```
36C5  18 09                    JR CH13
36C7  3E 06           CH11:    LD A,6 ;CHECK FOR CNTRL
36C9  B8                       CP B
36CA  28 03                    JR Z,CH12
36CC  F1                       POP AF
36CD  18 07                    JR VALID
36CF  F1              CH12:    POP AF ;RESTORE SENSE
36D0  CB C7           CH13:    SET 0,A ;RESET ALPHA,SHIFT AND CONTROL
36D2  FE FF           CH14:    CP 0FFH ;TRY AGAIN AFTER ELIMINATING ODD KEYS
36D4  28 09                    JR Z,SCAN3 ;IF STILL 0 THEN TRY OTHER DRIVE
                               ;VALID KEY FOUND
36D6  D1              VALID:   POP DE ;REMOVE DRIVE
36D7  0E 00                    LD C,0
36D9  0F              CH15:    RRCA
36DA  30 17                    JR NC,ENDSCAN ;B=1-8,C=0-7
36DC  0C                       INC C
36DD  18 FA                    JR CH15
36DF  DB 06           SCAN3:   IN A,(SENSE2) ;SECOND DRIVE
36E1  E6 03                    AND 03
36E3  FE 03                    CP 3
36E5  28 07                    JR Z,SCAN4
36E7  A7              CHECK2:  AND A
36E8  C6 07                    ADD A,7
36EA  4F                       LD C,A ;C=8 OR 9
36EB  F1                       POP AF

36EC  18 05           JR ENDSCAN
36EE  F1              SCAN4:   POP AF ;RESTORE DRIVE
36EF  10 BE                    DJNZ SCAN2 ;MOVE TO NEXT DRIVE
36F1  0E 00                    LD C,0
36F3  79              ENDSCAN:       LD A,C
36F4  87                       ADD A,A
36F5  87                       ADD A,A
36F6  87                       ADD A,A
36F7  80                       ADD A,B
36F8  21 FD7B                  LD HL,AUTOCT
36FB  20 04                    JR NZ,DEB1
36FD  36 B0                    LD (HL),0B0H
36FF  28 08                    JR Z,DEB2
3701  CB BE           DEB1:    RES 7,(HL)
3703  47                       LD B,A
3704  3A FD7C                  LD A,(LASTKY)
3707  B8                       CP B
3708  78                       LD A,B
3709  32 FD7C         DEB2:    LD (LASTKY),A
370C  C8                       RET Z
370D  F5                       PUSH AF
370E  AF                       XOR A
370F  BE                       CP (HL)
3710  20 04                    JR NZ,ENDAUTO
3712  CB D6           FAST:    SET 2,(HL)
3714  CB CE                    SET 1,(HL)
3716  F1              ENDAUTO:       POP AF
3717  C9                       RET
                               ;
3718  26 00           LOOK:    LD H,0 ;LOOK UP IN TABLE SPECIFIED BY BC
```

```
371A  6F                    LD L,A
371B  09                    ADD HL,BC
371C  7E                    LD A,(HL)
371D  32 FD7D               LD (LASTASC),A
3720  C9                    RET
                      ;
3721  00              BASE:  DB 00
3722  7A 00 61 A0           DB 122,0,97,160,113,0,27,49
3726  71 00 1B 31
372A  63 7B 64 73           DB 99,120,100,115,101,119,50,51
372E  65 77 32 33
3732  62 76 67 66           DB 98,118,103,102,116,114,52,53
3736  74 72 34 35
373A  6D 6E 6A 68           DB 109,110,106,104,117,121,54,55
373E  75 79 36 37
3742  2E 2C 6C 6B           DB 46,44,108,107,111,105,56,57
3746  6F 69 38 39
374A  5F 2F 3A 3B           DB 95,47,58,59,64,112,48,45
374E  40 70 30 2D
3752  15 00 0D 5D           DB 21,0,13,93,10,91,94,92
3756  0A 5B 5E 5C
375A  0C 0A 1A 19           DB 12,10,26,25,8,11,5,29
375E  08 0B 05 1D
                      ;
                      ;DB 134,135,132,133,131,130,129,128     ;f4,f8,f3,f7,f6,f2,f5,f1
3762  83 87 82 86           DB 131,135,130,134,133,129,132,128
3766  85 81 84 80
                      ;
376A  20 00 00 00           DB 32,0,0,0,127,9,8,3
376E  7F 09 08 03
3772  00              UPPER:  DB 0
3773  5A 00 41 A0           DB 90,0,65,160,81,0,27,33
3777  51 00 1B 21
377B  43 58 44 53           DB 67,88,68,83,69,87,34,35
377F  45 57 22 23
3783  42 56 47 46           DB 66,86,71,70,84,82,36,37
3787  54 52 24 25
378B  4D 4E 4A 48           DB 77,78,74,72,85,89,38,39
378F  55 59 26 27
3793  3E 3C 4C 4B           DB 62,60,76,75,79,73,40,41
3797  4F 49 28 29
379B  5F 3F 2A 2B           DB 95,63,42,43,96,80,48,61
379F  60 50 30 3D
37A3  30 00 0D 7D           DB 48,0,13,125,10,123,126,124
37A7  0A 7B 7E 7C
37AB  0D 2E 32 33           DB 13,46,50,51,49,53,56,55
37AF  31 35 38 37
                      ;
                      ;DB 142,143,140,141,139,138,137,136
37B3  8B 8F 8A 8E           DB 139,143,138,142,141,137,140,136      ;f4,f8,f3,f7,f6,f2,f5,f1
37B7  8D 89 8C 88
                      ;
37BB  20 00 00 00           DB 32,0,0,0,54,52,8,57
37BF  36 34 08 39
37C3  E5              ASCMAP: PUSH HL
37C4  C5                      PUSH BC
```

```
37C5  F5                    PUSH AF
37C6  47                    LD B,A
37C7  3A FB45               LD A,(CTYSLT)
37CA  21 37E6               LD HL,CONTRY
37CD  CD 0689               CALL FINDJP
37D0  7E              ASCTES: LD A,(HL)
37D1  FE FF                 CP 0FFH
37D3  20 04                 JR NZ,ASCCPR
37D5  F1                    POP AF
37D6  C1                    POP BC
37D7  E1                    POP HL
37D8  C9                    RET
37D9  90              ASCCPR: SUB B
37DA  28 04                 JR Z,ASCMAT
37DC  23                    INC HL
37DD  23                    INC HL
37DE  18 F0                 JR ASCTES
37E0  23              ASCMAT: INC HL
37E1  F1                    POP AF
37E2  C1                    POP BC
37E3  7E                    LD A,(HL)
37E4  E1                    POP HL
37E5  C9                    RET
37E6  37F0            CONTRY: DW ENGLAN
37E8  37F1                  DW FRANCE
37EA  382C                  DW GERMAN
37EC  3863                  DW SWEDIS
                      ;     DW DENMAR
37EE  3890                  DW SPAIN
37F0                  ENGLAND::
37F0  FF                    DB 0FFH
37F1                  FRANCE::
37F1  51 41                 DB 51H,41H
37F3  57 5A                 DB 57H,5AH
37F5  41 51                 DB 41H,51H
37F7  5A 57                 DB 5AH,57H
37F9  71 61                 DB 71H,61H
37FB  77 7A                 DB 77H,7AH
37FD  61 71                 DB 61H,71H
37FF  7A 77                 DB 7AH,77H
3801  7C 3E                 DB 7CH,3EH
3803  5C 3C                 DB 5CH,3CH
3805  5F 3D                 DB 5FH,3DH
3807  3D 3F                 DB 3DH,3FH
3809  2D 2F                 DB 2DH,2FH
380B  7E 5E                 DB 7EH,5EH
380D  5E 5B                 DB 5EH,5BH
380F  60 5D                 DB 60H,5DH
3811  40 7D                 DB 40H,7DH
3813  7B 2A                 DB 7BH,2AH
3815  5B 2B                 DB 5BH,2BH
3817  2B 5C                 DB 2BH,5CH
3819  3B 7B                 DB 3BH,7BH
381B  2A 7C                 DB 2AH,7CH
381D  3A 40                 DB 3AH,40H
381F  7D 7E                 DB 7DH,7EH
```

```
3821  5D 60                    DB 5DH,60H
3823  3C 3B                    DB 3CH,3BH
3825  3E 3A                    DB 3EH,3AH
3827  3F 5F                    DB 3FH,5FH
3829  2F 2D                    DB 2FH,2DH
382B  FF                       DB 0FFH
382C                     GERMAN::
382C  59 5A                    DB 59H,5AH
382E  5A 59                    DB 5AH,59H
3830  7A 79                    DB 7AH,79H
3832  79 7A                    DB 79H,7AH
3834  7C 3E                    DB 7CH,3EH
3836  5C 3C                    DB 5CH,3CH
3838  23 40                    DB 23H,40H
383A  60 2D                    DB 60H,2DH
383C  5F 3D                    DB 5FH,3DH
383E  3D 3F                    DB 3DH,3FH
3840  2D 7E                    DB 2DH,7EH
3842  7E 27                    DB 7EH,27H
3844  5E 7E                    DB 5EH,7EH
3846  40 7D                    DB 40H,7DH
3848  27 2F                    DB 27H,2FH
384A  7B 2A                    DB 7BH,2AH
384C  5B 2B                    DB 5BH,2BH
384E  2B 5C                    DB 2BH,5CH
3850  3B 7C                    DB 3BH,7CH
3852  2A 5B                    DB 2AH,5BH
3854  3A 7B                    DB 3AH,7BH
3856  7D 7C                    DB 7DH,7CH
3858  5D 23                    DB 5DH,23H
385A  3C 3B                    DB 3CH,3BH
385C  3E 3A                    DB 3EH,3AH
385E  3F 5F                    DB 3FH,5FH
3860  2F 2D                    DB 2FH,2DH
3862  FF                       DB 0FFH
3863                     SWEDIS::
3863  7C 3E                    DB 7CH,3EH
3865  5C 3C                    DB 5CH,3CH
3867  60 2F                    DB 60H,2FH
3869  5F 3D                    DB 5FH,3DH
386B  3D 3F                    DB 3DH,3FH
386D  2D 2B                    DB 2DH,2BH
386F  7E 40                    DB 7EH,40H
3871  5E 60                    DB 5EH,60H
3873  27 2F                    DB 27H,2FH
3875  40 7D                    DB 40H,7DH
3877  7B 5E                    DB 7BH,5EH
3879  5B 7E                    DB 5BH,7EH
387B  2B 5C                    DB 2BH,5CH
387D  3B 7C                    DB 3BH,7CH
387F  2A 5B                    DB 2AH,5BH
3881  3A 7B                    DB 3AH,7BH
3883  7D 2A                    DB 7DH,2AH
3885  5D 27                    DB 5DH,27H
3887  3C 3B                    DB 3CH,3BH
3889  3E 3A                    DB 3EH,3AH
```

```
388B  3F 5F                   DB 3FH,5FH
388D  2F 2D                   DB 2FH,2DH
388F  FF                      DB 0FFH
                      ;DENMAR:      DB 7CH,3FH ;¦,?
                      ;       DB 5CH,2FH ;\,/
                      ;       DB 27H,5DH ;',A
                      ;       DB 40H,7DH ;@,a
                      ;       DB 2BH,5BH ;+,AE
                      ;       DB 3BH,7BH ;;,ae
                      ;       DB 2AH,5CH ;*,0
                      ;       DB 3AH,7CH ;:,0
                      ;       DB 7BH,2BH ;{,+
                      ;       DB 5BH,3BH ;[,;
                      ;       DB 7DH,2AH ;),*
                      ;       DB 5DH,3AH ;],:
                      ;       DB 3FH,3DH ;?,=
                      ;       DB 2FH,2DH ;/,-
                      ;       DB 7EH,60H ;~,'
                      ;       DB 5EH,40H ;^,@
                      ;       DB 2DH,5EH ;=,U or ^
                      ;       DB 3DH,7EH ;-,u OR ~
                      ;       DB 0FFH
3890                  SPAIN::
3890  7C 3E                   DB 7CH,3EH ;¦,>
3892  5C 3C                   DB 5CH,3CH ;\,<
3894  5F 3D                   DB 5FH,3DH ;_,=
3896  3D 3F                   DB 3DH,3FH ;=,?
3898  2D 2B                   DB 2DH,2BH ;-,+
389A  7E 60                   DB 7EH,60H ;~,'
389C  5E 40                   DB 5EH,40H ;^,@
389E  60 2A                   DB 60H,2AH ;',*
38A0  40 2F                   DB 40H,2FH ;@,/
38A2  7B 7E                   DB 7BH,7EH ;{,~
38A4  5B 5E                   DB 5BH,5EH ;[,^
38A6  2B 5C                   DB 2BH,5CH ;+,N
38A8  3B 7C                   DB 3BH,7CH ;;,n
38AA  2A 5D                   DB 2AH,5DH ;*,?U
38AC  3A 7B                   DB 3AH,7BH ;:,c
38AE  7D 5B                   DB 7DH,5BH ;),!U
38B0  5D 7D                   DB 5DH,7DH ;],COC
38B2  3C 3B                   DB 3CH,3BH ;<,;
38B4  3E 3A                   DB 3EH,3AH ;>,:
38B6  3F 5F                   DB 3FH,5FH ;?,_
38B8  2F 2D                   DB 2FH,2DH ;/,-
38BA  FF                      DB 0FFH
                      ;
38BB  DF              LEX:    RST OSPRST      RST 18  Find next non-space characte
38BC  C8                      RET Z           Exit if ENDMARK
38BD  FE 22                   CP 34
38BF  20 0C                   JR NZ,TL2
38C1  23              TL1:    INC HL ;FIND MATCHING QUOTE
38C2  7E                      LD A,(HL)
38C3  FE FF                   CP ENDMARK
38C5  C8                      RET Z
38C6  FE 22           TL3:    CP 34
38C8  20 F7                   JR NZ,TL1
```

