g     103
5100:13EC INC  HL               #     35              skip pointer to pointer
5101:13ED INC  HL               #     35
5102:13EE LD   A,(HL)           .    126
5103:13EF EX   DE,HL            .    235
5104:13F0 EX   (SP),HL          .    227
5105:13F1 PUSH HL               .    229
5106:13F2 CALL 1C60:7264        .'.   205  96  28     define string
5109:13F5 POP  BC               .    193
5110:13F6 LD   A,L              }    125
5111:13F7 LD   (BC),A           .      2
5112:13F8 INC  BC               .      3
5113:13F9 LD   A,H              |    124
5114:13FA LD   (BC),A           .      2
5115:13FB INC  HL               #     35
5116:13FC INC  HL               #     35
5117:13FD LD   A,(HL)           .    126              length of string
5118:13FE OR   A                .    183
5119:13FF LD   C,A              O     79
5120:1400 LD   B,0000:0         ..     6   0
5122:1402 INC  DE               .     19
5123:1403 INC  HL               #     35
5124:1404 EX   DE,HL            .    235
5125:1405 JR   Z,1409:5129        (.    40          2
5127:1407 LDIR                  ..   237 176         move string
5129:1409 POP  DE               .    209
5130:140A POP  BC               .    193
5131:140B LD   A,00FF:255       >.    62 255
5133:140D SET  CF               7     55
5134:140E RET                   .    201

Evaluate variable in token line to FPA1
5135:140F POP  HL               .    225
5136:1410 CALL 142F:5167        ./.  205  47  20     do it
5139:1413 POP  BC               .    193
5140:1414 RET                   .    201

Evaluate variable in token line to FPA2
5141:1415 POP  HL               .    225
5142:1416 LD   (3F0D:16141),A 2.?     50  13  63     temporary storage
5145:1419 CALL 12B5:4789        ...  205 181  18     push FPA1
5148:141C LD   A,(3F0D:16141) :.?     58  13  63
5151:141F CALL 142F:5167        ./.  205  47  20     do it
5154:1422 LD   (3F21:16161),A 2!?     50  33  63     control byte
5157:1425 LD   B,A              G     71
5158:1426 CALL 1305:4869        ...  205   5  19     FPA1 to FPA2
5161:1429 CALL 12E3:4835        ...  205 227  18     pop FPA1
5164:142C LD   A,B              x    120
5165:142D POP  BC               .    193
5166:142E RET                   .    201
```

```
Send variable to FPA1
5167:142F INC  B                .      4
5168:1430 DEC  B                .      5
5169:1431 JP   Z,1EFD:7933        ...  202 253  30          formula too complex
5172:1434 BIT  6,A              .w    203 119
5174:1436 JR   NZ,1462:5218       *     32       42
5176:1438 PUSH BC               .     197
5177:1439 EXX                   .     217
5178:143A DEC  C                .      13            line counter
5179:143B EXX                   .     217
5180:143C INC  DE               .      19
5181:143D AND  0020:32          .     230   32
5183:143F JR   NZ,1457:5207       .    32       22
5185:1441 CALL 1733:5939        .3.   205  51  23    get equation
5188:1444 OR   A                .     183
5189:1445 POP  HL               .     225
5190:1446 PUSH DE               .     213
5191:1447 CALL 1732:5938        .2.   205  50  23    jump(HL)
5194:144A POP  DE               .     209
5195:144B EXX                   .     217
5196:144C DEC  C                .      13            line counter
5197:144D EXX                   .     217
5198:144E INC  DE               .      19
5199:144F LD   A,(DE)           .      26
5200:1450 CP   00B8:184         ..    254 184
5202:1452 JP   NZ,1F0C:7948       ...  194       12  31    syntax error
5205:1455 XOR  A                .     175
5206:1456 RET                   .     201
5207:1457 CALL 1733:5939        .3.   205  51  23    get equation
5210:145A OR   A                .     183
5211:145B POP  HL               .     225
5212:145C CALL 1732:5938        .2.   205  50  23    JP (HL)
5215:145F LD   A,00FF:255       >.     62 255
5217:1461 RET                   .     201

?????
5218:1462 CP   00C8:200         ..    254 200
5220:1464 JP   NZ,1F0C:7948       ...  194       12  31
5223:1467 EXX                   .     217
5224:1468 DEC  C                .      13            line counter
5225:1469 EXX                   .     217
5226:146A INC  DE               .      19
5227:146B PUSH BC               .     197
5228:146C CALL 1733:5939        .3.   205  51  23    get equation
5231:146F POP  HL               .     225
5232:1470 OR   A                .     183
5233:1471 JP   NZ,1F03:7939       ...  194        3  31    type mismatch
5236:1474 EXX                   .     217
5237:1475 DEC  C                .      13            line counter
5238:1476 EXX                   .     217
5239:1477 INC  DE               .      19
5240:1478 LD   A,(DE)           .      26
5241:1479 CP   00B8:184         ..    254 184
5243:147B JP   NZ,1F0C:7948       ...  194       12  31    syntax error
5246:147E PUSH DE               .     213
5247:147F LD   A,(HL)           .     126
5248:1480 LD   BC,0004:4        ...     1   4   0
5251:1483 ADD  HL,BC            .       9
5252:1484 LD   C,(HL)           N      78
5253:1485 INC  C                .      12
5254:1486 SUB   C               .     145
5255:1487 SUB   0006:6          ..    214   6
5257:1489 EXX                   .     217
5258:148A PUSH BC               .     197
5259:148B LD   C,A              O      79
5260:148C EXX                   .     217
5261:148D INC  HL               #      35
5262:148E ADD  HL,BC            .       9
5263:148F EX   DE,HL            .     235
5264:1490 CALL 199F:6559        ...   205 159  25    check variable type
5267:1493 SUB   L               .     149
5268:1494 INC  D                .      20
5269:1495 INC  DE               .      19
5270:1496 INC  DE               .      19
5271:1497 LD   A,E              {     123
5272:1498 EX   AF,AF'           .       8
```

```
5273:1499 LD   A,D              z     122
5274:149A LD   E,C              Y      89
5275:149B LD   D,B              P      80
5276:149C LD   HL,FFFB:65531    !..    33 251 255     DCB
5279:149F ADD  HL,SP            9      57             lower stack by 4
5280:14A0 LD   SP,HL            .     249
5281:14A1 LD   BC,0005:5        ...     1   5   0
5284:14A4 EX   DE,HL            .     235
5285:14A5 LDIR                  ..    237 176        (HL)->(DE) BC times
5287:14A7 DEC  HL               +      43
5288:14A8 EX   DE,HL            .     235
5289:14A9 LD   HL,3F26:16166    !&?    33  38  63     FPA1
5292:14AC LD   BC,0005:5        ...     1   5   0
5295:14AF LDDR                  ..    237 184
5297:14B1 INC  DE               .      19
5298:14B2 PUSH DE               .     213
5299:14B3 LD   D,A              W      87
5300:14B4 EX   AF,AF'           .       8
5301:14B5 LD   E,A              _      95
5302:14B6 CALL 1733:5939        .3.   205  51  23     get equation
5305:14B9 POP  DE               .     209
5306:14BA LD   BC,0005:5        ...     1   5   0
5309:14BD LD   HL,0000:0        !..    33   0   0
5312:14C0 ADD  HL,SP            9      57
5313:14C1 LDIR                  ..    237 176
5315:14C3 LD   SP,HL            .     249
5316:14C4 EXX                   .     217
5317:14C5 POP  BC               .     193
5318:14C6 EXX                   .     217
5319:14C7 POP  DE               .     209
5320:14C8 XOR  A                .     175
5321:14C9 RET                   .     201

?????
5322:14CA LD   HL,3F22:16162    !"?    33  34  63     FPA1
5325:14CD JP   14D3:5331        ...   195 211  20

?????
5328:14D0 LD   HL,3F2B:16171    !+?    33  43  63     FPA2
5331:14D3 EXX                   .     217
5332:14D4 DEC  C                .      13             line counter
5333:14D5 LD   A,C              y     121
5334:14D6 EXX                   .     217
5335:14D7 EX   DE,HL            .     235
5336:14D8 INC  HL               #      35
5337:14D9 PUSH BC               .     197
5338:14DA LD   B,0000:0         ..      6   0
5340:14DC LD   C,(HL)           N      78
5341:14DD SUB  C                .     145
5342:14DE EXX                   .     217
5343:14DF LD   C,A              O      79
5344:14E0 EXX                   .     217
5345:14E1 LD   A,C              y     121
5346:14E2 PUSH HL               .     229
5347:14E3 LD   H,D              b      98
5348:14E4 LD   L,E              k     107
5349:14E5 CALL 1C60:7264        .'.   205  96  28     define string
5352:14E8 EX   DE,HL            .     235
5353:14E9 LD   (HL),E           s     115
5354:14EA INC  HL               #      35
5355:14EB LD   (HL),D           r     114
5356:14EC POP  HL               .     225
5357:14ED LD   A,B              x     120
5358:14EE OR   C                .     177
5359:14EF JP   Z,14F9:5369      ...   202 249  20
5362:14F2 INC  HL               #      35
5363:14F3 INC  DE               .      19
5364:14F4 INC  DE               .      19
5365:14F5 INC  DE               .      19
5366:14F6 LDIR                  ..    237 176
5368:14F8 DEC  HL               +      43
5369:14F9 EX   DE,HL            .     235
5370:14FA POP  BC               .     193
5371:14FB LD   A,00FF:255       >.     62 255
5373:14FD RET                   .     201
```

```
5374:14FE LD   HL,3F22:16162  !"?   33  34  63    FPA1
5377:1501 JP   1507:5383      ...  195   7  21
5380:1504 LD   HL,3F2B:16171  !+?   33  43  63    FPA2
5383:1507 EXX                 .    217
5384:1508 LD   A,C            y    121
5385:1509 SUB   0005:5        ..   214   5        decrement line counter
5387:150B LD   C,A            O     79
5388:150C EXX                 .    217
5389:150D INC  DE             .     19
5390:150E PUSH BC             .    197
5391:150F EX   DE,HL          .    235
5392:1510 LDI                 ..   237 160       move it
5394:1512 LDI                 ..   237 160
5396:1514 LDI                 ..   237 160
5398:1516 LDI                 ..   237 160
5400:1518 LDI                 ..   237 160
5402:151A EX   DE,HL          .    235
5403:151B DEC  DE             .     27
5404:151C POP  BC             .    193
5405:151D XOR  A              .    175
5406:151E RET                 .    201

Move string at (HL) to (DE)
5407:151F LD   B,0000:0       ..     6   0
5409:1521 LD   C,(HL)         N     78         first byte is length
5410:1522 INC  HL             #     35
5411:1523 LD   A,B            x    120
5412:1524 OR   C              .    177
5413:1525 RET  Z              .    200         if null string
5414:1526 LDIR                ..   237 176     move it
5416:1528 RET                 .    201

ADD
5417:1529 JR   NZ,1532:5426         .    32        7       if pointer to numbers
5419:152B CALL 062F:1583      ./.  205  47   6    FPA1 + FPA2
5422:152E POP  DE             .    209
5423:152F POP  BC             .    193
5424:1530 XOR  A              .    175
5425:1531 RET                 .    201
5426:1532 LD   HL,(3F2B:16171)      42  43  63     FPA2
5429:1535 LD   DE,(3F22:16162)      237  91  34  63 FPA1
5433:1539 INC  HL             #     35              skip pointer bytes
5434:153A INC  HL             #     35
5435:153B INC  DE             .     19
5436:153C INC  DE             .     19
5437:153D LD   A,(DE)         .     26
5438:153E ADD  A,(HL)         .    134
5439:153F JP   C,1F09:7945       ...  218         9  31    string too long?
5442:1542 PUSH HL             .    229
5443:1543 LD   HL,3F22:16162  !"?   33  34  63    FPA1
5446:1546 CALL 1C60:7264      .'.  205  96  28    define string
5449:1549 LD   (3F22:16162),HL      34  34  63
5452:154C INC  HL             #     35
5453:154D INC  HL             #     35
5454:154E INC  HL             #     35
5455:154F EX   DE,HL          .    235
5456:1550 CALL 151F:5407      ...  205  31  21    move (HL) to (DE)
5459:1553 POP  HL             .    225
5460:1554 CALL 151F:5407      ...  205  31  21
5463:1557 POP  DE             .    209
5464:1558 POP  BC             .    193
5465:1559 LD   A,00FF:255     >.    62 255
5467:155B RET                 .    201

SUBTRACT
5468:155C CALL 0627:1575      .'.  205  39   6    FPA1-FPA2
5471:155F POP  DE             .    209
5472:1560 POP  BC             .    193
5473:1561 XOR  A              .    175
5474:1562 RET                 .    201

MULTIPLY
5475:1563 CALL 0760:1888      .'.  205  96   7    FPA1*FPA2
5478:1566 POP  DE             .    209
5479:1567 POP  BC             .    193
5480:1568 XOR  A              .    175
5481:1569 RET                 .    201
```