Skip double quoted strings

```
38CA  23                  INC HL
38CB  18 EE               JR LEX
38CD  EF            TL2:  RST JT
38CE  BD                  DB 0BDH
                    ;CALL ALPHA
38CF  30 2A               JR NC,NEXCH
38D1  CD 3980             CALL CAPITAL
38D4  01 253B             LD BC,WORDLST
38D7  16 80               LD D,128
38D9  CD 3935             CALL LEX1
38DC  28 08               JR Z,TL4
38DE  FE 80               CP REM
38E0  C8                  RET Z
38E1  FE 8F               CP DATA
38E3  C8                  RET Z
38E4  18 D5               JR LEX
38E6  01 2680       TL4:  LD BC,LEXFUN
38E9  16 DA               LD D,218
38EB  CD 3935             CALL LEX1
38EE  20 C8               JR NZ,LEX
                    ;
38F0  EF            OVERALN:      RST JT
38F1  BD                  DB 0BDH
                    ;CALL ALPHA ;MOVE OVER LETTERS AND DIGITS
38F2  38 04               JR C,NF1 ;IN STRING POINTED TO BY HL
38F4  EF                  RST JT
38F5  B7                  DB 0B7H
                    ;CALL DIGIT
38F6  30 C3               JR NC,LEX
38F8  23            NF1:  INC HL
38F9  18 F5               JR OVERALN
                    ;
38FB  E5            NEXCH:  PUSH HL
38FC  5F                  LD E,A ;LOOK FOR DOUBLES
38FD  0E D7               LD C,215
38FF  23                  INC HL
3900  DF                  RST OSPRST
3901  E5                  PUSH HL ;POSITION OF SECOND OPERATOR
3902  56                  LD D,(HL) ;E='>'
3903  CB FA               SET 7,D
3905  21 267A             LD HL,OPS+8 ;STARTOF DOUBLES <=,>=,<>
3908  06 03               LD B,3
                    ;
390A  CD 1BB6       NEXCH1: CALL TESTHLDE
390D  28 05               JR Z,NEXFOU
390F  0C                  INC C
3910  10 FB               DJNZ NEXCH1
3912  18 08               JR NEXCH5
3914  E1            NEXFOU: POP HL
3915  36 20               LD (HL),32
3917  E1            NEXCH4: POP HL
3918  71                  LD (HL),C
3919  23                  INC HL
391A  18 9F               JR LEX
                    ;
391C  E1            NEXCH5: POP HL
```

```
391D  E1                    POP HL
391E  7E                    LD A,(HL)
391F  0E CF                 LD C,207 ;START OF OPS
3921  06 08                 LD B,8 ;8 OPS
3923  E5                    PUSH HL ;SAVE CURRENT POS
3924  21 2672               LD HL,OPS ;START OF OPS TABLE
3927  CB FF                 SET 7,A
3929  BE              NEXCHO: CP (HL) ;FIND OPERATOR
392A  28 EB                 JR Z,NEXCH4 ;OP FOUND
392C  0C                    INC C
392D  23                    INC HL
392E  10 F9                 DJNZ NEXCHO ;SINGE OP NOT FOUND
3930  E1                    POP HL
3931  23                    INC HL
3932  C3 38BB               JP LEX
                    ;
                    ;LEX1 MATCHES A STRING WITH THE WORDLIST
                    ;
3935  E5            LEX1:  PUSH HL ;save basic name
3936  3E 2E          LEX2:  LD A,46 ;. implies end of word
3938  BE                    CP (HL)
3939  28 31                 JR Z,FOUND0
393B  0A                    LD A,(BC) ;letter of wordlist
393C  A7                    AND A   ;test if end of list
393D  20 02                 JR NZ,LEX3
393F  C1                    POP BC
3940  C9                    RET     ;if yes then exit
3941  BE            LEX3:  CP (HL) ;cp name with wordlist
3942  20 04                 JR NZ,NOCH ;if not same then try bit 7
3944  23                    INC HL ;otherwise continue with next bit
3945  03                    INC BC
3946  18 EE                 JR LEX2
                    ;
394B  C6 80          NOCH:  ADD A,128 ;if bit 7 then success
394A  BE                    CP (HL)
394B  28 0F                 JR Z,FFFOUND ;exit provided next letter is not alpha
394D  C6 80          LEX7:  ADD A,128 ;otherwise move up wordlist to next
394F  CB 7F          LEX8:  BIT 7,A
3951  20 04                 JR NZ,LEX9 ;FIND START OF NEXT RES WORD
3953  03                    INC BC
3954  0A                    LD A,(BC)
3955  18 F8                 JR LEX8
3957  03            LEX9:  INC BC
3958  14                    INC D   ;INC RW NUMBER
3959  E1                    POP HL ;RESTORE START OF WORD
395A  18 D9                 JR LEX1
                    ;
395C  23            FFFOUND:       INC HL
                          LD E,A ;must test here for no alpha to follow
395E  EF                  RST JT
395F  BD                  DB 0BDH
                    ;CALL ALPHA
3960  2B                  DEC HL
3961  38 06                JR C,FOUNDA
3963  23                  INC HL
3964  EF                  RST JT
```

```
3965  B7                    DB 0B7H
                          ;CALL DIGIT
3966  2B                    DEC HL
3967  30 03                 JR NC,FOUNDO
3969  7B                  FOUNDA: LD A,E
396A  18 E1                 JR LEX7
                          ;
396C  72                  FOUNDO: LD (HL),D
396D  D1                    POP DE
396E  7C                  FOUND1: LD A,H ;INSERT RS CODE AND FILL WITH SPACES
396F  BA                    CP D
3970  20 04                 JR NZ,FOUND2
3972  7D                    LD A,L
3973  BB                    CP E
3974  28 06                 JR Z,FOUNEND
3976  3E 20                FOUND2: LD A,32
3978  12                    LD (DE),A
3979  13                    INC DE
397A  18 F2                 JR FOUND1
397C  7E                  FOUNEND:      LD A,(HL)
397D  23                    INC HL
397E  A7                    AND A
397F  C9                    RET
                          ;
                          ;
3980  E5                  CAPITAL:      PUSH HL
3981  EF                  CAP3:   RST JT
3982  BD                    DB 0BDH
                          ;CALL ALPHA ;IF ALPH THEN RESET
3983  30 05                 JR NC,ENDCAP ;BIT 5
3985  CB AE                 RES 5,(HL)
3987  23                    INC HL
3988  18 F7                 JR CAP3
398A  E1                  ENDCAP: POP HL
398B  C9                    RET
                          ;
398C  E5                  SYN:    PUSH HL
398D  CD 38BB               CALL LEX        ;SEARCH FOR RESERVED WORDS
3990  E1                    POP HL
3991  E5                    PUSH HL
3992  21 02FE               LD HL,INSERT
3995  22 FD82               LD (COMMAND),HL
3998  E1                    POP HL
3999  E5                    PUSH HL
399A  CD 39D1               CALL SYN1
399D  E1                    POP HL
399E  E5                  ONESP:  PUSH HL
399F  7E                  ONESPA: LD A,(HL)
39A0  FE FF                 CP ENDMARK
39A2  28 1D                 JR Z,OUTSP
39A4  FE 22                 CP 34
39A6  20 08                 JR NZ,ONESP1
39A8  23                  ONESPO: INC HL
39A9  7E                    LD A,(HL)
39AA  FE 22                 CP 34
39AC  20 FA                 JR NZ,ONESPO
```

```
39AE  18 0E                 JR ONESP3
39B0  FE 80           ONESP1: CP REM
39B2  28 0D                 JR Z,OUTSP
39B4  FE 8F                 CP DATA
39B6  28 09                 JR Z,OUTSP
39B8  FE 20                 CP 32
39BA  20 02                 JR NZ,ONESP3
39BC  36 00           ONESP2: LD (HL),0
39BE  23              ONESP3: INC HL
39BF  18 DE                 JR ONESPA
39C1  E1              OUTSP:  POP HL
39C2  54                    LD D,H
39C3  5D                    LD E,L
39C4  7E              TTTOP:  LD A,(HL)       ;get char
39C5  12                    LD (DE),A
39C6  FE FF                 CP ENDMARK      ;end of input then exit
39C8  C8                    RET Z
39C9  FE 00           OUT2:   CP 0      ;look for and remove nulls
39CB  28 01                 JR Z,NEXT
39CD  13                    INC DE
39CE  23              NEXT:   INC HL
39CF  18 F3                 JR TTTOP

                      ;
39D1  DF              SYN1:   RST OSPRST
39D2  EF                    RST JT
39D3  B7                    DB 0B7H
                      ;CALL DIGIT     ;TEST IF DIRECT OR PROG
39D4  F5                    PUSH AF  ;SAVE CARRY
39D5  30 07                 JR NC,DIRECT     ;IF PROG THEN TEST LINE NO.
39D7  CD 3D7D               CALL INT64
39DA  28 02                 JR Z,DIRECT      ;LINE NUMBER OUT OF RANGE
39DC  EF                    RST ERROR
39DD  12                    DB 18
39DE  DF              DIRECT: RST OSPRST
39DF  20 0D                 JR NZ,DIR1 ;CARRY IS STILL SAVED IN AF
39E1  21 06C7         DEL:    LD HL,DELETE
39E4  22 FD82         DEL1:   LD (COMMAND),HL
39E7  F1              DEL2:   POP AF
39E8  AF                      XOR A
39E9  C9                      RET
39EA  23              IFTHEN: INC HL
39EB  DF                      RST OSPRST
39EC  37                      SCF
39ED  F5                      PUSH AF

                      ;
39EE  FE 80           DIR1:   CP 128 ;<128 IMPLIES CHARACTER
39F0  38 04                   JR C,DIR3       ;C IMPLIES NOT RESERVED
39F2  FE C2                   CP WDTOP ;200 IMPLIES CANNOT START A LINE
39F4  38 02           DIR2:   JR C,FINDAD      ;GO TO APPROPRIATE ROUTINE
39F6  EF              DIR3:   RST ERROR
39F7  01                      DB 1
39F8  E5              FINDAD: PUSH HL
39F9  21 26F7                 LD HL,RWTAB ;START OF TABLE
39FC  CD 0689                 CALL FINDJP
39FF  E5                      PUSH HL
```

```
3A00  DD E1                   POP IX
3A02  E1                      POP HL
3A03  23                      INC HL
                        ;
3A04  F1              SYNT1:  POP AF
3A05  38 04                   JR C,SYNT
3A07  DD 22 FD82              LD (COMMAND),IX
3A0B  DF              SYNT:   RST OSPRST
3A0C  DD 2B                   DEC IX
3A0E  DD 7E 00                LD A,(IX+0) ;START OF RW HANDLER
3A11  47                      LD B,A  ;SAVE BECAUSE MAY BE LITERAL
3A12  FE C9                   CP 0C9H
3A14  20 10                   JR NZ,CLASS0
3A16  DF              ENDLLLIN:       RST OSPRST
3A17  20 01                   JR NZ,EL1
3A19  C9                      RET     ;Z IMPLIES NO ERROR
3A1A  FE 3A           EL1:    CP ':'
3A1C  20 06                   JR NZ,NOTCOL
3A1E  37                      SCF
3A1F  F5                      PUSH AF
3A20  23                      INC HL
3A21  DF                      RST OSPRST
3A22  18 CA                   JR DIR1
                        ;
3A24  EF              NOTCOL: RST ERROR
3A25  01                      DB 1
3A26  A7              CLASS0: AND A
3A27  20 0A                   JR NZ,CLASS1
3A29  CD 3BF5                 CALL IDENT0
3A2C  3E 01           CL01:   LD A,1
3A2E  B9                      CP C
3A2F  28 DA                   JR Z,SYNT
3A31  EF              CL0ERR: RST ERROR
3A32  0B                      DB 11
3A33  3D              CLASS1: DEC A
3A34  20 06                   JR NZ,CLASS2
3A36  EF                      RST JT
3A37  BE                      DB 0BEH
                        ;CALL SE
3A38  28 D1                   JR Z,SYNT
3A3A  EF                      RST ERROR
3A3B  07                      DB 7
3A3C  3D              CLASS2: DEC A   ;ARITHMETIC EXPRESSION
3A3D  20 06                   JR NZ,CLASS3 ;R-PART
3A3F  EF              CL2ENT: RST JT
3A40  B6                      DB 0B6H
                        ;CALL AE
3A41  28 C8                   JR Z,SYNT
3A43  EF                      RST ERROR
3A44  06                      DB 6
3A45  3D              CLASS3: DEC A
3A46  20 1E                   JR NZ,CLASS4
3A4B  DF              CLAS33: RST OSPRST
3A49  FE 2C                   CP ','
3A4B  28 15                   JR Z,ENDC3
3A4D  FE 3B                   CP ';'
```

If direct command skip
address in (COMMAND)
Get character

Get Syntax byte

C9 = End of syntax

Exit if ENDMARK

If : then scan
syntax for real
command

Syntax error

Jump if syntax byte ≠0
Identify variable

Fail if not
simple numeric
variable.

Jump if syntax byte ≠ 1

Fail if not valid
character expression

Jump if syntax byte ≠2

Fail if not valid
arithmetic
Expression

Jump if syntax byte ≠ 3

Jump if , or ;

```
3A4F  28 11                   JR Z,ENDC3
3A51  EF                      RST JT
3A52  B6                      DB 0B6H
                        ;CALL AE
3A53  C4 3C91                 CALL NZ,SE
3A56  20 B3                   JR NZ,SYNT
3A58  FE 2C             CLAS34: CP ','
3A5A  28 06                   JR Z,ENDC3
3A5C  FE 3B                   CP ';'
3A5E  28 02                   JR Z,ENDC3
3A60  18 A9             SYNTA:  JR SYNT
3A62  23                ENDC3:  INC HL
3A63  7E                      LD A,(HL)
3A64  18 E2                   JR CLAS33
                        ;
                        ;
3A66  3D                CLASS4: DEC A
3A67  20 18                   JR NZ,CLASS5
3A69  EF                      RST JT
3A6A  B7                      DB 0B7H
                        ;CALL DIGIT
3A6B  30 9E                   JR NC,SYNT
3A6D  CD 3D7D           CLAS42: CALL INT64
3A70  28 02                   JR Z,CLAS41
3A72  EF                      RST ERROR
3A73  12                      DB 18
                        ;NUMBER OUT OF RANGE
3A74  DF                CLAS41: RST OSPRST
3A75  FE 2C                   CP ','
3A77  20 92                   JR NZ,SYNT
3A79  23                      INC HL
3A7A  DF                      RST OSPRST
3A7B  EF                      RST JT
3A7C  B7                      DB 0B7H
                        ;CALL DIGIT
3A7D  38 EE                   JR C,CLAS42
3A7F  EF                      RST ERROR
3A80  06                      DB 6
                        ;
3A81  3D                CLASS5: DEC A
3A82  20 05                   JR NZ,CLASS6 ;NUMBER LIST
3A84  CD 3B5C                 CALL NUMLIST
3A87  18 D7             SYNTB:  JR SYNTA
                        ;
3A89  3D                CLASS6: DEC A    ;INT64K FOR LINE NUMBERS
3A8A  20 08                   JR NZ,CLASS7
3A8C  DF                      RST OSPRST
3A8D  CD 3D7D                 CALL INT64
3A90  28 CE                   JR Z,SYNTA
3A92  EF                      RST ERROR
3A93  12                      DB 18
                        ;
3A94  3D                CLASS7: DEC A
3A95  20 01                   JR NZ,CLASS8 ;ANYTHING: DATA, NODDY ETC
3A97  C9                      RET
3A98  3D                CLASS8: DEC A
```

Terminate if not valid Numeric or Character expression

Terminate if not , or ;

Advance one character and repeat.

Jump if result byte ≠ 0

Terminate if not Number. Fail if number out of range

Terminate if not comma

Advance character and test if not numeric. Repeat error.

Jump if syntax byte ±5

Test for list of numeric expressions

Jump if syntax byte ≠6

Test for a single 64k integer constant

Jump if syntax byte ≠6

No tests. Abandon syntax checking.

```
                                        ; Jump if syntax byte ≠ 8
3A99  20 1C                             JR NZ,CLAS9
3A9B  DF                                RST OSPRST
3A9C  FE 22             CLA881:         CP '"'          ; Jump if not quote
3A9E  20 09                             JR NZ,CLAS81
3AA0  CD 3CA7                           CALL SPRLIT     ; Skip string literal
3AA3  DF                                RST OSPRST
3AA4  FE 3B                             CP ';'          ; Fail if not ;
3AA6  23                                INC HL          ; separator
3AA7  20 0C                             JR NZ,CL8ERROR
3AA9  CD 3B4E           CLAS81:         CALL IDLIST     ; Check for list of variables
3AAC  20 07                             JR NZ,CL8ERROR  ; and fail if not
3AAE  FE 3B                             CP ';'          ; Skip ; then
3AB0  20 D5                             JR NZ,SYNTB     ; exit.
3AB2  23                                INC HL
3AB3  18 D2                             JR SYNTB
                                ;
3AB5  EF                CL8ERROR:       RST ERROR
3AB6  01                                DB 1
                                ;
3AB7  3D                CLAS9:  DEC A           ; Jump if syntax byte ≠ 9
3AB8  20 1F                             JR NZ,CLAS10
3ABA  CD 3D53                           CALL BOOLEAN    ; Fail if not valid boolean expression
3ABD  28 02                             JR Z,CLAS90
3ABF  EF                                RST ERROR
3AC0  08                                DB 8
3AC1  DF                CLAS90: RST OSPRST      ; Fail if not THEN token
3AC2  FE C7                             CP THEN
3AC4  28 02                             JR Z,C91
3AC6  EF                                RST ERROR
3AC7  05                                DB 5
3AC8  CD 39EA           C91:    CALL IFTHEN     ; - Check syntax of THEN clause
3ACB  FE C3                             CP ELSE         ; Test for ELSE token
3ACD  20 05                             JR NZ,C92
3ACF  CD 39EA                           CALL IFTHEN     ; - check syntax of ELSE clause
3AD2  FE C3                             CP ELSE         ; Fail if second ELSE
                                ;*****?????             ; otherwise jump to exit
3AD4  C2 3A16           C92:    JP NZ,ENDLLLIN          ; routine
3AD7  EF                                RST ERROR
3AD8  01                                DB 1
3AD9  3D                CLAS10: DEC A           ; Jump if syntax
3ADA  20 09                             JR NZ,CLAS11    ; byte # 0A
3ADC  DF                                RST OSPRST ;SPECIAL FOR STEP AE ; - Skip space
3ADD  FE C6                             CP STEP         ; Exit if not STEP token
3ADF  20 F3                             JR NZ,C92
3AE1  23                                INC HL          ; Test for valid arithmetic
3AE2  C3 3A3F                           JP CL2ENT       ; expression
3AE5  3D                CLAS11: DEC A           ; Jump if syntax byte ≠ 0B
3AE6  20 0C                             JR NZ,CLAS12
3AE8  DF                                RST OSPRST      ; Skip space
3AE9  23                                INC HL
3AEA  FE 96                             CP GOTO         ; Fail if not GOTO or GOSUB.
3AEC  28 99                             JR Z,SYNTB      ; otherwise continue
3AEE  FE 97                             CP GOSUB
3AF0  28 95             SYNTC:  JR Z,SYNTB
3AF2  EF                                RST ERROR
3AF3  05                                DB 5
```

```
3AF4  3D          CLAS12: DEC A          } Jump if syntax byte ≠ 0C
3AF5  20 1C               JR NZ,CLAS13
3AF7  CD 3BF5             CALL IDENTO    - Identify variable
3AFA  C5                  PUSH BC
3AFB  DF                  RST OSPRST     - Skip spaces
3AFC  FE D4               CP 212
3AFE  28 02               JR Z,C121      } Error if not '='
3B00  EF                  RST ERROR
3B01  05                  DB 5
3B02  23          C121:   INC HL         } Skip spaces
3B03  DF                  RST OSPRST
3B04  C1                  POP BC         } Jump if string
3B05  CB 41               BIT 0,C        } variable
3B07  28 04               JR Z,C12SE
3B09  EF                  RST JT         } Test for arithmetic
3B0A  B6                  DB 0B6H        } Expression
                ;CALL AE
3B0B  18 02               JR C12END
3B0D  EF          C12SE:  RST JT         } Test for string expression
3B0E  BE                  DB 0BEH
                ;CALL SE
3B0F  28 DF       C12END: JR Z,SYNTC     } Fail if not valid
3B11  EF                  RST ERROR      } expression of correct type
3B12  0B                  DB 11
3B13  3D          CLAS13: DEC A          } Jump if syntax byte # 0D
3B14  20 0D               JR NZ,CLAS14
3B16  CD 3BF5             CALL IDENTO    - Identify variable
3B19  C2 3A0B             JP NZ,SYNT     - Jump if fails
3B1C  3E 01               LD A,1         } Jump if simple
3B1E  B9                  CP C           } variable
3B1F  28 CF               JR Z,SYNTC
3B21  EF                  RST ERROR      } Fail otherwise
3B22  07                  DB 7
3B23  3D          CLAS14: DEC A          } Jump if syntax byte # 0E
3B24  20 0D               JR NZ,CLAS15
3B26  CD 3B5C             CALL NUMLIST   - Test for list of arithmetic expressions
3B29  DD 2B               DEC IX         } Obtain next syntax byte
3B2B  DD 7E 00            LD A,(IX+0)
3B2E  B9                  CP C           } Fail if number of
3B2F  28 BF               JR Z,SYNTC     } expressions ≠ syntax byte
3B31  EF                  RST ERROR
3B32  00                  DB      0      }
3B33  3D          CLAS15: DEC A          } Jump if syntax byte # 0F
3B34  20 07               JR NZ,CLAS16
3B36  CD 3B4E             CALL IDLIST    } Test for list of variable
3B39  28 B5               JR Z,SYNTC     } names and fail if not
3B3B  EF                  RST ERROR      }
3B3C  01                  DB 1
3B3D  3D          CLAS16: DEC A          } Jump if syntax byte # 10
3B3E  20 07               JR NZ,CLAS17
3B40  CD 3BF5             CALL IDENTO    } Confirm if numeric variable
3B43  CB 41               BIT 0,C        }
3B45  28 A9               JR Z,SYNTC
3B47  DF          CLAS17: RST OSPRST     - Skip spaces
3B48  B8                  CP B
3B49  23                  INC HL
```

```
3B4A  28 A4              JR Z,SYNTC
3B4C  EF                 RST ERROR
3B4D  05                 DB 5
3B4E  E5           IDLIST: PUSH HL
3B4F  CD 3BF5      ID1:  CALL IDENTO
3B52  20 4E              JR NZ,SYFL
3B54  DF                 RST OSPRST
3B55  FE 2C              CP ','
3B57  20 17              JR NZ,NUML2
3B59  23                 INC HL
3B5A  18 F3              JR ID1
                         ;
                         ;
                         ;NUMLIST FINDS A LIST OF AES SEPARATED BY ,S AND RETURNS THE
                         ;NUMBER FOUND IN C
3B5C  E5           NUMLIST:      PUSH HL
3B5D  EF                 RST JT
3B5E  B6                 DB 0B6H
                         ;CALL AE
3B5F  0E 00              LD C,0
3B61  20 0D              JR NZ,NUML2
3B63  0C           NUML1:  INC C
3B64  FE 2C              CP ','
3B66  20 08              JR NZ,NUML2
3B68  23                 INC HL
3B69  C5                 PUSH BC
3B6A  EF                 RST JT
3B6B  B6                 DB 0B6H
                         ;CALL AE
3B6C  C1                 POP BC
3B6D  28 F4              JR Z,NUML1
3B6F  2B                 DEC HL
3B70  C3 3C38      NUML2:  JP SYSUCC


                         ;
3B73  23           OVERHL: INC HL
3B74  7E           OVERSP: LD A,(HL) ;INCREMENTS HL UNTIL
3B75  22 FD84            LD (ERRPOS),HL
3B78  FE FF              CP ENDMARK ;A NON SPACE IS FOUND
3B7A  C8                 RET Z
3B7B  FE C3              CP ELSE
3B7D  C8                 RET Z
3B7E  FE 20              CP 32
3B80  C0                 RET NZ
3B81  18 F0              JR OVERHL
3B83  7E           ALPH2:  LD A,(HL)
3B84  FE 20              CP 32
3B86  D8                 RET C
3B87  7E           ALPHA:  LD A,(HL)      ;RETURNS CARRY IMPLIES ALPHA
3B88  FE 41        ALPHA1: CP 65
3B8A  38 05              JR C,ALPHEND
3B8C  FE 5B              CP 91
3B8E  D8                 RET C
3B8F  FE 61              CP 97
3B91  3F           alphend:      ccf
```