```
DIVIDE
5482:156A CALL 07E2:2018      ...  205 226   7    FPA1/FPA2
5485:156D POP  DE             .    209
5486:156E POP  BC             .    193
5487:156F XOR  A              .    175
5488:1570 RET                 .    201

EXPONENT
5489:1571 CALL 0E70:3696      .p.  205 112  14    FPA1^FPA2
5492:1574 POP  DE             .    209
5493:1575 POP  BC             .    193
5494:1576 XOR  A              .    175
5495:1577 RET                 .    201

Evaluate logical operators    entry point depends on operator
do AND
5496:1578 LD   A,(3F26:16166) :&?   58  38  63    FPA1 exponent
5499:157B OR   A              .    183
5500:157C JP   Z,15F8:5624     ...  202 248  21
5503:157F LD   A,(3F2F:16175) :/?   58  47  63    FPA2 exponent
5506:1582 OR   A              .    183
5507:1583 JP   Z,15F8:5624      ...  202 248  21
5510:1586 JP   15DA:5594      ...  195 218  21

do OR
5513:1589 LD   A,(3F26:16166) :&?   58  38  63    FPA1 exponent
5516:158C LD   B,A            G    71
5517:158D LD   A,(3F2F:16175) :/?   58  47  63    FPA2 exponent
5520:1590 OR   B              .    176
5521:1591 JP   Z,15F8:5624      ...  202 248  21          if both zero
5524:1594 JP   15DA:5594      ...  195 218  21

do <
5527:1597 JR   NZ,15A0:5536       .    32        7
5529:1599 CALL 09B8:2488      ...  205 184   9    compare FPA1 and FPA2
5532:159C JR   C,15DA:5594     8<    56        60
5534:159E JR   15F8:5624      .X    24  88
5536:15A0 CALL 1604:5636      ...  205   4  22    compare FPA1 and FPA2
5539:15A3 JR   159C:5532      ..    24 247

do >
5541:15A5 JR   NZ,15B0:5552       .    32        9
5543:15A7 CALL 09B8:2488      ...  205 184   9    comapre FPA1 and FPA2
5546:15AA JR   C,15F8:5624     8L    56        76
5548:15AC JR   Z,15F8:5624     (J    40        74
5550:15AE JR   15DA:5594      .*    24  42
5552:15B0 CALL 1604:5636      ...  205   4  22    compare fPA1 and FPA2
5555:15B3 JR   15AA:5546      ..    24 245
do <=
5557:15B5 JR   NZ,15C0:5568       .    32        9
5559:15B7 CALL 09B8:2488      ...  205 184   9    compare FPA1 and FPA2
5562:15BA JR   C,15DA:5594    8.    56  30
5564:15BC JR   Z,15DA:5594    (.    40  28
5566:15BE JR   15F8:5624      .8    24  56
5568:15C0 CALL 1604:5636      ...  205   4  22    compare FPA1 and FPA2
5571:15C3 JR   15BA:5562      ..    24 245

do >=
5573:15C5 JR   NZ,15CE:5582    .    32   7
5575:15C7 CALL 09B8:2488      ...  205 184   9    compare FPA1 and FPA2
5578:15CA JR   NC,15DA:5594   0.    48  14
5580:15CC JR   15F8:5624      .*    24  42
5582:15CE CALL 1604:5636      ...  205   4  22    comapre FPA1 and FPA2
5585:15D1 JR   15CA:5578      ..    24 247
```

```
do <>
5587:15D3 JR   NZ,15EC:5612     .    32  23
5589:15D5 CALL 09B8:2488        ...  205 184   9    compare FPA1 and FPA2
5592:15D8 JR   Z,15F8:5624      (.    40  30
5594:15DA LD   A,0081:129       >.    62 129        set FPA1 to 1
5596:15DC LD   (3F26:16166),A 2&?   50  38  63
5599:15DF LD   HL,0000:0        !..   33   0   0
5602:15E2 LD   (3F22:16162),HL ""?   34  34  63
5605:15E5 LD   (3F24:16164),HL "$?   34  36  63
5608:15E8 POP  DE               .    209
5609:15E9 POP  BC               .    193
5610:15EA XOR  A                .    175
5611:15EB RET                   .    201
5612:15EC CALL 1604:5636        ...  205   4  22    compare FPA1 to FPA2
5615:15EF JR   15D8:5592        ..    24 231

do =
5617:15F1 JR   NZ,15FF:5631       .     32       12
5619:15F3 CALL 09B8:2488        ...  205 184   9    compare FPA1 and FPA2
5622:15F6 JR   Z,15DA:5594        (.    40 226
5624:15F8 XOR  A                .    175
5625:15F9 LD   (3F26:16166),A 2&?   50  38  63    set FPA1 to 0
5628:15FC POP  DE               .    209
5629:15FD POP  BC               .    193
5630:15FE RET                   .    201
5631:15FF CALL 1604:5636        ...  205   4  22
5634:1602 JR   15F6:5622        ..    24 242

Compare strings pointed to by FPA1 and FPA2
5636:1604 LD   HL,(3F22:16162) *"?   42  34  63    FPA1
5639:1607 EX   DE,HL            .    235
5640:1608 LD   HL,(3F2B:16171) *+?   42  43  63    FPA2
5643:160B INC  HL               #     35           skip pointer bytes
5644:160C INC  HL               #     35
5645:160D INC  DE               .     19
5646:160E INC  DE               .     19
5647:160F CALL 2E3F:11839       .?.  205  63  46    string compare
5650:1612 EX   AF,AF'           .      8
5651:1613 XOR  A                .    175
5652:1614 EX   AF,AF'           .      8
5653:1615 RET                   .    201

do NOT?
5654:1616 LD   A,(DE)           .     26
5655:1617 CP   00B7:183         ..   254 183
5657:1619 JP   NZ,1625:5669       .%.  194       37  22
5660:161C CALL 1733:5939        .3.  205  51  23    get equation from line
5663:161F EXX                   .    217
5664:1620 DEC  C                .     13           decrement line counter
5665:1621 EXX                   .    217
5666:1622 INC  DE               .     19
5667:1623 RRC  A                .     15
5668:1624 RET                   .    201
5669:1625 CP   00A1:161         ..   254 161
5671:1627 JP   NZ,1637:5687       .7.  194       55  22
5674:162A CALL 1C83:7299        ...  205 131  28
5677:162D CALL 16CD:5837        ...  205 205  22    get number from line
5680:1630 AND  0037:55        .7    230  55         option to set CF
5682:1632 CALL 0908:2312        ...  205   8   9    toggle sign
5685:1635 XOR  A                .    175
5686:1636 RET                   .    201
5687:1637 CP   00AD:173         ..   254 173
5689:1639 JP   NZ,16CD:5837       ...  194 205  22          get number from line
5692:163C CALL 1C83:7299        ...  205 131  28    check stack
5695:163F CALL 16CD:5837        ...  205 205  22    get number from line
5698:1642 AND  0037:55        .7    230  55         option to set CF
5700:1644 CALL 0916:2326        ...  205  22   9    toggle sign
5703:1647 XOR  A                .    175
5704:1648 RET                   .    201
```

```
do NOT?
5705:1649 LD   A,(DE)          .     26
5706:164A CP   00B7:183        ..   254 183
5708:164C JP   NZ,1664:5732      .d.  194 100  22
5711:164F PUSH BC              .    197
5712:1650 CALL 12B5:4789       ...  205 181  18    push FPA1
5715:1653 CALL 1733:5939       .3.  205  51  23    get equation from line
5718:1656 LD   B,A             G     71
5719:1657 CALL 1305:4869       ...  205   5  19    move FPA1 to FPA2
5722:165A CALL 12E3:4835       ...  205 227  18    pop FPA1
5725:165D LD   A,B             x    120
5726:165E POP  BC              .    193
5727:165F EXX                  .    217
5728:1660 DEC  C               .     13           line counter
5729:1661 EXX                  .    217
5730:1662 INC  DE              .     19
5731:1663 RET                  .    201
5732:1664 CP   00A1:161        ..   254 161
5734:1666 JP   NZ,1672:5746      .r.  194 114  22
5737:1669 CALL 1C83:7299       ...  205 131  28    check stack
5740:166C CALL 16E5:5861       ...  205 229  22    get number from line
5743:166F JP   1631:5681       .1.  195  49  22    set CF for FPA2
5746:1672 CP   00AD:173        ..   254 173
5748:1674 JP   NZ,16E5:5861      ...  194 229  22          get number from line
5751:1677 CALL 1C83:7299       ...  205 131  28    check stack
5754:167A CALL 16E5:5861       ...  205 229  22    get number from line
5757:167D JP   1643:5699       .C.  195  67  22    set CF for FPA2

Find operator from token line
5760:1680 EXX                  .    217
5761:1681 LD   A,C             y    121
5762:1682 EXX                  .    217
5763:1683 OR   A               .    183           end of line?
5764:1684 JR   Z,16A0:5792       (.    40        26          no token
5766:1686 EXX                  .    217
5767:1687 DEC  C               .     13           line counter
5768:1688 EXX                  .    217
5769:1689 INC  DE              .     19
5770:168A LD   A,(DE)          .     26
5771:168B SUB   00A0:160       ..   214 160       lowest operator token
5773:168D CP   000D:13         ..   254  13       number of operators
5775:168F JR   NC,169C:5788      0.    48        11          not an operator
5777:1691 LD   C,A             O     79
5778:1692 ADD  A,C             .    129
5779:1693 ADD  A,C             .    129           3 bytes per entry
5780:1694 LD   BC,16A5:5797    ...    1 165  22    start of table
5783:1697 ADD  A,C             .    129
5784:1698 LD   C,A             O     79           get offset
5785:1699 RET  NC              .    208
5786:169A INC  B               .      4
5787:169B RET                  .    201
5788:169C EXX                  .    217
5789:169D INC  C               .     12           put token back
5790:169E EXX                  .    217
5791:169F DEC  DE              .     27           restore counter
5792:16A0 LD   BC,16CC:5836    ...    1 204  22    end of table (no token)
5795:16A3 XOR  A               .    175
5796:16A4 RET                  .    201

Jump table and order of precedence for tokens
5797:16A5   5   41   21       5417:1529           +
5800:16A8   5   92   21       5468:155C           -
5803:16AB   4   99   21       5475:1563           *
5806:16AE   4  106   21       156A:5482           /
5809:16B1   0  113   21       1571:5489           ^
5812:16B4   6  151   21       1597:5527           >
5815:16B7   6  165   21       15A5:5541           <
5818:16BA   6  181   21       15B5:5557           <=
5821:16BD   6  197   21       15C5:5573           >=
5824:16C0   6  211   21       15D3:5587           <>
5827:16C3   6  241   21       15F1:5617           =
5830:16C6   7  120   21       1578:5496           AND
5833:16C9   8  137   21       1589:5513           OR
5836:16CC 255                 end of table--no token flag
```

```
Get number from Token Line     to FPA1
5837:16CD EXX                  .    217
5838:16CE DEC  C               .     13
5839:16CF EXX                  .    217
5840:16D0 INC  DE              .     19
5841:16D1 LD   A,(DE)          .     26
5842:16D2 CALL 1341:4929       .A.  205  65  19    get number

PC points here for return address table--popped to HL
5845:16D5  118  19             1376:4982           get integer digit
5847:16D7  131  19             1383:4995           get second digit
5849:16D9  144  19             1390:5008           get number >255
5851:16DB  159  19             139F:5023           load number to FPA1
5853:16DD  201  24             18C9:6345           return?
5855:16DF  202  20             14CA:5322           ?????
5857:16E1  254  20             14FE:5374           move string to FPA1
5859:16E3   22  22             1616:5654           NOT??