check valid variable
load d reg
Success exit if not
comma separator
repeat.
List of variable names.

ARITHMETIC EXPRESSIONS

Tests for a valid
list of numeric
expressions.

Finds next non-space
character.
Zero flag set if ENDMARK
or ELSE

Returns carry reset
if control byte.

Returns carry flag set if (HL) =
A contains A-Z or a-z

```
3B92  D0                    ret nc
3B93  FE 7B                 CP 123
3B95  C9                    RET
3B96  7E              DIGIT:  LD A,(HL)       ;CARRY IMPLIES DIGIT
3B97  FE 30           DIGIT1: CP 48
3B99  3F                    CCF
3B9A  D0                    RET NC
3B9B  FE 3A                 CP 58
3B9D  C9                    RET
3B9E  E5              NUMIN1: PUSH HL
3B9F  CD 3BBB         NUMBER: CALL FIXED
3BA2  20 5D           SYFL:   JR NZ,SYFLA ;NO NUMBER FOUND SO IGNORE
3BA4  FE 45           EXPON:  CP 69    ;EXPONENT PART
3BA6  20 C8                   JR NZ,NUML2
3BA8  23                      INC HL
3BA9  DF                      RST OSPRST
3BAA  CD 3BCB                 CALL SINTEG
3BAD  3F                      CCF
3BAE  9F                      SBC A,A ;C IMPLIES SUCCESS
3BAF  E7                      RST SYNRST
                      ;
3BB0  FE 2E           DECFRAC:        CP 46    ;CARRY IMPLIES FRAC
3BB2  28 02                   JR Z,DF1
3BB4  A7                      AND A    ;FAIL. CLEAR CARRY
3BB5  C9                      RET      ;AT LEAST DEC POINT
3BB6  23              DF1:    INC HL
3BB7  CD 3BCE                 CALL INTEG
3BBA  C9                      RET
                      ;
3BBB  E5              FIXED:  PUSH HL
3BBC  CD 3BB0                 CALL DECFRAC
3BBF  38 AF                   JR C,NUML2
3BC1  CD 3BCE                 CALL INTEG
3BC4  30 34                   JR NC,SYFLB
3BC6  CD 3BB0                 CALL DECFRAC
3BC9  18 A5                   JR NUML2
                      ;
3BCB  CD 3BD8         SINTEG: CALL ADOP
3BCE  EF              INTEG:  RST JT
3BCF  B7                      DB 0B7H
                      ;CALL DIGIT     ;CARRY IMPLIES INTEGER
3BD0  D0                      RET NC  ;SPACES ARE IGNORED
3BD1  23              INTEG1: INC HL
3BD2  EF                      RST JT
3BD3  B7                      DB 0B7H
                      ;CALL DIGIT
3BD4  38 FB                   JR C,INTEG1
3BD6  3F                      CCF
3BD7  C9                      RET

3BD8  FE CF           ADOP:   CP 207
3BDA  28 03                   JR Z,ADOP1
3BDC  FE D0                   CP 208
3BDE  C0                      RET NZ
3BDF  23              ADOP1:  INC HL
3BE0  C9                      RET
```

Returns carry set if (HL) = A contains a digit

Test for valid number

Succeeds if no exponent

Test for valid signed integer

Test for decimal point followed by fractional part

Test for valid integer

Test for decimal fraction only

Test for integer part followed by decimal fraction

Test for valid digit

Skip remaining digits

If + or - advance character pointer and return with Zero set. Otherwise no effect

```
                        ;
3BE1  E5                ARITHOP:    PUSH HL
3BE2  FE CF                         CP 207
3BE4  38 5A                         JR C,SYFAIL
3BE6  FE D4                         CP 212
3BE8  18 07                         JR OPCON
                        ;
3BEA  E5                RELOP:  PUSH HL         ;212-217
3BEB  FE D4                     CP 212
3BED  38 51                     JR C,SYFAIL
3BEF  FE DA                     CP 218
3BF1  38 44             OPCON:  JR C,SYSUC1
3BF3  18 4B                     JR SYFAIL
3BF5  DF                IDENT0: RST OSPRST
3BF6  E5                IDENT:  PUSH HL
3BF7  CD 3B83                   CALL ALPH2 ;LOOKS FOR IDENTIFIER
3BFA  30 44             SYFLB:  JR NC,SYFAIL ;C IMPLIES FOUND
3BFC  E5                        PUSH HL
3BFD  CD 3C0C                   CALL IDENT1
3C00  D1                        POP DE
3C01  20 3D             SYFLA:  JR NZ,SYFAIL
3C03  CB 41                     BIT 0,C
3C05  20 31                     JR NZ,SYSUCC
3C07  EB                        EX DE,HL
3C08  CB B6                     RES 6,(HL)
3C0A  EB                        EX DE,HL
3C0B  E7                        RST SYNRST

3C0C  E5                IDENT1: PUSH HL
3C0D  0E 01             IDENT2: LD C,1
3C0F  23                        INC HL
3C10  EF                        RST JT
3C11  B7                        DB 0B7H
                        ;CALL DIGIT
3C12  38 F9                     JR C,IDENT2
3C14  EF                        RST JT
3C15  BD                        DB 0BDH
                        ;CALL ALPHA
3C16  38 F5                     JR C,IDENT2
3C18  DF                        RST OSPRST ;FOUND IDENTIFIER HL POINTS
3C19  FE 24                     CP '$'  ;AT NEXT NON SPACE
3C1B  20 03                     JR NZ,SSSUBS
3C1D  23                        INC HL
3C1E  0C                        INC C   ;3.X$
3C1F  DF                        RST OSPRST
3C20  FE 28             SSSUBS: CP'('
3C22  20 14                     JR NZ,SYSUCC
3C24  0C                        INC C
3C25  0C                        INC C
3C26  23                        INC HL
3C27  C5                        PUSH BC
3C28  CD 3B5C                   CALL NUMLIST
3C2B  AF                        XOR A
3C2C  B9                        CP C
3C2D  20 02                     JR NZ,SUBS2
3C2F  EF                SUBS19: RST ERROR
```

Test for arithmetic operation

Test for relational operator

Error if not letter or control code

Identify variable

Succeed if numeric

Convert leading byte of char variable name to control code

Identify variables and flag leading byte of string variable names.

— Get character

— Save position

— Stack Pointer.

Look for first non letter or number and set C=1

If character is $ increment C and load next

Success return if not (

C = C+2

Test for a valid subscript list. Fail if not.

Test for valid variable and identify type

```
3C30  06              DB 6
3C31  C1        SUBS2: POP BC
3C32  DF               RST OSPRST
3C33  FE 29            CP ')'
3C35  20 F8            JR NZ,SUBS19
                ;
3C37  23        SYSUC1: INC HL
3C38  AF        SYSUCC: XOR A   ;SET Z FLAG
3C39  33               INC SP
3C3A  33               INC SP
3C3B  22 FD84          LD (ERRPOS),HL
3C3E  7E               LD A,(HL)
3C3F  C9               RET
3C40  AF        SYFAIL: XOR A   ;SET NZ FLAG
3C41  3C               INC A
3C42  E1               POP HL ;AND RESTORE POINTER
3C43  7E               LD A,(HL)
3C44  C9               RET
                ;
3C45  E5        AE:    PUSH HL
3C46  DF        AE0:   RST OSPRST
3C47  FE 29            CP ')'
3C49  20 02            JR NZ,AE1
3C4B  EF               RST ERROR
3C4C  01               DB 1
3C4D  CD 3BD8   AE1:   CALL ADOP
3C50  DF        AE2:   RST OSPRST
3C51  CD 3C60          CALL PRIMARY
3C54  20 EA            JR NZ,SYFAIL
3C56  DF               RST OSPRST
3C57  28 DF            JR Z,SYSUCC
3C59  CD 3BE1          CALL ARITHOP
3C5C  28 F2            JR Z,AE2
3C5E  18 D8     SYSCA: JR SYSUCC
3C60  E5        PRIMARY:       PUSH HL
3C61  DF               RST OSPRST
3C62  FE 2E            CP 46
3C64  CA 3B9F          JP Z,NUMBER
3C67  EF               RST JT
3C68  B7               DB 0B7H
                ;CALL DIGIT
3C69  DA 3B9F          JP C,NUMBER
3C6C  EF        PRIMVAR:       RST JT
3C6D  BD               DB 0BDH
                ;CALL ALPHA
3C6E  30 0B            JR NC,PRIMBOO
3C70  CD 3BF6          CALL IDENT
3C73  20 CB            JR NZ,SYFAIL
3C75  3E FF            LD A,0FFH
3C77  A9               XOR C
3C78  CB 47            BIT 0,A
3C7A  E7               RST SYNRST
3C7B  CD 3D70  PRIMBOO:       CALL BOOPR
3C7E  28 B8            JR Z,SYSUCC
3C80  FE DD    PRIMFUNC:      CP 221
3C82  38 04            JR C,PRIMPAR
```

Test for). Fail if not.

Load old pointer. Advance error pointer.

Exit Points for Error Checker Routines

Restore old pointer

Get character. Correct)

Skip unitary +, -. Get character. Test for primary numeric types. Fail if not. Succeed if no more. Test for arithmetic operator. Check second parameter.

If decimal point jump to number. If digit jump to number.

If not letter jump to PRIMBOO.

Check variable syntax.

Test for bracketed boolean expression.

C = 1 Simple Variable
C = 2 Simple String
C = 3 Numeric Array
C = 4 String Array

Test for valid Arithmetic Expression

```
3C84  FE F5           CP 245
3C86  38 54           JR C,PARAMS
3C88  FE 28     PRIMPAR:    CP '('
3C8A  20 B4           JR NZ,SYFAIL
3C8C  23              INC HL
3C8D  EF              RST JT
3C8E  86              DB 0B6H
                      ;CALL AE
3C8F  18 39           JR PAREND
                      ;
3C91  E5        SE:   PUSH HL
3C92  DF        SE2:  RST OSPRST
3C93  CD 3CA0         CALL SPRIM
3C96  20 A8     SYFLC: JR NZ,SYFAIL
3C98  DF              RST OSPRST
3C99  FE CF           CP 207 ;+
3C9B  20 9B           JR NZ,SYSUCC
3C9D  23              INC HL
3C9E  18 F2           JR SE2
3CA0  E5        SPRIM: PUSH HL
3CA1  DF              RST OSPRST
3CA2  FE 22           CP '"'
3CA4  20 0F           JR NZ,SPRIMVAR
3CA6  E1              POP HL
3CA7  E5        SPRLIT: PUSH HL
3CA8  23        SPRIMLIT:   INC HL
3CA9  7E              LD A,(HL)
3CAA  FE FF           CP ENDMARK
3CAC  28 92           JR Z,SYFAIL
3CAE  FE 22           CP '"'
3CB0  20 F6           JR NZ,SPRIMLIT
3CB2  23              INC HL
3CB3  18 A9     SYSCB: JR SYSCA
                      ;
3CB5  CD 3B83   SPRIMVAR:   CALL ALPH2
3CB8  30 08           JR NC,SPRIMPAR
3CBA  CD 3BF6         CALL IDENT
3CBD  20 D7           JR NZ,SYFLC
3CBF  CB 41           BIT 0,C
3CC1  E7              RST SYNRST
3CC2  FE 28     SPRIMPAR:   CP '('
3CC4  20 0C           JR NZ,SPRIMFUNC
3CC6  23              INC HL
3CC7  7E              LD A,(HL)
3CC8  EF              RST JT
3CC9  BE              DB 0BEH
                      ;CALL SE
3CCA  C2 3C96   PAREND: JP NZ,SYFLC
3CCD  DF              RST OSPRST
3CCE  FE 29           CP ')'
3CD0  23              INC HL
3CD1  E7              RST SYNRST
3CD2  FE F5     SPRIMFUNC:  CP 245
3CD4  DA 3C96         JP C,SYFLC
3CD7  FE FE           CP 254
3CD9  D2 3C40         JP NC,SYFAIL
```

Test for numeric function

Test for bracketed Numeric expression

- Get character
Test for valid string

Test for + operator
Return success if not found

Test for end of valid string

Skip to end of quoted string. Fail if ENDMARK found.

Jump if not double quote

Jump if not letter or control code

Check variable syntax

Jump if not (

Try again within brackets

Check for close bracket

Test for string function

Test for valid character expression

```
3CDC  23           PARAMS: INC HL
3CDD  CD 3D0B              CALL CARAC
3CE0  17                   RLA
3CE1  30 D0                JR NC,SYSCB
3CE3  F5           PA1:    PUSH AF
3CE4  DF                   RST 0SPRST
3CE5  FE 28                CP '('
3CE7  20 20                JR NZ,PARERR
3CE9  23           PA2:    INC HL
3CEA  F1                   POP AF
3CEB  17                   RLA
3CEC  F5                   PUSH AF
3CED  30 04                JR NC,STR
3CEF  EF                   RST JT
3CF0  B6                   DB 0B6H
                   ;CALL AE
3CF1  18 02                JR PA3
3CF3  EF           STR:    RST JT
3CF4  BE                   DB 0BEH
                   ;CALL SE
3CF5  20 12        PA3:    JR NZ,PARERR
3CF7  F1                   POP AF
3CF8  17                   RLA
3CF9  30 0B                JR NC,PA4
3CFB  F5                   PUSH AF
3CFC  DF                   RST 0SPRST
3CFD  FE 2C                CP ','
3CFF  20 0B                JR NZ,PARERR
3D01  18 E6                JR PA2
3D03  DF           PA4:    RST 0SPRST
3D04  FE 29                CP ')'
3D06  23                   INC HL
3D07  28 AA                JR Z,SYSCB
3D09  EF           PARERR: RST ERROR
3D0A  00                   DB 0
3D0B  FE ED        CARAC:  CP 237 ;NEEDS TO BE ADJUSTED
3D0D  30 03                JR NC,CAR1
3D0F  3E C0                LD A,0C0H
3D11  C9                   RET
3D12  4F           CAR1:   LD C,A
3D13  06 00                LD B,0
3D15  E5                   PUSH HL
3D16  21 3C30              LD HL,CTABLE-237
3D19  09                   ADD HL,BC
3D1A  7E                   LD A,(HL)
3D1B  E1                   POP HL
3D1C  C9                   RET
3D1D  B0 B0 B0 B0  CTABLE:DB B0H,B0H,B0H,B0H,0F0H,0,0,0,0C0H,0,0,0B0H,0BCH,0B0H,0FCH,0C0H,0
3D21  F0 00 00 00
3D25  C0 00 00 B0
3D29  BC B0 FC C0
3D2D  00
                   ;ASC,LEN,VAL,VARPTR,MOD,PI,RND,,CHR,SPK,INKEY,LEFT,MID,RT,GREED,STR,TI
                   ;
                   ;
3D2E  E5           RELAT:  PUSH HL
```

Get parameter byte
Return success if msb not set

Get next character and test if not (

Test next bit.
If set test for arithmetic expression else test for string expression

Test for another parameter

Test for comma separator. Fail if not found

Test for) terminator and fail if not found.

Test for valid function parameters.