Get number from Token Line     to FPA2
5861:16E5 EXX                  .    217
5862:16E6 DEC  C               .     13           line counter
5863:16E7 EXX                  .    217
5864:16E8 INC  DE              .     19
5865:16E9 LD   A,(DE)          .     26
5866:16EA CALL 1341:4929       .A.  205  65  19    get number

PC points here for return address table--popped to HL
5869:16ED  119  19             1377:4983           get integer digit
5871:16EF  132  19             1384:4996           get second number
5873:16F1  145  19             1391:5009           get number >255
5875:16F3  177  19             13B1:5041           load number to FPA2
5877:16F5  201  24             18C9:6345           return?
5879:16F7  208  20             14D0:5328           ?????
5881:16F9    4  21             1504:5380           move string to FPA2
5883:16FB   73  22             1649:5705           NOT??

evaluate equation
5885:16FD CALL 16E5:5861           205 229  22    load number from line
5888:1700 LD   (3F2A:16170),A       50  42  63    control byte
5891:1703 LD   H,B                  96
5892:1704 LD   L,C             i    105
5893:1705 CALL 1680:5760       ...  205 128  22    find operator
5896:1708 JP   Z,1727:5927       .'.  202        39  23
5899:170B LD   A,(BC)          .     10
5900:170C CP   (HL)            .    190           precedence of operators
5901:170D JP   NC,1727:5927      .'.  210        39  23
5904:1710 PUSH HL              .    229
5905:1711 CALL 12B5:4789       ...  205 181  18    push FPA1
5908:1714 CALL 1323:4899       .#.  205  35  19    move FPA2 to FPA1
5911:1717 CALL 16FD:5885       ...  205 253  22    repeat until done
5914:171A LD   (3F21:16161),A 2!?   50  33  63    control byte
5917:171D CALL 1305:4869       ...  205   5  19    move FPA1 to FPA2
5920:1720 CALL 12E3:4835       ...  205 227  18    pop FPA1
5923:1723 POP  HL              .    225
5924:1724 JP   170B:5899       ...  195  11  23    check previous precedence

if precedence is ok
5927:1727 LD   A,(3F21:16161) :!?   58  33  63    control byte
5930:172A OR   A               .    183           check for zero
5931:172B INC  HL              #     35
5932:172C LD   A,(HL)          .    126           get execute address
5933:172D INC  HL              #     35
5934:172E LD   H,(HL)          f    102
5935:172F LD   L,A             o    111
5936:1730 PUSH BC              .    197
5937:1731 PUSH DE              .    213
5938:1732 JP   (HL)            .    233           do it
```

```
get equation from token line
5939:1733 EXX                    .    217
5940:1734 LD   A,C               y    121
5941:1735 EXX                    .    217
5942:1736 OR   A                 .    183
5943:1737 RET  Z                 .    200                    if end of token line
5944:1738 CALL 1C83:7299         ...  205 131  28            check stack
5947:173B CALL 16CD:5837         ...  205 205  22            get number from line
5950:173E LD   (3F21:16161),A    2!?   50  33  63            control byte
5953:1741 EX   AF,AF'            .     8
5954:1742 CALL 1680:5760         ...  205 128  22            find operator
5957:1745 JP   Z,1754:5972         .T.  202          84  23      if none
5960:1748 CALL 16FD:5885         ...  205 253  22            evaluate equation
5963:174B LD   (3F21:16161),A    2!?   50  33  63            control byte
5966:174E EX   AF,AF'            .     8
5967:174F LD   A,(BC)            .     10
5968:1750 INC  A                 <     60
5969:1751 JP   NZ,1748:5960        .H.  194          72  23
5972:1754 EX   AF,AF'            .     8
5973:1755 SET  CF                7     55
5974:1756 RET                    .    201

SET UP STACK
5975:1757 POP  HL                .    225                    address of calling routin
5976:1758 LD   SP,D380:54144     l..   49 128 211            top of stack
5979:175B LD   BC,1EDC:7900      ...    1 220  30            stack underflow routine
5982:175E PUSH BC                .    197                    save it
5983:175F LD   IX,0000:0         .!.. 221  33   0   0
5987:1763 ADD  IX,SP             .9   221  57
5989:1765 PUSH IX                ..   221 229
5991:1767 POP  IY                ..   253 225                set IX IY to top of stack
5993:1769 JP   (HL)              .    233                    return

find line number for RUN
5994:176A CALL 1FCD:8141         ...  205 205  31            clear
5997:176D EXX                    .    217
5998:176E LD   HL,(3EDD:16093)   *.>   42 221  62            size of table
6001:1771 LD   DE,(3ED9:16089)   .[.>237  91 217 62 start of program
6005:1775 ADD  HL,DE             .     25
6006:1776 LD   (3EE5:16101),HL   ".>    34 229  62           start of token table
6009:1779 EX   DE,HL             .    235
6010:177A LD   A,B               x    120
6011:177B AND  00F8:248          ..   230 248
6013:177D OR   0040:64           .@   246  64
6015:177F LD   B,A               G     71

find next line number
6016:1780 BIT  6,B               .p   203 112                status byte
6018:1782 JP   Z,17A2:6050         ...  202 162  23                end of program
6021:1785 LD   DE,(3EE5:16101)   .[.>237  91 229 62 start of table
6025:1789 LD   A,L               }    125
6026:178A SUB  E                 .    147
6027:178B LD   A,H               |    124
6028:178C SBC  A,D               .    154
6029:178D JR   NC,17A2:6050        0.     48       19
6031:178F INC  HL                #     35
6032:1790 INC  HL                #     35
6033:1791 LD   E,(HL)            ^     94
6034:1792 INC  HL                #     35
6035:1793 LD   D,(HL)            V     86
6036:1794 INC  HL                #     35
6037:1795 LD   A,(DE)            .     26
6038:1796 LD   C,A               O     79
6039:1797 BIT  7,B               .x   203 120                is trace on?
6041:1799 PUSH DE                .    213
6042:179A EXX                    .    217
6043:179B POP  DE                .    209
6044:179C JR   NZ,17C2:6082      $     32  36                do trace
6046:179E RET                    .    201
```

```
END
6047:179F EXX                          .    217
6048:17A0 INC  DE                      .     19
6049:17A1 INC  DE                      .     19
6050:17A2 BIT  6,B                     .p   203 112          control byte
6052:17A4 JR   Z,17AF:6063             (.    40   9          if continue not allowed
6054:17A6 RES  2,B                     ..   203 144          continue on
6056:17A8 LD   (3EFA:16122),DE         .S.>237  83 250 62    save pointers
6060:17AC LD   (3EFC:16124),HL         ".>   34 252  62
6063:17AF EXX                          .    217
6064:17B0 INC  SP                      3     51
6065:17B1 INC  SP                      3     51
6066:17B2 JP   3EA6:16038              ..>  195 166  62      goto main loop

TRACE
6069:17B5 INC  DE                      .     19
6070:17B6 LD   A,(DE)                  .     26
6071:17B7 OR   A                       .    183
6072:17B8 PUSH DE                      .    213
6073:17B9 EXX                          .    217
6074:17BA POP  DE                      .    209
6075:17BB JR   Z,1780:6016               (.    40 195        get next line
6077:17BD LD   C,A                     O     79
6078:17BE BIT  7,B                     .x   203 120
6080:17C0 EXX                          .    217
6081:17C1 RET  Z                       .    200             trace not on
6082:17C2 EXX                          .    217
6083:17C3 BIT  6,B                     .p   203 112
6085:17C5 JR   Z,17DE:6110               (.    40          23        print screen off??
6087:17C7 PUSH DE                      .    213
6088:17C8 PUSH HL                      .    229
6089:17C9 DEC  HL                      +     43
6090:17CA DEC  HL                      +     43
6091:17CB DEC  HL                      +     43
6092:17CC DEC  HL                      +     43
6093:17CD LD   A,0023:35               >#    62  35          print '#' for trace
6095:17CF CALL 2EDA:11994              ...  205 218  46      print character
6098:17D2 XOR  A                       .    175
6099:17D3 CALL 32A1:12961              ..2  205 161  50      print CHR in (HL)
6102:17D6 LD   A,0020:32               >     62  32
6104:17D8 CALL 2EDA:11994              ...  205 218  46      write a space
6107:17DB POP  HL                      .    225
6108:17DC POP  DE                      .    209
6109:17DD LD   A,(DE)                  .     26
6110:17DE EXX                          .    217
6111:17DF RET                          .    201             back to program

EXECUTE COMMAND
6112:17E0 PUSH DE                      .    213
6113:17E1 EXX                          .    217
6114:17E2 LD   A,B                     x    120
6115:17E3 AND  009C:156                ..   230 156
6117:17E5 LD   B,A                     G     71
6118:17E6 AND  0010:16                 ..   230  16
6120:17E8 JR   Z,17F8:6136             (.    40  14
6122:17EA LD   HL,(3ED9:16089)         *.>   42 217  62      start of program
6125:17ED CALL 250D:9485               ..%  205  13  37      restore?
6128:17F0 LD   DE,(3EDD:16093)         .[.>237  91 221 62    size of table
6132:17F4 ADD  HL,DE                   .     25
6133:17F5 LD   (3EE5:16101),HL         ".>   34 229  62      start of table
6136:17F8 LD   HL,FFFF:65535           !..   33 255 255
6139:17FB POP  DE                      .    209
6140:17FC BIT  3,B                     .X   203  88
6142:17FE RES  3,B                     ..   203 152
6144:1800 EXX                          .    217
6145:1801 CALL NZ 1FD0:8144            ...  196 208  31      clear
6148:1804 EXX                          .    217
6149:1805 BIT  2,B                     .P   203  80
6151:1807 EXX                          .    217
6152:1808 CALL NZ 1757:5975            .W.  196  87  23      reset stack
6155:180B DEC  DE                      .     27
6156:180C JP   1848:6216               .H.  195  72  24      program loop
```

```
RUN or RUN line number
6159:180F CALL 1757:5975       .W.   205  87  23    clear stack
6162:1812 CALL 2E83:11907      ...   205 131  46    set random seed
6165:1815 EXX                  .     217
6166:1816 LD   A,C             y     121
6167:1817 EXX                  .     217
6168:1818 OR   A               .     183
6169:1819 PUSH DE              .     213
6170:181A PUSH AF              .     245
6171:181B CALL 176A:5994       .j.   205 106  23    find first line number
6174:181E POP  AF              .     241
6175:181F JP   Z,182E:6190       ...   202 46  24    if line specified
6178:1822 POP  DE              .     209
6179:1823 EXX                  .     217
6180:1824 RES  6,B             ..    203 176       RUN on
6182:1826 EXX                  .     217
6183:1827 LD   HL,1848:6216    !H.    33  72  24    set return address
6186:182A PUSH HL              .     229
6187:182B JP   2096:8342       ..    195 150  32    do GOTO

GET NEXT PROGRAM LINE
6190:182E POP  HL              .     225
6191:182F EXX                  .     217
6192:1830 DEC  C               .      13
6193:1831 BIT  4,B             .'    203  96       BREAK flag
6195:1833 EXX                  .     217
6196:1834 CALL Z  184E:6222    .N.   204  78  24
6199:1837 INC  DE              .      19
6200:1838 LD   A,(DE)          .      26            get token for first command
6201:1839 ADD  A,A             .     135            double it
6202:183A LD   B,0000:0        ..      6   0
6204:183C LD   C,A             O      79
6205:183D LD   HL,1915:6421    !..    33  21  25    start of token table
6208:1840 ADD  HL,BC           .       9            add line
6209:1841 LD   A,(HL)          .     126            get line address
6210:1842 INC  HL              #      35
6211:1843 LD   H,(HL)          f     102
6212:1844 LD   L,A             o     111
6213:1845 CALL 1732:5938       .2.   205  50  23    JP (HL)
6216:1848 CALL 17B5:6069       ...   205 181  23    check TRACE
6219:184B JP   182F:6191       ./.   195  47  24    main loop
6222:184E CALL 439E:17310      ..C   205 158  67    read keyboard
6225:1851 INC  A               <      60
6226:1852 RET  NZ              .     192
6227:1853 LD   HL,426B:17003   !kB    33 107  66    input byte
6230:1856 LD   A,(3F07:16135)  :.?    58   7  63    pause ASCII value
6233:1859 XOR  (HL)            .     174
6234:185A LD   (3F08:16136),A  2.?    50   8  63    pause on flag
6237:185D LD   A,(3F06:16134)  :.?    58   6  63    ASCII of BREAK
6240:1860 CP   (HL)            .     190
6241:1861 RET  NZ              .     192
6242:1862 LD   A,0013:19       >.     62  19       BREAK error message
6244:1864 JP   1F0E:7950       ...   195  14  31    print it

LET
6247:1867 CALL 199B:6555       ...   205 155  25    Get variable type
6250:186A      116 24          1874:6260            FP number
6252:186C      149 24          1895:6293            Integer
6254:186E      170 24          18AA:6314            String
6256:1870        6 31          1F06:7942            incorrect function
6258:1872        6 31          1F06:7942            incorrect function
6260:1874 PUSH BC              .     197            Floating point
6261:1875 EXX                  .     217
6262:1876 DEC  C               .      13
6263:1877 EXX                  .     217
6264:1878 INC  DE              .      19
6265:1879 CALL 1733:5939       .3.   205  51  23    get equation
6268:187C POP  HL              .     225
6269:187D LD   BC,(3F22:16162)       237  75  34  63 FPA1
6273:1881 LD   (HL),C          q     113
6274:1882 INC  HL              #      35
6275:1883 LD   (HL),B          p     112
6276:1884 INC  HL              #      35
6277:1885 LD   BC,(3F24:16164)       237  75  36  63 FPA1
6281:1889 LD   (HL),C          q     113
6282:188A INC  HL              #      35
6283:188B LD   (HL),B          p     112
6284:188C INC  HL              #      35
```