Returns a parameter byte depending on function op-code in A. Parameter byte consists of bit pairs. First bit set indicates another parameter. Second bit set indicates numeric parameter.

```
3D2F  EF              RST JT          } Test for string
3D30  BE              DB 0BEH         } expression and
                    ;CALL SE          } return if not
3D31  20 07           JR NZ,RELBOO    }
3D33  CD 3BEA         CALL RELOP      } Test for relational operator. Fail if not
3D36  CC 3C91         CALL Z,SE       } Test for second string expression
3D39  E7              RST SYNRST      } and return result
3D3A  CD 3D70       RELBOO: CALL BOOPR  - Test for boolean expressions
3D3D  20 0B           JR NZ,RELNUM    - Jump if not
3D3F  CD 3BEA         CALL RELOP      } Test for relational operation,
3D42  C2 3C38       SYSCC:  JP NZ,SYSUCC} Succeed if not
3D45  DF              RST OSPRST      - Get next character
3D46  CD 3D70         CALL BOOPR      } Test for another valid
3D49  E7              RST SYNRST      } relational and return status
3D4A  EF            RELNUM: RST JT    } Test for arithmetic
3D4B  B6              DB 0B6H         } expression
                    ;CALL AE
3D4C  CC 3BEA         CALL Z,RELOP    - Test for relational operation
3D4F  CC 3C45         CALL Z,AE       - Test for second arithmetic expression
3D52  E7              RST SYNRST      - Return status
                    ;
3D53  E5            BOOLEAN:    PUSH HL
3D54  DF            BOO:    RST OSPRST  }
3D55  FE DC           CP NNOT         } Test and skip
3D57  20 02           JR NZ,BOOO      } NOT operation
3D59  23              INC HL          }
3D5A  DF              RST OSPRST      }
3D5B  CD 3D2E       BOOO:   CALL RELAT  - Test for relational
3D5E  C4 3D70         CALL NZ,BOOPR   - Test for bracketed boolean
3D61  C2 3C40       SYFLD:  JP NZ,SYFAIL - Fail if not one or other
3D64  DF            BOO2:   RST OSPRST  }
3D65  FE DA         BOO3:   CP NAND     } Test for AND or OR
3D67  28 04           JR Z,BOO4       } Succeed if not
3D69  FE DB           CP NOR          }
3D6B  20 D5           JR NZ,SYSCC     }
3D6D  23            BOO4:   INC HL      } Repeat for second
3D6E  18 E4           JR BOO          } parameter.
                    ;
3D70  E5            BOOPR:  PUSH HL
3D71  FE 28           CP '('          } Error if not (
3D73  20 EC           JR NZ,SYFLD     }
3D75  23              INC HL          }
3D76  CD 3D53         CALL BOOLEAN    - Test for boolean expression
3D79  FE 29           CP ')'          } Error if not )
3D7B  23              INC HL          } and exit
3D7C  E7              RST SYNRST      }
                    ;
3D7D  E5            INT64:  PUSH HL     Save pointer    }
3D7E  EB              EX DE,HL        Pointer to DE   } Gets an integer number
3D7F  CD 121D         CALL INT64K     Get number      } <64K. Zero flag set if
3D82  EB              EX DE,HL        New pointer back to HL } valid
3D83  E7              RST SYNRST      RST 08
                    ;
                    ;
0028                JT      EQU     28H
0028                ERRORS  EQU     28H
```

Test for valid relational expression

Test for valid boolean expression

```
00CF                    OPBASE  EQU     207
00F4                    COMP    EQU     244
00DC                    NOT     EQU     220
                        ;
                        ;
                        ;
                        ;
3D84                    EVALAB:
3D84    21 0000                 LD HL,0
3D87    E5              EVAL1:  PUSH HL
3D88    1A              EVAL2:  LD A,(DE)
3D89    CD 124F         EV0:    CALL TOKMIN             ;Ignore +, stack comp
3D8C    38 04                   JR C,EV1
3D8E    21 F405                 LD HL,COMP*256+5
3D91    E5              EV01:   PUSH HL
3D92    1A              EV1:    LD A,(DE)
3D93    CD 3B97                 CALL DIGIT1
3D96    30 08                   JR NC,EV2               ;Jump if not a number
                        ;
                        ;Must be a number
                        ;
3D98    CD 224C         EV11:   CALL MOVEACC
                        ;       RST JT
                        ;       DB 085H
3D9B    CD 1286          CALL STACKACC          ;Put number on stack
3D9E    18 4E                   JR EV8
                        ;
3DA0    FE 2E           EV2:    CP '.'
3DA2    28 F4                   JR Z,EV11               ;Jump if decimal point
                        ;
3DA4    CD 3B88         EV3:    CALL ALPHA1
3DA7    30 08                   JR NC,EV4               ;Jump if not a character
                        ;
                        ;Must be a variable
                        ;
3DA9    CD 3037                 CALL PSHNAM             ;Put variable name on stack
3DAC    CD 3066                 CALL GETVAL
3DAF    18 3D                   JR EV8
                        ;
3DB1    FE 28           EV4:    CP '('
3DB3    20 06                   JR NZ,EV5
3DB5    13                      INC DE
3DB6    EF                      RST JT
3DB7    80                      DB 80H
                        ;CALL EVALAB            ;Evaluate expression
3DB8    13                      INC DE
3DB9    18 33                   JR EV8
                        ;
3DBB    FE 29           EV5:    CP ')'
3DBD    28 6C                   JR Z,DOWNST             ;Jump if end of expression
                        ;
3DBF    FE 80           EV6:    CP 128
3DC1    30 20                   JR NC,EV7
3DC3    EF              EV65:   RST JT
3DC4    BB                      DB 0BBH
                        ;       CALL EVALSE
```

```
3DC5  1A                      LD A,(DE)
3DC6  D6 CF                   SUB OPBASE
3DC8  21 0D72                 LD HL,OPTABLE
3DCB  CD 0689                 CALL FINDJP
3DCE  E5                      PUSH HL
3DCF  13                      INC DE
3DD0  EF                      RST JT
3DD1  BB                      DB 0BBH
                          ;CALL EVALSE
3DD2  D5                      PUSH DE
3DD3  CD 3F5B                 CALL COMP$
3DD6  E1                      POP HL
3DD7  E3                      EX (SP),HL
3DD8  23                      INC HL
3DD9  23                      INC HL
3DDA  23                      INC HL
3DDB  CD 3E3B                 CALL DOJ1
3DDE  EF                      RST JT
3DDF  85                      DB 85H
                          ;CALL STACKACC
3DE0  D1                      POP DE
3DE1  18 0B                   JR EV8
                          ;
3DE3  FE DD               EV7:  CP 221
3DE5  38 20                   JR C,OP         ;Jump if not a function
3DE7  FE F5                   CP 245          ;First string function no.
3DE9  30 D8                   JR NC,EV65
                          ;
3DEB  CD 3E5B             FUNC: CALL EVALFN         ;Evaluate function and fall through to
                          ;
3DEE  1A                  EV8:  LD A,(DE)           ;Get next char
3DEF  FE 80                   CP 128
3DF1  38 38                   JR C,DOWNST         ;Must be , or ; or :
                          ;
3DF3  FE FF                   CP 0FFH
3DF5  28 34                   JR Z,DOWNST         ;Jump if end of line
                          ;
3DF7  18 0E                   JR OP               ;Must be operator or ELSE or THEN
                          ;
                          ;
                          ;           +,-,*,/,^,=,=,=,=,=,=,=,A,O,N
3DF9  05 05 06 06         OPST: DB    5,5,6,6,7,4,4,4,4,4,4,4,2,1,3
3DFD  07 04 04 04
3E01  04 04 04 02
3E05  01 03
                          ;
3E07  FE CF               OP:   CP 207
3E09  38 20                   JR C,DOWNST
3E0B  06 00                   LD B,0
3E0D  4F                      LD C,A
3E0E  21 3D2A                 LD HL,OPST-207
3E11  09                      ADD HL,BC
3E12  6E                      LD L,(HL)       ;PREC
3E13  67                      LD H,A  ;OP
3E14  7D                      LD A,L
3E15  E3                  TOP:  EX (SP),HL
```

```
3E16  BD                    CP L
3E17  28 02                 JR Z,TOP1
3E19  30 0A                 JR NC,CONT
3E1B  7C              TOP1: LD A,H
3E1C  D5                    PUSH DE
                      ;     CALL PAR2
3E1D  CD 3E3C               CALL PARSSS
3E20  D1                    POP DE
3E21  E1                    POP HL
3E22  7D                    LD A,L
3E23  18 F0                 JR TOP
3E25  E3              CONT: EX (SP),HL
3E26  13                    INC DE
3E27  E5                    PUSH HL
3E28  C3 3D88               JP EVAL2
3E2B  F1              DOWNST: POP AF
3E2C  A7                    AND A
3E2D  C8                    RET Z               ;Return if A = 0
3E2E  CD 3E3C               CALL PARSSS
3E31  18 F8                 JR DOWNST
                    ;
                    ;       CP NOT
                    ;       JR Z,DS2
                    ;       CP COMP
                    ;       JR Z,DS2
                    ;DS1:   CALL PAR2
                    ;       JR DOWNST
                    ;DS2:   CALL PAR1
                    ;       JR DOWNST
3E33  D6 CF         DOJMP1: SUB OPBASE
3E35  21 0D72               LD HL,OPTABLE
3E38                DOJUMP:
3E3B  CD 0689               CALL FINDJP
3E3B  E9            DOJ1:   JP (HL)
                    ;
                    ;
                    ;Calling ARITH executes the instruction in A
                    ;
                    ;
                    ;
3E3C  FE DC         PARSSS: CP NOT
3E3E  28 0C                 JR Z,PAR1
3E40  FE F4         PARRRR: CP COMP
3E42  28 0B                 JR Z,PAR1
3E44                PAR2:;  RST JT
                    ;       DB 0C1H,84H
3E44  CD 128F                CALL ACCSTACK
3E47  CD 129E                CALL OP1STACK
3E4A  18 02                 JR PAR0
3E4C  EF            PAR1:   RST JT
3E4D  81                    DB 81H
                    ;CALL ACCSTACK
3E4E  D5            PAR0:   PUSH DE
3E4F  F5                    PUSH AF
3E50  CD 3E33               CALL DOJMP1
3E53  F1                    POP AF
```

```
3E54  D1                    POP DE
3E55  FE F5                 CP 245
3E57  D0                    RET NC
3E58  C3 1286               JP STACKACC
                      ;
                      ;
                      ;
3E5B                  EVALFN:
3E5B  F5                    PUSH AF
3E5C  CD 3D0B               CALL CARAC
3E5F  13                    INC DE
3E60  17              EVFN0:  RLA
3E61  30 10                 JR NC,NOPARAMS
3E63  13              EVFN1:  INC DE
3E64  17                    RLA
3E65  F5                    PUSH AF
3E66  D4 3E7E               CALL NC,EVALSE
3E69  F1                    POP AF
3E6A  F5                    PUSH AF
3E6B  DC 3D84               CALL C,EVALAB
3E6E  F1              EVFN3:  POP AF
3E6F  17                    RLA
3E70  38 F1                 JR C,EVFN1
3E72  13                    INC DE
3E73                  NOPARAMS:
3E73  F1                    POP AF
3E74  FE ED                 CP 237
3E76  38 D4                 JR C,PAR1
3E78  FE F1                 CP 241
3E7A  28 C8                 JR Z,PAR2
3E7C  18 D0                 JR PAR0
                      ;
                      ;
                      ;String expression routines
                      ;
                      ;
3E7E  2A FA81         EVALSE: LD HL,(CALCST)
3E81  CD 3FE1               CALL FIND2$
3E84  1A              NEXPR:  LD A,(DE)
3E85  FE CF                 CP 207