```
6285:188D LD   A,(3F26:16166) :&?   58  38  63    FPA1
6288:1890 LD   (HL),A         w     119
6289:1891 RET                 .     201
6290:1892 PUSH BC             .     197
6291:1893 JR   189D:6301      ..     24   8
6293:1895 PUSH BC             .     197           integer
6294:1896 EXX                 .     217
6295:1897 DEC  C              .      13
6296:1898 EXX                 .     217
6297:1899 INC  DE             .      19
6298:189A CALL 1733:5939      .3.   205  51  23    get equation
6301:189D CALL 0932:2354      .2.   205  50   9    LD HL with FPA1
6304:18A0 LD   B,H            D      68
6305:18A1 LD   C,L            M      77
6306:18A2 POP  HL             .     225
6307:18A3 JP   C,1F00:7936      ...   218         0  31    Illegal quantity
6310:18A6 LD   (HL),C         q     113
6311:18A7 INC  HL             #      35
6312:18A8 LD   (HL),B         p     112
6313:18A9 RET                 .     201
6314:18AA PUSH BC             .     197           string
6315:18AB EXX                 .     217
6316:18AC DEC  C              .      13
6317:18AD EXX                 .     217
6318:18AE INC  DE             .      19
6319:18AF CALL 1733:5939      .3.   205  51  23    get equation
6322:18B2 POP  BC             .     193
6323:18B3 LD   HL,(3F22:16162) *"?  42  34  63    FPA control byte
6326:18B6 LD   A,L            }     125
6327:18B7 LD   (BC),A         .       2
6328:18B8 INC  BC             .       3
6329:18B9 LD   A,H            |     124
6330:18BA LD   (BC),A         .       2
6331:18BB DEC  BC             .      11
6332:18BC LD   (HL),C         q     113
6333:18BD INC  HL             #      35
6334:18BE LD   (HL),B         p     112
6335:18BF RET                 .     201

TRACE
6336:18C0 EXX                 .     217
6337:18C1 SET  7,B            ..    203 248       control byte
6339:18C3 EXX                 .     217
6340:18C4 RET                 .     201

NOTRACE
6341:18C5 EXX                 .     217
6342:18C6 RES  7,B            ..    203 184       control byte
6344:18C8 EXX                 .     217
6345:18C9 RET                 .     201

BREAK
6346:18CA EXX                 .     217
6347:18CB RES  4,B            ..    203 160
6349:18CD EXX                 .     217
6350:18CE RET                 .     201

NOBREAK
6351:18CF EXX                 .     217
6352:18D0 SET  4,B            ..    203 224
6354:18D2 EXX                 .     217
6355:18D3 RET                 .     201

NEW
6356:18D4 CALL 1757:5975      .W.   205  87  23    Clear stack
6359:18D7 CALL 2E74:11892     .t.   205 116  46    reset pointers
6362:18DA LD   HL,001D:29     !..    33  29   0
6365:18DD LD   (3EEB:16107),HL ".>   34 235  62    number of variables
6368:18E0 CALL 1FCD:8141      ...   205 205  31    do CLEAR
6371:18E3 EXX                 .     217
6372:18E4 SET  2,B            ..    203 208       clear subroutines flag
6374:18E6 EXX                 .     217
6375:18E7 JP   3EA3:16035     ..>   195 163  62    main loop
```

```
STOP
6378:18EA EXX                          .    217
6379:18EB INC  DE                      .     19
6380:18EC INC  DE                      .     19
6381:18ED EXX                          .    217

?????
6382:18EE LD   A,0014:20               >.    62  20          Break message
6384:18F0 JP   1F0E:7950               ...  195  14  31     print it

CONT
6387:18F3 EXX                          .    217
6388:18F4 LD   A,B                     x    120
6389:18F5 AND  0044:68                .D   230   68          CONT and RUN flags
6391:18F7 JP   NZ,1911:6417              ...  194        17  25
6394:18FA SET  6,B                     ..   203 240          set RUN
6396:18FC LD   DE,(3EFA:16122)         .    237  91 250  62 program pointers
6400:1900 PUSH DE                      .    213
6401:1901 LD   HL,(3EFC:16124)        *.>    42 252  62     program pointers
6404:1904 LD   A,(DE)                  .     26
6405:1905 LD   C,A                     O     79
6406:1906 EXX                          .    217
6407:1907 POP  DE                      .    209
6408:1908 LD   A,D                     z    122
6409:1909 OR   E                       .    179
6410:190A JP   Z,1912:6418               ...  202        18  25
6413:190D DEC  DE                      .     27
6414:190E JP   1848:6216               .H.  195  72  24     check TRACE & continue
6417:1911 EXX                          .    217
6418:1912 LD   A,0015:21               >.    62  21          Can't continue
6420:1914 JP   1F0E:7950               ...  195  14  31     print it

COMMAND JUMP TABLE by token -- see 1113
6423:1917   103  24    1867   default
6425:1919   235  32    20EB   GOSUB
6427:191B   150  32    2096   GOTO
6429:191D   253  34    22FD   INPUT
6431:191F   103  24    1867   LET
6433:1921   107  34    226B   NEXT
6435:1923   174  30    1EAE   PRINT
6437:1925    27  37    251B   READ
6439:1927   227  32    2DE3   REM
6441:1929   109  33    216D   FOR
6443:192B    25  30    1E19   IF
6445:192D   227  32    20E3   DATA
6446:192E    30  27    1B1E   DIM
6449:1931   189  32    20BD   ON
6451:1933   178  31    1FB2   ONERR
6453:1935   234  24    18EA   STOP
6455:1937    29  33    211D   RETURN
6457:1939   159  23    179F   END
6459:193B    52  32    2034   DEF
6461:193D   205  31    1FCD   CLEAR
6463:193F   121  32    2079   RESUME
6465:1941   212  24    1804   NEW
6467:1943    45  33    212D   POP
6469:1945    15  24    180F   RUN
6471:1947   239  28    16EF   LIST
6473:1949   192  24    18C0   TRACE
6475:194B   197  24    18C5   NOTRACE
6477:194D   131  29    1D83   DEL
6479:194F    58  39    273A   CALL
6481:1951   243  24    18F3   CONT
6483:1953   173  31    1FAD   CLRERR
6485:1955   162  36    2A42   GET
6487:1957   120  39    2778   POKE
6489:1959    10  37    250A   RESTORE
6491:195B    82  43    2B52   HOME
6493:195D    94  44    2C5E   DRAW
6495:195F   148  44    2C94   XDRAW
6497:1961    42  43    2B2A   FLASH
6499:1963    47  43    2B2F   INVERSE
6501:1965    52  43    2B34   NORMAL
6503:1967    57  43    2B39   TEXT
6505:1969    62  43    2B3E   GR
```

```
6507:196B    67  43    2B43   HGR
6509:196D    72  43    2B48   HGR2
6511:196F   162  43    2BA2   HLIN
6513:1971   211  43    2BD#   VLIN
6515:1973   223  44    2CDF   HPLOT
6517:1975   131  43    2B83   PLOT
6519:1977    56  44    2C38   HTAB
6521:1979    66  44    2C42   VTAB
6523:197B    77  43    2B4D   SHLOAD
6525:197D   244  45    2DF4   RECALL
6527:197F   236  45    2DEC   STORE
6529:1981   142  39    278E   WAIT
6531:1983    80  42    2A50   SPEED
6533:1985   195  44    2CC3   ROT
6535:1987   209  44    2CD1   SCALE
6537:1989    91  43    2B5B   COLOR
6539:198B   111  43    2B6F   HCOLOR
6541:198D    52  47    2F34   IN
6543:198F    26  47    2F1A   PR
6545:1991     2  43    2B02   HIMEM
6547:1993   118  42    2A76   LOMEM
6549:1995   202  24    18CA   BREAK
6551:1997   207  24    186F   NOBREAK
6552:1998   180  39    27B4   &

CHECK VARIABLE TYPE
6555:199B EXX                .    217
6556:199C DEC  C             .     13
6557:199D EXX                .    217
6558:199E INC  DE            .     19
6559:199F AND  0037:55      .7   230   55        option to set CF
6561:19A1 EX   AF,AF'        .      8
6562:19A2 EX   DE,HL         .    235
6563:19A3 LD   A,(HL)        .    126
6564:19A4 AND  0003:3        ..   230   3        get # of letters in name
6566:19A6 LD   B,A           G     71
6567:19A7 INC  HL            #     35
6568:19A8 LD   C,(HL)        N     78
6569:19A9 INC  HL            #     35
6570:19AA LD   D,0000:0      ..    22   0
6572:19AC LD   E,(HL)                94
6573:19AD ADD  HL,DE         .     25
6574:19AE EX   DE,HL         .    235
6575:19AF LD   A,L           }    125
6576:19B0 EXX                .    217
6577:19B1 INC  A             <     60
6578:19B2 SUB   C            .    145
6579:19B3 CP   L,A           /     47
6580:19B4 LD   C,A           O     79
6581:19B5 EXX                ;    217
6582:19B6 LD   H,B           `     96
6583:19B7 LD   L,C           i    105
6584:19B8 ADD  HL,BC         .      9
6585:19B9 ADD  HL,HL         )     41
6586:19BA ADD  HL,BC         .      9        *5
6587:19BB LD   BC,(3EDF:16095)     237  75 223  62 Start of Variables
6591:19BF ADD  HL,BC         .      9
6592:19C0 LD   A,(HL)        .    126          variable type
6593:19C1 INC  HL            #     35
6594:19C2 LD   C,(HL)        N     78          address of variable
6595:19C3 INC  HL            #     35
6596:19C4 LD   B,(HL)        F     70
6597:19C5 INC  B             .      4
6598:19C6 DEC  B             .      5
6599:19C7 JR   Z,19F6:6646     (-    40        45
6601:19C9 AND  00FC:252      ..   230 252        strip length bits
6603:19CB JR   NZ,19D5:6613        .   32        8
6605:19CD POP  HL            .    225          PC from calling routine
6606:19CE EX   AF,AF'        .      8
6607:19CF LD   A,(HL)        .    126
6608:19D0 INC  HL            #     35
6609:19D1 LD   H,(HL)        f    102
6610:19D2 LD   L,A           o    111
6611:19D3 EX   AF,AF'        .      8
6612:19D4 JP   (HL)          .    233          return
6613:19D5 JP   M,19EB:6635     ...  250 235  25
```

```
6616:19D8 BIT  3,A              _    203  95          DIM variable
6618:19DA JP   NZ,1A7D:6781     _ .}.  194 125  26
6621:19DD BIT  5,A              .o   203 111          String
6623:19DF DEC  HL               +     43
6624:19E0 LD   B,H              D     68
6625:19E1 LD   C,L              M     77
6626:19E2 POP  HL               .    225
6627:19E3 INC  HL               #     35
6628:19E4 INC  HL               #     35
6629:19E5 JR   Z,19CE:6606        (.    40 231
6631:19E7 INC  HL               #     35
6632:19E8 INC  HL               #     35
6633:19E9 JR   19CE:6606        ..    24 227
6635:19EB POP  HL               .    225
6636:19EC PUSH BC               .    197
6637:19ED LD   BC,0006:6        ...    1   6   0    4th address (MATH)
6640:19F0 ADD  HL,BC            .      9
6641:19F1 POP  BC               .    193
6642:19F2 BIT  6,A              .w   203 119          FN variable (5th address)
6644:19F4 JR   19E5:6629        ..    24 239          check for it
6646:19F6 AND  00FC:252         ..   230 252          strrip length bits
6648:19F8 JR   NZ,1A17:6679           .    32      29
6650:19FA PUSH HL               .    229
6651:19FB LD   HL,0005:5        !..   33   5   0
6654:19FE CALL 1C8F:7311        ...  205 143  28    check string space
6657:1A01 LD   HL,(3EED:16109)  *.>   42 237  62    start of variable table
6660:1A04 LD   (HL),0000:0      6.    54   0
6662:1A06 DEC  HL               +     43
6663:1A07 DEC  HL               +     43
6664:1A08 DEC  HL               +     43
6665:1A09 DEC  HL               +     43
6666:1A0A LD   B,H              D     68
6667:1A0B LD   C,L              M     77
6668:1A0C DEC  HL               +     43
6669:1A0D LD   (3EED:16109),HL ".>   34 237  62    start of variable table
6672:1A10 POP  HL               .    225
6673:1A11 LD   (HL),B           p    112
6674:1A12 DEC  HL               +     43
6675:1A13 LD   (HL),C           q    113
6676:1A14 XOR  A                .    175
6677:1A15 JR   19CD:6605        ..    24 182          get offset and go
6679:1A17 JP   M,19EB:6635        ...  250 235  25          use 4th or 5th address
6682:1A1A BIT  3,A              .    203  95          DIM variable
6684:1A1C JR   NZ,1A49:6729     _  +    32       43
6686:1A1E BIT  5,A              .o   203 111          String
6688:1A20 JR   Z,19DF:6623        (.    40 189        Get offset
6690:1A22 LD   BC,3F52:16210    .R?    1  82  63    Null string pointer
6693:1A25 LD   (HL),B           p    112
6694:1A26 DEC  HL               +     43
6695:1A27 LD   (HL),C           q    113
6696:1A28 JR   19E0:6624        ..    24 182
6698:1A2A EX   DE,HL            .    235
6699:1A2B LD   A,(HL)           .    126
6700:1A2C AND  0003:3           ..   230   3          length of variable name
6702:1A2E LD   B,A              G     71
6703:1A2F INC  HL               #     35
6704:1A30 LD   C,(HL)           N     78
6705:1A31 INC  HL               #     35
6706:1A32 LD   D,0000:0         .     22   0
6708:1A34 LD   E,(HL)           ^     94
6709:1A35 ADD  HL,DE            .     25
6710:1A36 EX   DE,HL            .    235
6711:1A37 LD   A,L              }    125
6712:1A38 EXX                   .    217
6713:1A39 INC  A                <     60
6714:1A3A SUB  C                .    145
6715:1A3B CP   L,A              /     47
6716:1A3C LD   C,A              O     79
6717:1A3D EXX                   .    217
6718:1A3E LD   H,B              `     96
6719:1A3F LD   L,C              i    105
6720:1A40 ADD  HL,BC            .      9
6721:1A41 ADD  HL,HL            )     41
6722:1A42 ADD  HL,BC            .      9
6723:1A43 LD   BC,(3EDF:16095) .K.>237  75 223  62 LOMEM setting
6727:1A47 ADD  HL,BC            .      9
6728:1A48 RET                   .    201
```

```
do DIM array
6729:1A49 PUSH DE              .    213
6730:1A4A PUSH HL              .    229
6731:1A4B PUSH AF              .    245
6732:1A4C CALL 1C05:7173       ...  205   5  28    check data length
6735:1A4F POP  AF              .    241
6736:1A50 LD   B,C             A     65             size of array
6737:1A51 LD   HL,0001:1       !..   33   1   0
6740:1A54 LD   DE,000B:11      ...   17  11   0    minimum for each DIM
6743:1A57 CALL 1B9A:7066       ...  205 154  27    HL * DE
6746:1A5A JP   C,1EF7:7927       ...  218 247  30          out of memory
6749:1A5D CALL 1C83:7299       ...  205 131  28    check stack
6752:1A60 PUSH DE              .    213
6753:1A61 DJNZ 1A57:6743       ..    16 244
6755:1A63 LD   E,A                   95             DIM array size
6756:1A64 AND  0030:48        .0   230   48
6758:1A66 LD   A,E             {    123
6759:1A67 LD   DE,0002:2       ...   17   2   0
6762:1A6A JR   NZ,1A6F:6767          .    32        3        use specified size
6764:1A6C LD   DE,0005:5       ...   17   5   0    use default size
6767:1A6F CALL 1B9A:7066       ...  205 154  27    HL * DE
6770:1A72 JP   C,1EF7:7927       ...  218 247  30          out of memory
6773:1A75 CALL 1BB9:7097       ...  205 185  27    continue
6776:1A78 POP  HL              .    225
6777:1A79 LD   (HL),B          p    112
6778:1A7A DEC  HL              +     43
6779:1A7B LD   (HL),C          q    113
6780:1A7C POP  DE              .    209
6781:1A7D PUSH AF              .    245
6782:1A7E EXX                  .    217
6783:1A7F DEC  C               .     13
6784:1A80 EXX                  .    217
6785:1A81 INC  DE              .     19
6786:1A82 LD   A,(DE)          .     26
6787:1A83 CP   00B7:183        ..   254 183
6789:1A85 JP   NZ,1B0D:6925      ...  194       13  27
6792:1A88 EX   AF,AF'          .      8
6793:1A89 JR   C,1AAD:6829      8"    56       34
6795:1A8B CALL 1ABE:6846       ...  205 190  26
6798:1A8E POP  AF              .    241
6799:1A8F PUSH DE              .    213
6800:1A90 LD   D,A             W     87
6801:1A91 EX   AF,AF'          .      8
6802:1A92 LD   A,D             z    122
6803:1A93 AND  0030:48        .0   230   48
6805:1A95 LD   D,H             T     84
6806:1A96 LD   E,L             ]     93
6807:1A97 JP   NZ,1A9C:6812      ...  194 156  26
6810:1A9A ADD  HL,DE           .     25
6811:1A9B ADD  HL,HL           )     41
6812:1A9C ADD  HL,DE           .     25
6813:1A9D ADD  HL,BC           .      9
6814:1A9E LD   B,H             D     68
6815:1A9F LD   C,L             M     77
6816:1AA0 POP  DE              .    209
6817:1AA1 JR   NZ,1AA7:6823          .    32        4
6819:1AA3 EX   AF,AF'          .      8
6820:1AA4 JP   19CD:6605       ...  195 205  25    exit to execute routine
6823:1AA7 EX   AF,AF'          .      8
6824:1AA8 BIT  5,A             .o   203 111
6826:1AAA JP   19E2:6626       ...  195 226  25    get exit address
6829:1AAD CALL 12B5:4789       ...  205 181  18    push FPA1
6832:1AB0 CALL 1ABE:6846       ...  205 190  26
6835:1AB3 LD   (3F0B:16139),HL ".?   34  11  63    storage
6838:1AB6 CALL 12E3:4835       ...  205 227  18    pop FPA1
6841:1AB9 LD   HL,(3F0B:16139) *.?   42  11  63
6844:1ABC JR   1A8E:6798       ..    24 208
6846:1ABE LD   HL,0000:0       !..   33   0   0
6849:1AC1 PUSH HL              .    229
6850:1AC2 LD   H,B             `     96
6851:1AC3 LD   L,C             i    105
6852:1AC4 LD   B,(HL)          F     70
6853:1AC5 INC  HL              #     35
6854:1AC6 JP   1AD3:6867       ...  195 211  26
6857:1AC9 EX   (SP),HL         .    227
```

```
6858:1ACA EXX                    .     217
6859:1ACB DEC  C                 .      13
6860:1ACC EXX                    .     217
6861:1ACD INC  DE                .      19
6862:1ACE LD   A,(DE)            .      26
6863:1ACF CP   00B9:185          ..    254 185
6865:1AD1 JR   NZ,1B0D:6925      :      32  58
6867:1AD3 PUSH BC                .     197
6868:1AD4 PUSH HL                .     229
6869:1AD5 CALL 1733:5939         .3.   205  51  23    get equation
6872:1AD8 JR   NC,1B0D:6925      03     48  51
6874:1ADA OR   A                 .     183
6875:1ADB JR   NZ,1B0D:6925       0     32  48

6877:1ADD POP  HL                .     225
6878:1ADE POP  BC                .     193
6879:1ADF LD   C,E               K      75
6880:1AE0 LD   A,D               z     122
6881:1AE1 LD   E,(HL)            ^      94
6882:1AE2 INC  HL                #      35
6883:1AE3 LD   D,(HL)            V      86
6884:1AE4 INC  HL                #      35
6885:1AE5 EX   (SP),HL           .     227
6886:1AE6 CALL 1B9A:7066         ...   205 154  27    HL * DE
6889:1AE9 JR   C,1B10:6928       8%     56  37
6891:1AEB PUSH HL                .     229
6892:1AEC PUSH AF                .     245
6893:1AED CALL 0932:2354         .2.   205  50   9    LD HL with FPA1
6896:1AF0 JR   C,1B10:6928         8.      56        30
6898:1AF2 POP  AF                .     241
6899:1AF3 INC  H                 $      36
6900:1AF4 DEC  H                 %      37
6901:1AF5 JP   M,1B10:6928           ...   250        16  27
6904:1AF8 EX   DE,HL             .     235
6905:1AF9 DEC  HL                +      43
6906:1AFA SBC  HL,DE             .R    237  82
6908:1AFC JR   C,1B10:6928         8.      56        18
6910:1AFE POP  HL                .     225
6911:1AFF ADD  HL,DE             .      25
6912:1B00 LD   D,A               W      87
6913:1B01 LD   E,C               Y      89
6914:1B02 DJNZ 1AC9:6857         ..     16 197
6916:1B04 EXX                    .     217
6917:1B05 DEC  C                 .      13
6918:1B06 EXX                    .     217
6919:1B07 INC  DE                .      19
6920:1B08 LD   A,(DE)            .      26
6921:1B09 CP   00B8:184          ..    254 184
6923:1B0B POP  BC                .     193
6924:1B0C RET  Z                 .     200
6925:1B0D LD   A,0001:1          >.     62   1        error code
6927:1B0F LD   HL,093E:2366      !>.    33  62   9    skip next
6928:1B10 LD   A,09:9 only if JP 1B10   62   9    error code
6930:1B12 JP   1F0E:7950         ...    195  14  31    print it

DIM...enters at 1B1E
6933:1B15 EXX                    .     217
6934:1B16 DEC  C                 .      13
6935:1B17 EXX                    .     217
6936:1B18 INC  DE                .      19
6937:1B19 LD   A,(DE)            .      26
6938:1B1A CP   00B9:185          ..    254 185
6940:1B1C JR   NZ,1B0D:6925            .    32 239
```

```
DIM execution
6942:1B1E EXX                  .    217
6943:1B1F DEC  C                .     13
6944:1B20 EXX                   .    217
6945:1B21 INC  DE               .     19
6946:1B22 CALL 1A2A:6698        .*.  205  42  26
6949:1B25 LD   B,(HL)           F     70
6950:1B26 INC  HL               #     35
6951:1B27 LD   A,(HL)           .    126
6952:1B28 INC  HL               #     35
6953:1B29 OR   (HL)             .    182
6954:1B2A JP   NZ,1EF4:7924       ...  194 244  30          redimensioned array
6957:1B2D OR   B                .    176
6958:1B2E JP   M,1F06:7942        ...  250         6  31    incorrect function
6961:1B31 DEC  HL               +     43
6962:1B32 PUSH HL               .    229
6963:1B33 EXX                   .    217
6964:1B34 DEC  C                .     13
6965:1B35 EXX                   .    217
6966:1B36 INC  DE               .     19
6967:1B37 LD   A,(DE)           .     26
6968:1B38 CP   00B7:183         ..   254 183
6970:1B3A JR   NZ,1B0D:6925           .    32 209     error
6972:1B3C LD   HL,0001:1        !..   33   1   0
6975:1B3F LD   C,0000:0         ..    14   0
6977:1B41 PUSH BC               .    197
6978:1B42 PUSH HL               .    229
6979:1B43 CALL 1733:5939        .3.  205  51  23     get equation
6982:1B46 POP  HL               .    225
6983:1B47 POP  BC               .    193
6984:1B48 JR   NC,1B10:6928       0.     48 198     error
6986:1B4A OR   A                .    183
6987:1B4B JR   NZ,1B10:6928           .    32 195     error
6989:1B4D CALL 1C83:7299        ...  205 131  28     check stack
6992:1B50 PUSH DE               .    213
6993:1B51 EX   DE,HL            .    235
6994:1B52 CALL 0932:2354        .2.  205  50   9     LD HL with FPA1
6997:1B55 EX   DE,HL            .    235
6998:1B56 JR   C,1B10:6928        8.     56 184     error
7000:1B58 INC  D                .     20
7001:1B59 DEC  D                .     21
7002:1B5A JP   M,1B10:6928        ...  250        16  27    error
7005:1B5D INC  DE               .     19
7006:1B5E CALL 1B9A:7066        ...  205 154  27     HL * DE
7009:1B61 JP   C,1EF7:7927        ...  218 247  30          out of memory
7012:1B64 EX   DE,HL            .    235
7013:1B65 EX   (SP),HL          .    227
7014:1B66 EX   DE,HL            .    235
7015:1B67 INC  C                .     12
7016:1B68 EXX                   .    217
7017:1B69 DEC  C                .     13
7018:1B6A EXX                   .    217
7019:1B6B INC  DE               .     19
7020:1B6C LD   A,(DE)           .     26
7021:1B6D CP   00B9:185         ..   254 185
7023:1B6F JP   Z,1B41:6977        .A.  202        65  27
7026:1B72 CP   00B8:184         ..   254 184
7028:1B74 JR   NZ,1B0D:6925           .    32 151     error
7030:1B76 LD   A,B              x    120
7031:1B77 PUSH DE               .    213
7032:1B78 LD   E,A                    95
7033:1B79 AND  0030:48          .0   230   48
7035:1B7B LD   A,E              {    123
7036:1B7C LD   DE,0002:2        ...   17   2   0
7039:1B7F JR   NZ,1B84:7044           .    32        3
7041:1B81 LD   DE,0005:5        ...   17   5   0
7044:1B84 CALL 1B9A:7066        ...  205 154  27     HL * DE
7047:1B87 JP   C,1EF7:7927        ...  218 247  30          out of memory
7050:1B8A POP  DE               .    209
7051:1B8B CALL 1BB9:7097        ...  205 185  27
7054:1B8E POP  HL               .    225
7055:1B8F LD   (HL),C           q    113
7056:1B90 INC  HL               #     35
7057:1B91 LD   (HL),B           p    112
7058:1B92 EXX                   .    217
7059:1B93 LD   A,C              y    121
7060:1B94 EXX                   .    217
7061:1B95 OR   A                .    183
7062:1B96 RET  Z                .    200
```

```
7063:1B97 JP   1B15:6933       ...  195  21  27

HL * DE
7066:1B9A PUSH DE              .    213
7067:1B9B PUSH BC              .    197
7068:1B9C PUSH AF              .    245
7069:1B9D LD   C,H             L     76
7070:1B9E LD   A,L             }    125
7071:1B9F LD   HL,0000:0       !..   33   0   0
7074:1BA2 LD   B,0010:16       ..     6  16        shifts required
7076:1BA4 SRL  C               .9   203  57
7078:1BA6 RR   A               .     31
7079:1BA7 JP   NC,1BAD:7085    ...  210 173  27
7082:1BAA ADD  HL,DE           .     25
7083:1BAB JR   C,1BB4:7092     8.    56   7        overflow
7085:1BAD EX   DE,HL           .    235
7086:1BAE ADD  HL,HL           )     41            double DE
7087:1BAF EX   DE,HL           .    235
7088:1BB0 DJNZ 1BA4:7076       ..    16 242
7090:1BB2 LD   A,H             |    124
7091:1BB3 RL   A               .     23
7092:1BB4 POP  BC              .    193
7093:1BB5 LD   A,B             x    120
7094:1BB6 POP  BC              .    193
7095:1BB7 POP  DE              .    209
7096:1BB8 RET                  .    201

more DIM
7097:1BB9 EX   (SP),HL         .    227
7098:1BBA LD   (3F0D:16141),HL ".?   34  13  63     storage
7101:1BBD LD   (3F0B:16139),DE      237  83  11  63 storage
7105:1BC1 POP  HL              .    225
7106:1BC2 LD   B,C             A     65
7107:1BC3 PUSH BC              .    197
7108:1BC4 PUSH AF              .    245
7109:1BC5 LD   A,C             y    121
7110:1BC6 SET  CF              7     55
7111:1BC7 ADC A,A              .    143
7112:1BC8 LD   C,A             O     79            C*2+1
7113:1BC9 LD   B,0000:0        ..     6   0
7115:1BCB ADD  HL,BC           .      9
7116:1BCC CALL 1C8F:7311       ...  205 143  28    check string space
7119:1BCF OR   A               .    183           clear carry
7120:1BD0 SBC  HL,BC           .B   237  66
7122:1BD2 POP  AF              .    241
7123:1BD3 BIT  5,A             .o   203 111
7125:1BD5 LD   B,H             D     68
7126:1BD6 LD   C,L             M     77
7127:1BD7 LD   HL,(3EED:16109) *.>    42 237  62   start of variable table
7130:1BDA LD   DE,0000:0       ...   17   0   0
7133:1BDD JR   Z,1BE2:7138       (.    40        3
7135:1BDF LD   DE,3F52:16210   .R?   17  82  63    null string
7138:1BE2 LD   (HL),D          r    114
7139:1BE3 CPD                  ..   237 169
7141:1BE5 JP   PO,1BEE:7150      ...  226 238  27
7144:1BE8 LD   (HL),E          s    115
7145:1BE9 CPD                  ..   237 169
7147:1BEB JP   PE,1BE2:7138      ...  234 226  27         until match found
7150:1BEE POP  BC              .    193
7151:1BEF POP  DE              .    209
7152:1BF0 LD   (HL),D          r    114
7153:1BF1 DEC  HL              +     43
7154:1BF2 LD   (HL),E          s    115
7155:1BF3 DEC  HL              +     43
7156:1BF4 DJNZ 1BEF:7151       ..    16 249
7158:1BF6 LD   (HL),C          q    113
7159:1BF7 LD   B,H             D     68
7160:1BF8 LD   C,L             M     77
7161:1BF9 DEC  HL              +     43
7162:1BFA LD   (3EED:16109),HL ".>   34 237  62    start of variable table
7165:1BFD LD   DE,(3F0B:16139)      237  91  11  63 get back
7169:1C01 LD   HL,(3F0D:16141) *.?   42  13  63              from storage
7172:1C04 JP   (HL)            .    233
```

```
Check length of DATA--also used by DIM for array size
7173:1C05 LD   BC,0000:0      ...    1   0   0
7176:1C08 JR   1C12:7186      ..    24   8
7178:1C0A INC  DE             .     19
7179:1C0B LD   A,(DE)         .     26
7180:1C0C INC  DE             .     19
7181:1C0D ADD  A,E            .    131
7182:1C0E LD   E,A                  95
7183:1C0F JR   NC,1C12:7186   -  0.    48        1
7185:1C11 INC  D              .     20
7186:1C12 LD   A,(DE)         .     26
7187:1C13 INC  DE             .     19
7188:1C14 CP   00B9:185       ..   254 185
7190:1C16 JR   Z,1C49:7241      (1    40       49
7192:1C18 CP   00B8:184       ..   254 184
7194:1C1A JR   Z,1C5B:7259      (?    40       63
7196:1C1C CP   00B7:183       ..   254 183
7198:1C1E JR   Z,1C53:7251      (3    40       51
7200:1C20 CP   008A:138       ..   254 138
7202:1C22 JP   C,1C12:7186      ...  218       18  28    skip token
7205:1C25 JR   Z,1C45:7237      (.    40       30        skip 2 element token
7207:1C27 CP   008B:139       ..   254 139
7209:1C29 JR   Z,1C44:7236      (.    40       25        skip 3 element token
7211:1C2B CP   0090:144       ..   254 144
7213:1C2D JR   C,1C0A:7178    8.    56 219      next element
7215:1C2F CP   0092:146       ..   254 146
7217:1C31 JR   Z,1C41:7233      (.    40       14        skip 6 element token
7219:1C33 JP   NC,1C12:7186      ... 210       18  28
7222:1C36 LD   A,(DE)         .     26
7223:1C37 INC  DE             .     19
7224:1C38 ADD  A,E            .    131
7225:1C39 LD   E,A                  95
7226:1C3A JP   NC,1C12:7186   -  ...  210       18  28
7229:1C3D INC  D              .     20
7230:1C3E JP   1C12:7186      ...  195  18  28
7233:1C41 INC  DE             .     19
7234:1C42 INC  DE             .     19
7235:1C43 INC  DE             .     19
7236:1C44 INC  DE             .     19
7237:1C45 INC  DE             .     19
7238:1C46 JP   1C12:7186      ...  195  18  28
7241:1C49 LD   A,B            x    120
7242:1C4A DEC  A              =     61
7243:1C4B JR   NZ,1C12:7186       .   32 197
7245:1C4D INC  C              .     12
7246:1C4E JR   NZ,1C12:7186       .   32 194
7248:1C50 JP   1B0D:6925      ...  195  13  27    error
7251:1C53 INC  B              .      4
7252:1C54 JR   NZ,1C12:7186       .   32 188
7254:1C56 LD   A,000E:14      >.    62  14        error number
7256:1C58 JP   1F0E:7950      ...  195  14  31    print it
7259:1C5B DEC  B              .      5
7260:1C5C JR   NZ,1C12:7186       .   32 180
7262:1C5E INC  C              .     12
7263:1C5F RET                 .    201

DEFINE STRING
7264:1C60 PUSH BC             .    197
7265:1C61 PUSH HL             .    229
7266:1C62 LD   C,A            0     79            length of string
7267:1C63 LD   H,0000:0       &.    38   0
7269:1C65 ADD  A,0003:3       ..   198   3        + length byte & address
7271:1C67 LD   L,A            o    111
7272:1C68 JR   NC,1C6B:7275     0.    48        1
7274:1C6A INC  H              $     36
7275:1C6B CALL 1C8F:7311      ...  205 143  28    check string space
7278:1C6E LD   A,C            y    121
7279:1C6F LD   BC,(3EEF:16111)     237  75 239  62 end of string space
7283:1C73 ADD  HL,BC          .      9
7284:1C74 LD   (3EEF:16111),HL ".>  34 239  62    new end marker
7287:1C77 LD   H,B            `     96
7288:1C78 LD   L,C            i    105            start to write here
7289:1C79 POP  BC             .    193
7290:1C7A LD   (HL),C         q    113            write address
7291:1C7B INC  HL             #     35
7292:1C7C LD   (HL),B         p    112
7293:1C7D INC  HL             #     35
7294:1C7E LD   (HL),A         w    119            write length
```

```
7295:1C7F DEC  HL              +     43
7296:1C80 DEC  HL              +     43
7297:1C81 POP  BC              .    193
7298:1C82 RET                  .    201

CHECK STACK (200 levels)
7299:1C83 PUSH HL              .    229
7300:1C84 LD   HL,2E00:11776   !..   33   0  46
7303:1C87 ADD  HL,SP           9     57
7304:1C88 POP  HL              .    225
7305:1C89 RET  C               .    216              if SP >D1FF
7306:1C8A LD   A,0007:7        >.    62   7          error number
7308:1C8C JP   1F0E:7950       ...  195  14  31      print it

CHECK STRING SPACE
7311:1C8F PUSH BC              .    197
7312:1C90 PUSH HL              .    229
7313:1C91 LD   BC,(3EEF:16111)      237  75 239  62 end of string space
7317:1C95 ADD  HL,BC           .      9
7318:1C96 JP   C,1CA4:7332       ...  218 164  28
7321:1C99 LD   BC,(3EED:16109)      237  75 237  62 start of variable table
7325:1C9D SBC  HL,BC           .B   237  66
7327:1C9F POP  HL              .    225
7328:1CA0 POP  BC              .    193
7329:1CA1 RET  C               .    216
7330:1CA2 PUSH BC              .    197
7331:1CA3 PUSH HL              .    229
7332:1CA4 CALL 27E1:10209      ..'  205 225  39      do FRE
7335:1CA7 POP  HL              .    225
7336:1CA8 PUSH HL              .    229
7337:1CA9 LD   BC,(3EEF:16111)      237  75 239  62 try again
7341:1CAD ADD  HL,BC           .      9
7342:1CAE JP   C,1EF7:7927       ...  218 247  30          out of memory
7345:1CB1 LD   BC,(3EED:16109)      237  75 237  62 start of variable table
7349:1CB5 SBC  HL,BC           .B   237  66
7351:1CB7 POP  HL              .    225
7352:1CB8 POP  BC              .    193
7353:1CB9 RET  C               .    216
7354:1CBA JP   1EF7:7927       ...  195 247  30      out of memory

LIST PROGRAM
7357:1CBD PUSH DE              .    213
7358:1CBE CALL 2F60:12128      .'/  205  96  47      print a <CR>
7361:1CC1 LD   HL,(3ED9:16089) *.>   42 217  62      start of table
7364:1CC4 LD   BC,(3EDB:16091)      237  75 219  62 number of lines
7368:1CC8 JR   1CD5:7381       ..    24  11
7370:1CCA CALL 3493:13459      ..4  205 147  52      print line
7373:1CCD PUSH BC              .    197
7374:1CCE PUSH HL              .    229
7375:1CCF CALL 184E:6222       .N.  205  78  24      ^S or ^C
7378:1CD2 POP  HL              .    225
7379:1CD3 POP  BC              .    193
7380:1CD4 DEC  BC              .     11              one less line to print
7381:1CD5 LD   A,B             x    120
7382:1CD6 OR   C               .    177
7383:1CD7 JR   NZ,1CCA:7370      .    32 241        if lines left
7385:1CD9 POP  DE              .    209
7386:1CDA RET                  .    201

LIST -- entry point at 1CEF
7387:1CDB EXX                  .    217
7388:1CDC LD   A,C             y    121
7389:1CDD EXX                  .    217
7390:1CDE OR   A               .    183
7391:1CDF JR   Z,1CBD:7357       (.    40 220       list whole program
7393:1CE1 EXX                  .    217
7394:1CE2 DEC  C               .     13
7395:1CE3 EXX                  .    217
7396:1CE4 INC  DE              .     19
7397:1CE5 LD   A,(DE)          .     26
7398:1CE6 CALL 05EF:1519       ...  205 239   5      LD BC with token line #
7401:1CE9 LD   HL,0000:0       !..   33   0   0
7404:1CEC JP   1D29:7465       .).   195  41  29
7407:1CEF EXX                  .    217
7408:1CF0 LD   A,C             y    121
7409:1CF1 EXX                  .    217
7410:1CF2 OR   A               .    183
7411:1CF3 JR   Z,1CBD:7357       (.    40 200       list whole program
```

```
7413:1CF5 LD   HL,(3EDB:16091) *.>  42 219  62    number of lines
7416:1CF8 LD   A,H              |    124
7417:1CF9 OR   L                .    181
7418:1CFA JP   Z,20E3:8419        ..   202 227  32          ignore command
7421:1CFD EXX                   .    217
7422:1CFE DEC  C                .     13
7423:1CFF EXX                   .    217
7424:1D00 INC  DE               .     19
7425:1D01 LD   A,(DE)           .     26
7426:1D02 CP   00B9:185         ..   254 185          , separator
7428:1D04 JR   Z,1CDB:7387        (.    40 213
7430:1D06 CP   00A1:161         ..   254 161          - separator
7432:1D08 JR   Z,1CDB:7387        (.    40 209
7434:1D0A CALL 05EF:1519          ...  205 239   5    LD BC with token line #
7437:1D0D EXX                   .    217
7438:1D0E LD   A,C              y    121
7439:1D0F EXX                   .    217
7440:1D10 OR   A                .    183
7441:1D11 JP   Z,1D6F:7535        .o.  202 111  29          find line
7444:1D14 EXX                   .    217
7445:1D15 DEC  C                .     13
7446:1D16 EXX                   .    217
7447:1D17 INC  DE               .     19
7448:1D18 LD   H,B              `     96
7449:1D19 LD   L,C              i    105
7450:1D1A EXX                   .    217
7451:1D1B LD   A,C              y    121
7452:1D1C EXX                   .    217
7453:1D1D OR   A                .    183
7454:1D1E JP   Z,1D7D:7549        .}.  202 125  29          list to end
7457:1D21 EXX                   .    217
7458:1D22 DEC  C                .     13
7459:1D23 EXX                   .    217
7460:1D24 INC  DE               .     19
7461:1D25 LD   A,(DE)           .     26
7462:1D26 CALL 05EF:1519          ...  205 239   5    LD BC with token line #
7465:1D29 PUSH HL               .    229
7466:1D2A OR   A                .    183
7467:1D2B SBC  HL,BC            .B   237  66
7469:1D2D POP  HL               .    225
7470:1D2E JP   Z,1D6F:7535        .o.  202 111  29          find first line
7473:1D31 RET  NC               .    208
7474:1D32 LD   (3F4F:16207),HL "O?  34  79  63    line for GOTO
7477:1D35 PUSH BC               .    197
7478:1D36 CALL 30F0:12528       ..0  205 240  48    find it
7481:1D39 POP  BC               .    193
7482:1D3A LD   (3F4F:16207),BC      237  67  79  63 line for GOTO
7486:1D3E PUSH HL               .    229
7487:1D3F CALL 30F0:12528       ..0  205 240  48    find it
7490:1D42 POP  BC               .    193
7491:1D43 LD   A,L              }    125            exchange BC and HL
7492:1D44 LD   L,C              i    105
7493:1D45 LD   C,A              O     79            why not PUSH PUSH POP POP
7494:1D46 LD   A,H              |    124
7495:1D47 LD   H,B              `     96
7496:1D48 LD   B,A              G     71
7497:1D49 JP   C,1D57:7511        .W.  218        87  29
7500:1D4C PUSH HL               .    229
7501:1D4D OR   A                .    183
7502:1D4E SBC  HL,BC            .B   237  66
7504:1D50 POP  HL               .    225
7505:1D51 JR   Z,1D73:7539        (     40        32
7507:1D53 DEC  BC               .     11
7508:1D54 DEC  BC               .     11
7509:1D55 DEC  BC               .     11
7510:1D56 DEC  BC               .     11
7511:1D57 PUSH DE               .    213
7512:1D58 PUSH BC               .    197
7513:1D59 PUSH HL               .    229
7514:1D5A CALL 184E:6222        .N.  205  78  24    check ^S ^C
7517:1D5D POP  HL               .    225
7518:1D5E POP  BC               .    193
7519:1D5F CALL 3493:13459       ..4  205 147  52    print line
7522:1D62 PUSH HL               .    229
7523:1D63 OR   A                .    183
7524:1D64 SBC  HL,BC            .B   237  66
7526:1D66 POP  HL               .    225
7527:1D67 JP   C,1D58:7512        .X.  218        88  29
7530:1D6A JP   Z,1D58:7512        .X.  202        88  29
```

```
7533:1D6D POP  DE                .    209
7534:1D6E RET                    .    201
7535:1D6F LD   (3F4F:16207),BC        237  67  79  631line for GOTO
7539:1D73 CALL 30F0:12528       ..0  205 240  48    find it
7542:1D76 RET  NC                .    208
7543:1D77 PUSH DE                .    213
7544:1D78 CALL 3493:13459       ..4  205 147  52    print it
7547:1D7B POP  DE                .    209
7548:1D7C RET                    .    201

?????
7549:1D7D LD   BC,FFFF:65535    ...    1 255 255    list to end
7552:1D80 JP   1D29:7465        .).  195  41  29    do it
7555:1D83 EXX                    .    217
7556:1D84 LD   A,C              y     121
7557:1D85 EXX                    .    217
7558:1D86 OR   A                 .    183
7559:1D87 RET  Z                 .    200              if no line specified
7560:1D88 LD   HL,(3EDB:16091) *.>    42 219  62    # of program lines
7563:1D8B LD   A,H              |     124
7564:1D8C OR   L                 .    181
7565:1D8D JP   Z,20E3:8419        ..    202 227  32          ignore if none
7568:1D90 EXX                    .    217
7569:1D91 DEC  C                 .     13
7570:1D92 EXX                    .    217
7571:1D93 INC  DE                .     19
7572:1D94 LD   A,(DE)            .     26
7573:1D95 CP   00B9:185         ..    254 185          - separator
7575:1D97 JP   Z,1F0C:7948        ...   202        12  31    syntax error
7578:1D9A CP   00A1:161         ..    254 161          , separator
7580:1D9C JP   Z,1F0C:7948        ...   202        12  31    syntax error
7583:1D9F CALL 05EF:1519        ...   205 239   5    LD BC with token line #
7586:1DA2 EXX                    .    217
7587:1DA3 LD   A,C              y     121
7588:1DA4 EXX                    .    217
7589:1DA5 OR   A                 .    183
7590:1DA6 JP   Z,1E0F:7695        ...   202        15  30    delete 1 line
7593:1DA9 LD   H,B              `      96
7594:1DAA LD   L,C              i     105
7595:1DAB EXX                    .    217
7596:1DAC DEC  C                 .     13
7597:1DAD LD   A,C              y     121
7598:1DAE DEC  C                 .     13
7599:1DAF EXX                    .    217
7600:1DB0 OR   A                 .    183
7601:1DB1 INC  DE                .     19
7602:1DB2 INC  DE                .     19
7603:1DB3 JP   Z,1F0C:7948        ...   202        12  31    sysntax error
7606:1DB6 LD   A,(DE)            .     26
7607:1DB7 CALL 05EF:1519        ...   205 239   5    LD BC with token line #
7610:1DBA LD   A,B              x     120
7611:1DBB LD   B,H              D      68
7612:1DBC LD   H,A              g     103
7613:1DBD LD   A,C              y     121
7614:1DBE LD   C,L              M      77
7615:1DBF LD   L,A              o     111
7616:1DC0 PUSH HL                .    229
7617:1DC1 OR   A                 .    183
7618:1DC2 SBC  HL,BC            .B    237  66
7620:1DC4 POP  HL                .    225
7621:1DC5 JP   Z,1E0F:7695        ...   202        15  30    delete 1 line
7624:1DC8 RET  C                 .    216
7625:1DC9 PUSH DE                .    213
7626:1DCA PUSH HL                .    229
7627:1DCB PUSH BC                .    197
7628:1DCC LD   (3F4F:16207),HL "O?    34  79  63    line for GOTO
7631:1DCF CALL 30F0:12528       ..0  205 240  48    find it
7634:1DD2 JR   NC,1E04:7684       00    48        48
7636:1DD4 JR   1DFA:7674        .$     24  36
7638:1DD6 PUSH HL                .    229
7639:1DD7 PUSH BC                .    197
7640:1DD8 LD   (3F4F:16207),HL "O?    34  79  63    line for GOTO
7643:1DDB CALL 30F0:12528       ..0  205 240  48    find it
7646:1DDE LD   DE,(3ED9:16089) .[.>237  91 217  62 start of PGM table
7650:1DE2 PUSH HL                .    229
7651:1DE3 OR   A                 .    183
7652:1DE4 SBC  HL,DE            .R    237  82
7654:1DE6 POP  HL                .    225
7655:1DE7 JR   Z,1E09:7689        (     40        32
```

```
7657:1DE9 DEC  HL              +     43
7658:1DEA DEC  HL              +     43
7659:1DEB DEC  HL              +     43
7660:1DEC LD   D,(HL)          V     86
7661:1DED DEC  HL              +     43
7662:1DEE LD   E,(HL)          ^     94
7663:1DEF EX   (SP),HL         .    227
7664:1DF0 EX   DE,HL           .    235
7665:1DF1 PUSH HL              .    229
7666:1DF2 OR   A               .    183
7667:1DF3 SBC  HL,DE           .R   237  82
7669:1DF5 POP  HL              .    225
7670:1DF6 EX   DE,HL           .    235
7671:1DF7 EX   (SP),HL         .    227
7672:1DF8 JR   C,1E09:7689      8.     56        15
7674:1DFA CALL 31F2:12786      ..1  205 242  49    delete range
7677:1DFD LD   HL,(3EDB:16091) *.>  42 219  62    number of lines
7680:1E00 LD   A,H             |    124
7681:1E01 OR   L               .    181
7682:1E02 JR   Z,1E09:7689      (.    40         5
7684:1E04 POP  BC              .    193
7685:1E05 POP  HL              .    225
7686:1E06 JP   1DD6:7638       ...  195 214  29
7689:1E09 POP  BC              .    193
7690:1E0A POP  HL              .    225
7691:1E0B POP  DE              .    209
7692:1E0C JP   3EA3:16035      ..>  195 163  62    main loop
7695:1E0F PUSH DE              .    213
7696:1E10 LD   H,B             `     96
7697:1E11 LD   L,C             i    105
7698:1E12 CALL 31EB:12779      ..1  205 235  49    do delete
7701:1E15 POP  DE              .    209
7702:1E16 JP   3EA3:16035      ..>  195 163  62    main loop

IF execution
7705:1E19 CALL 1733:5939       .3.  205  51  23    get equation from token
7708:1E1C OR   A               .    183
7709:1E1D JP   NZ,1E2B:7723      .+.  194        43  30    condition true
7712:1E20 LD   A,(3F26:16166) :&?   58  38  63    FPA1 exponent
7715:1E23 OR   A               .    183
7716:1E24 JP   Z,1E36:7734      .6.  202        54  30    skip line
7719:1E27 DEC  A               =     61
7720:1E28 JP   Z,1E36:7734      .6.  202        54  30    skip line
7723:1E2B EXX                  .    217
7724:1E2C DEC  C               .     13
7725:1E2D EXX                  .    217
7726:1E2E INC  DE              .     19
7727:1E2F LD   A,(DE)          .     26                   get next token
7728:1E30 CP   00B2:178        ..   254 178              THEN token
7730:1E32 RET  Z               .    200                  do it
7731:1E33 JP   2096:8342       ..   195 150  32    do GOTO (assumed)
7734:1E36 EXX                  .    217
7735:1E37 CALL 1780:6016       ...  205 128  23    find next line number
7738:1E3A DEC  DE              .     27
7739:1E3B RET                  .    201

PRINT
7740:1E3C EXX                  .    217
7741:1E3D DEC  C               .     13
7742:1E3E DEC  C               .     13
7743:1E3F EXX                  .    217
7744:1E40 INC  DE              .     19
7745:1E41 INC  DE              .     19
7746:1E42 CALL 05DC:1500           205 220   5    LD HL with token number
7749:1E45 CALL 6641:26177      .Af  205  65 102    find POS
7752:1E48 CP   L               .    189
7753:1E49 JP   NC,1E6B:7787      .k.  210 107  30
7756:1E4C NEG  A               .D   237  68
7758:1E4E ADD  A,L             .    133
7759:1E4F LD   L,A             o    111
7760:1E50 DEC  L               -     45
7761:1E51 JP   NZ,1E63:7779      .c.  194        99  30
7764:1E54 JP   1E6B:7787       .k.  195 107  30
```

```
7767:1E57 EXX                    .    217
7768:1E58 DEC  C                 .     13
7769:1E59 DEC  C                 .     13
7770:1E5A EXX                    .    217
7771:1E5B INC  DE                .     19
7772:1E5C INC  DE                .     19
7773:1E5D CALL 05DC:1500         ...  205 220   5    LD HL with token number
7776:1E60 OR   L                 .    181
7777:1E61 JR   Z,1E6B:7787        (.    40          8
7779:1E63 LD   A,0020:32         >     62  32         space
7781:1E65 CALL 2EDA:11994        ...  205 218  46    print it
7784:1E68 DEC  L                 -     45
7785:1E69 JR   NZ,1E65:7781          .    32 250     clear to end of line
7787:1E6B EXX                    .    217
7788:1E6C DEC  C                 .     13
7789:1E6D EXX                    .    217
7790:1E6E INC  DE                .     19
7791:1E6F LD   A,(DE)            .     26
7792:1E70 CP   00B8:184          ..   254 184          ) token
7794:1E72 JP   Z,1ED4:7892         ...  202 212  30
7797:1E75 JP   1F0C:7948         ...  195  12  31    syntax error
7800:1E78 EXX                    .    217
7801:1E79 DEC  C                 .     13
7802:1E7A EXX                    .    217
7803:1E7B INC  DE                .     19
7804:1E7C LD   A,(DE)            .     26
7805:1E7D CP   00B9:185          ..   254 185          , token
7807:1E7F JP   Z,1EC3:7875         ...  202 195  30
7810:1E82 CP   00BA:186          ..   254 186          ; token
7812:1E84 JP   Z,1ED4:7892         ...  202 212  30
7815:1E87 CP   008C:140          ..   254 140          equation token
7817:1E89 JP   NZ,1E9D:7837        ...  194 157  30
7820:1E8C EXX                    .    217
7821:1E8D DEC  C                 .     13
7822:1E8E EXX                    .    217
7823:1E8F INC  DE                .     19
7824:1E90 LD   A,(DE)            .     26
7825:1E91 OR   A                 .    183
7826:1E92 JP   Z,1E57:7767         .W.  202        87  30
7829:1E95 DEC  A                 =     61
7830:1E96 JP   Z,1E3C:7740         .<.  202        60  30
7833:1E99 EXX                    .    217
7834:1E9A INC  C                 .     12
7835:1E9B EXX                    .    217
7836:1E9C DEC  DE                .     27
7837:1E9D EXX                    .    217
7838:1E9E INC  C                 .     12
7839:1E9F EXX                    .    217
7840:1EA0 DEC  DE                .     27
7841:1EA1 CALL 1733:5939         .3.  205  51  23    get equation
7844:1EA4 JP   NC,1E78:7800        .x.  210 120  30
7847:1EA7 OR   A                 .    183
7848:1EA8 JP   NZ,1EB8:7864        ...  194 184  30
7851:1EAB CALL 0611:1553         ...  205  17   6    print FPA1
7854:1EAE EXX                    .    217
7855:1EAF LD   A,C               y    121
7856:1EB0 EXX                    .    217
7857:1EB1 OR   A                 .    183
7858:1EB2 JP   NZ,1E78:7800        .x.  194 120  30
7861:1EB5 JP   2F60:12128        .'/  195  96  47    print a <CR>
7864:1EB8 LD   HL,(3F22:16162)   *"?  42  34  63    get string pointer (FPA1)
7867:1EBB INC  HL                #     35
7868:1EBC INC  HL                #     35
7869:1EBD CALL 2F4E:12110        .N/  205  78  47    print string
7872:1EC0 JP   1EAE:7854         ...  195 174  30
7875:1EC3 CALL 6641:26177        .Af  205  65 102    find POS
7878:1EC6 AND  000F:15           ..   230   15         move to
7880:1EC8 SUB   0010:16          ..   214  16                1/2 screen
7882:1ECA LD   C,A               O     79
7883:1ECB LD   A,0020:32         >     62  32         space
7885:1ECD CALL 2EDA:11994        ...  205 218  46    print it
7888:1ED0 INC  C                 .     12
7889:1ED1 JP   NZ,1ECD:7885        ...  194 205  30          clear to 1/2 screen
7892:1ED4 EXX                    .    217
7893:1ED5 LD   A,C               y    121
7894:1ED6 EXX                    .    217
7895:1ED7 OR   A                 .    183
7896:1ED8 JP   NZ,1E78:7800        .x.  194 120  30
7899:1EDB RET                    .    201
```

```
ERROR HANDLER -- entry points 1EEB,1EEE,1EF1, etc load A with error code
7900:1EDC LD   HL,1EDC:7900   !..    33 220  30     this address
7903:1EDF PUSH HL             .      229
7904:1EE0 LD   IX,0000:0      .!..  221  33   0   0
7908:1EE4 ADD  IX,SP          .9    221  57
7910:1EE6 PUSH IX             ..    221 229         clear GOTO
7912:1EE8 POP  IY             ..    253 225                 and GOSUB
7914:1EEA LD   BC,113E:4414   .>.     1  62  17     illegal mode
7917:1EED LD   BC,0B3E:2878   .>.     1  62  11     division by 0
7920:1EF0 LD   BC,053E:1342   .>.     1  62   5     overflow
7923:1EF3 LD   BC,0A3E:2622   .>.     1  62  10     redim array
7926:1EF6 LD   BC,063E:1598   .>.     1  62   6     out of memory
7929:1EF9 LD   BC,033E:830    .>.     1  62   3     out of data
7932:1EFC LD   BC,0F3E:3902   .>.     1  62  15     formula too complex
7935:1EFF LD   BC,043E:1086   .>.     1  62   4     illegal quantity
7938:1F02 LD   BC,0C3E:3134   .>.     1  62  12     type mismatch
7941:1F05 LD   BC,103E:4158   .>.     1  62  16     incorrect function
7944:1F08 LD   BC,0D3E:3390   .>.     1  62  13     string too long
7947:1F0B LD   BC,013E:318    .>.     1  62   1     syntax error
7950:1F0E LD   (3F00:16128),A 2.?    50   0  63     save error code
7953:1F11 EXX                 .      217
7954:1F12 LD   (3EFA:16122),DE        237  83 250  62 save PGM
7958:1F16 LD   (3EFC:16124),HL ".>    34 252  62             pointers
7961:1F19 SET  2,B            ..     203 208         set immediate mode
7963:1F1B EXX                 .      217
7964:1F1C CP   0014:20        ..    254  20         if error from table??
7966:1F1E JR   NC,1F27:7975     0.     48        7
7968:1F20 EXX                 .      217
7969:1F21 BIT  0,B            .@     203  64         ONERR flag
7971:1F23 EXX                 .      217
7972:1F24 JP   NZ,1F92:8082     ...   194 146  31          do ONERR
7975:1F27 CALL 5F63:24419     .c    205  99  95     close files
7978:1F2A CALL 2F60:12128     .'7   205  96  47     print a <CR>
7981:1F2D LD   A,003F:63      >?     62  63
7983:1F2F CALL 2EDA:11994     ...   205 218  46     print ?
7986:1F32 LD   A,(3F00:16128) :.?    58   0  63     error code
7989:1F35 CP   0013:19        ..    254  19
7991:1F37 JR   NZ,1F3E:7998     .     32        5
7993:1F39 LD   A,0014:20      >.     62  20
7995:1F3B LD   (3F00:16128),A 2.?    50   0  63     new error code
7998:1F3E LD   B,A            G      71             # of messages to skip
7999:1F3F LD   HL,0480:1152   !..    33 128   4     error table
8002:1F42 OR   A              .     183
8003:1F43 JR   Z,1F4E:8014      (.     40        9
8005:1F45 LD   A,(HL)         .     126             length byte
8006:1F46 INC  HL             #      35
8007:1F47 ADD  A,L            .     133
8008:1F48 LD   L,A            o     111             move to end of message
8009:1F49 JR   NC,1F4C:8012     0.     48        1
8011:1F4B INC  H              $      36
8012:1F4C DJNZ 1F45:8005      ..     16 247         skip required messages
8014:1F4E CALL 2F4E:12110     .N/   205  78  47     print message
8017:1F51 LD   A,(3F00:16128) :.?    58   0  63     error code
8020:1F54 CP   0014:20        ..    254  20
8022:1F56 JR   NC,1F5E:8030     0.     48        6
8024:1F58 LD   HL,1FA1:8097   !..    33 161  31     'ERROR'
8027:1F5B CALL 2F4E:12110     .N/   205  78  47     print it
8030:1F5E EXX                 .      217
8031:1F5F BIT  6,B            .p     203 112         run mode?
8033:1F61 EXX                 .      217
8034:1F62 JR   Z,1F77:8055      (.     40       19
8036:1F64 LD   HL,1FA8:8104   !..    33 168  31     'IN'
8039:1F67 CALL 2F4E:12110     .N/   205  78  47     print it
8042:1F6A EXX                 .      217
8043:1F6B PUSH HL             .      229             line number
8044:1F6C EXX                 .      217
8045:1F6D POP  HL             .      225
8046:1F6E DEC  HL             +       43
8047:1F6F DEC  HL             +       43
8048:1F70 DEC  HL             +       43
8049:1F71 LD   D,(HL)         V       86
8050:1F72 DEC  HL             +       43
8051:1F73 LD   E,(HL)         ^       94
8052:1F74 CALL 32A7:12967     ..2   205 167  50     print (HL)
8055:1F77 CALL 2F60:12128     .'/   205  96  47     print <CR>
```

```
8058:1F7A LD   HL,0000:0      !..    33   0   0
8061:1F7D LD   (3F22:16162),HL ""?  34  34  63     zero FPA1
8064:1F80 LD   (3F2B:16171),HL "+?  34  43  63          FPA2
8067:1F83 LD   A,(3F00:16128) :.?   58   0  63     error code
8070:1F86 CP   0014:20         ..   254  20
8072:1F88 JP   NZ,3EA3:16035     ..>  194 163  62          clear stack
8075:1F8B EXX                  .    217
8076:1F8C RES  2,B             ..   203 144         main loop
8078:1F8E EXX                  .    217
8079:1F8F JP   3EA6:16038      ..>  195 166  62     main loop
8082:1F92 EXX                  .    217
8083:1F93 SET  1,B             ..   203 200         ONERR flag
8085:1F95 EXX                  .    217
8086:1F96 LD   BC,1848:6216    .H.    1  72  24     main loop
8089:1F99 PUSH BC              .    197
8090:1F9A LD   BC,(3EFE:16126)      237  75 254  62 line for ONERR
8094:1F9E JP   209E:8350       ..   195 158  32     GOTO it

ERROR MESSAGES
8097:1FA1        6  32
8099:1FA3 Error 69 114 114 111 114
8104:1FA8 .      4  32
8105:1FA9 In    73 110  32

CLRERR
8109:1FAD EXX                       217
8110:1FAE RES  0,B             ..   203 128         ONERR flag
8112:1FB0 EXX                  .    217
8113:1FB1 RET                  .    201

ONERR
8114:1FB2 EXX                  .    217
8115:1FB3 BIT  6,B             .p   203 112         run mode?
8117:1FB5 JR   Z,1FC9:8137       (.     40        18
8119:1FB7 DEC  C               .     13
8120:1FB8 DEC  C               .     13
8121:1FB9 EXX                  .    217
8122:1FBA INC  DE              .     19
8123:1FBB INC  DE              .     19
8124:1FBC LD   A,(DE)          .     26
8125:1FBD CALL 05EF:1519       ...  205 239   5     LD BC with token line #
8128:1FC0 LD   (3EFE:16126),BC .C.>237  67 254 62 line for ONERR
8132:1FC4 EXX                  .    217
8133:1FC5 SET  0,B             ..   203 192         ONERR flag
8135:1FC7 EXX                  .    217
8136:1FC8 RET                  .    201
8137:1FC9 EXX                  .    217
8138:1FCA JP   1EEB:7915       ...  195 235  30     illegal mode

CLEAR
8141:1FCD CALL 250A:9482       ..%  205  10  37     RESTORE
8144:1FD0 LD   HL,(3ED9:16089) *.>  42 217  62     Start of PGM table
8147:1FD3 DEC  HL                +    43
8148:1FD4 LD   (3EED:16109),HL ".>   34 237  62     start of variables
8151:1FD7 LD   HL,(3EDF:16095) *.>   42 223  62     start of variable table
8154:1FDA INC  HL              #     35
8155:1FDB LD   BC,0091:145     ...    1 145   0     default value?
8158:1FDE ADD  HL,BC           .      9
8159:1FDF LD   BC,(3EEB:16107) .K.>237  75 235 62 number of variables
8163:1FE3 LD   A,B             x    120
8164:1FE4 OR   A               .    183
8165:1FE5 JR   NZ,1FEC:8172        .    32         5
8167:1FE7 LD   A,001D:29       >.    62  29
8169:1FE9 CP   C               .    185
8170:1FEA JR   Z,2002:8194       (.     40        22
8172:1FEC NEG  A               .D   237  68
8174:1FEE ADD  A,C             .    129
8175:1FEF LD   C,A             O     79
8176:1FF0 JR   C,1FF3:8179       8.     56         1
8178:1FF2 DEC  B               .      5
```

```
8179:1FF3 XOR  A                .    175
8180:1FF4 PUSH DE               .    213
8181:1FF5 LD   D,A              W     87
8182:1FF6 LD   E,0003:3         ..    30   3
8184:1FF8 LD   (HL),A           w    119
8185:1FF9 INC  HL               #     35
8186:1FFA LD   (HL),A           w    119
8187:1FFB CPI                   ..   237 161
8189:1FFD ADD  HL,DE            .     25
8190:1FFE JP   PE,1FF8:8184        ...  234 248  31
8193:2001 POP  DE               .    209
8194:2002 DEC  HL               +     43
8195:2003 LD   A,(3EE1:16097) :.>    58 225  62    end of variable table
8198:2006 CP   L                .    189
8199:2007 JR   NZ,200F:8207        .    32        6
8201:2009 LD   A,(3EE2:16098) :.>    58 226  62    end of variable table
8204:200C CP   H                .    188
8205:200D JR   Z,202A:8234       (.    40        27
8207:200F PUSH DE               .    213
8208:2010 LD   D,H              T     84
8209:2011 LD   E,L              ]     93
8210:2012 LD   BC,(3EE1:16097)        237  75 225 62 end of variable table
8214:2016 PUSH BC               .    197
8215:2017 LD   (3EE1:16097),HL ".>    34 225  62
8218:201A LD   HL,(3EE3:16099) *.>    42 227  62    end of variable commands
8221:201D OR   A                .    183
8222:201E SBC  HL,BC            .B   237  66
8224:2020 LD   B,H              D     68
8225:2021 LD   C,L              M     77
8226:2022 POP  HL               .    225
8227:2023 LDIR                  ..   237 176        clear all variables
8229:2025 LD   (3EE3:16099),DE        237  83 227 63 end of variable commands
8233:2029 POP  DE                     209
8234:202A LD   HL,(3EE3:16099) *.>    42 227  62    end of variable commands
8237:202D LD   (3EF3:16115),HL ".>    34 243  62    end of string space
8240:2030 LD   (3EEF:16111),HL ".>    34 239  62    start of string space
8243:2033 RET                   .    201
```