3E87  C0                    RET NZ
3E88  C5                    PUSH BC         ;Save length
3E89  13                    INC DE
3E8A  22 FA81               LD (CALCST),HL
3E8D  CD 3FE1               CALL FIND2$
3E90  E3                    EX (SP),HL              ;Get length
3E91  09                    ADD HL,BC
3E92  44                    LD B,H
3E93  4D                    LD C,L
3E94  E1                    POP HL
3E95  E5                    PUSH HL         ;Save HL
3E96  CD 3EB9               CALL NULL$
3E99  E1                    POP HL          ;Restore HL
3E9A  18 EB                 JR NEXPR
                      ;
```

```
                        ;This puts the string pointed to
                        ;by DE onto the calculator stack
                        ;without its length
                        ;
3E9C  1A                GETST:  LD A,(DE)
3E9D  FE 22                     CP '"'
3E9F  20 1B                     JR NZ,SEVAR
3EA1  01 0000           SEQ0:   LD BC,0
3EA4  13                SEQ1:   INC DE
3EA5  1A                        LD A,(DE)
3EA6  FE 22                     CP '"'
3EA8  28 0E                     JR Z,SEQEND
3EAA  FE FF                     CP 0FFH
3EAC  20 02                     JR NZ,SEQ2
3EAE  EF                        RST ERRORS
3EAF  35                        DB 35H
3EB0  CD 3EC9           SEQ2:   CALL STKTEST            ;Preserves HL
3EB3  77                        LD (HL),A
3EB4  03                        INC BC
3EB5  23                        INC HL
3EB6  18 EC                     JR SEQ1
3EB8  13                SEQEND: INC DE
                        ;
                        ;
3EB9  C3 315E           NULL$:  JP STKLN1               ;Put length on stack and return
                        ;
                        ;
3EBC  FE 80             SEVAR:  CP 128
3EBE  30 06                     JR NC,SEFUN
3EC0  CD 3037                   CALL PSHNAM
3EC3  C3 3066                   JP GETVAL
                        ;
3EC6  C3 3E5B           SEFUN:  JP EVALFN
                        ;
                        ;
3EC9  D5                STKTEST:        PUSH DE
3ECA  E5                        PUSH HL
3ECB  C3 32A1                   JP CHKST1               ;Test if HL > (STKLIM) and return
                        ;
                        ;
                        ;
3ECE  AF                INKEY$: XOR A
3ECF  32 FD7C                   LD (LASTKY),A
3ED2  CD 0079                   CALL KBD
3ED5  C2 0A41                   JP NZ,CHR1              ;Jump to CHR1 if key pressed
3ED8  2A FA81                   LD HL,(CALCST)
3EDB  18 1A                     JR MID11                ;Put length of zero on stack and return
                        ;
                        ;
                        ;
3EDD  CD 3FF9           MID$:   CALL GETPAR
3EE0  C5                        PUSH BC         ;Save param. 2
3EE1  CD 3FF9                   CALL GETPAR
3EE4  C5                        PUSH BC         ;Save param. 1
                        ;
3EE5  EF                        RST JT
```

```
3EE6  B9                    DB 0B9H
                          ;CALL FIND1$          ;BC=length, DE=start of string
3EE7  62                    LD H,D
3EE8  6B                    LD L,E                ;HL=start
3EE9  D1                    POP DE                ;DE=param. 1
3EEA  E3                    EX (SP),HL            ;HL=param. 2, (SP)=start
3EEB  C5                    PUSH BC         ;Save length
3EEC  E3                    EX (SP),HL            ;HL=length, (SP)=param. 2
3EED  C1                    POP BC                ;BC=param.2, (SP)=start
                          ;
                          ;Here   BC = param. 2, DE = param. 1, HL = length, (SP) = start
                          ;
3EEE  7A                    LD A,D
3EEF  B3                    OR E                  ;If param. 1 = 0 then error
3EF0  28 26                 JR Z,MID$ERR
                          ;
3EF2  78                    LD A,B
3EF3  B1                    OR C                  ;If param. 2 = 0 then null string
3EF4  20 06                 JR NZ,MID2
                          ;
3EF6  E1                  MID1:   POP HL                ;Pop start address
3EF7  01 0000             MID11:  LD BC,0         ;Zero string length
3EFA  18 BD                 JR NULL$
                          ;
3EFC  A7                  MID2:   AND A
3EFD  ED 52                 SBC HL,DE             ;HL=length - param. 1
3EFF  38 F5                 JR C,MID1             ;If length < param. 1 then null string
3F01  1B                    DEC DE                ;DE=param. 1 - 1
3F02  23                    INC HL                ;HL=length - (param.1 - 1)
                          ;Carry flag is already reset
3F03  ED 42                 SBC HL,BC             ;HL=length - (param. 1 - 1 + param. 2)
3F05  30 03                 JR NC,MID3            ;If HL)=0 then param. 2 is alright, else
3F07  09                    ADD HL,BC             ;HL=truncated param. 2
3F08  44                    LD B,H
3F09  4D                    LD C,L                ;BC=truncated param. 2
                          ;
3F0A  E1                  MID3:   POP HL                ;HL=start address
3F0B  EB                    EX DE,HL              ;DE=start, HL = param.1 - 1 = offset
3F0C  19                    ADD HL,DE             ;HL=address of 1st char. required
                          ;
3F0D  78                  MID4:   LD A,B
3F0E  B1                    OR C
3F0F  28 04                 JR Z,ENDMID           ;Jump if null string
3F11  C5                  MID5:   PUSH BC         ;Save length of string
3F12  ED B0                 LDIR                  ;Move chars
3F14  C1                    POP BC          ;Restore new length
                          ;
3F15  EB                  ENDMID: EX DE,HL              ;HL=calc stack pointer
3F16  18 A1                 JR NULL$              ;Store string length
                          ;
3F18  EF                  MID$ERR:        RST ERRORS              ;Parameter 1 = 0
3F19  24                    DB 36           ;"SUBSCRIPT OUT OF RANGE"
                          ;
                          ;
3F1A  CD 3F27             RIGHT$: CALL LEFTRIGHT  ;BC=no. of chars, DE=start, HL=offset
3F1D  19                    ADD HL,DE             ;HL=address of first char required
```

```
3F1E  18 ED                   JR MID4         ;Move chars
                              ;
                              ;
3F20  CD 3F27          LEFT$: CALL LEFTRIGHT          ;BC,DE,HL - see above
3F23  62                      LD H,D
3F24  6B                      LD L,E                  ;HL=start of string
3F25  18 E6                   JR MID4         ;Move chars
                              ;
                              ;
3F27                   LEFTRIGHT:
3F27  CD 3FF9                 CALL GETPAR
3F2A  C5                      PUSH BC         ;Save no. of chars
3F2B  EF                      RST JT
3F2C  B9                      DB 0B9H
                       ;CALL FIND1$           ;BC=length of string
3F2D  60                      LD H,B
3F2E  69                      LD L,C          ;HL=length of string
3F2F  C1                      POP BC          ;BC=no. of chars
3F30  A7                      AND A
3F31  ED 42                   SBC HL,BC               ;HL=length - no. of chars
3F33  38 01                   JR C,LR1                ;Jump if length < no. of chars
3F35  C0                      RET NZ          ;Return if length > no. of chars
                              ;
3F36  E1               LR1:   POP HL          ;Clear stack and return
3F37  C9                      RET                     ;(string is unchanged)
                              ;
                              ;
3F38  01 0006          TIME$: LD BC,6         ;BC=length of time string
3F3B  CD 3299                 CALL CHKSTK             ;Is there room for string on calc stack ?
                              ;
                       ;Carry reset if returned to here
                              ;
3F3E  2A FA81                 LD HL,(CALCST)
3F41  C5                      PUSH BC
3F42  EB                      EX DE,HL                ;DE = (CALCST)
3F43  21 FD57                 LD HL,CLOCK             ;HL = CLOCK
3F46  ED B0                   LDIR
3F48  C1                      POP BC
3F49  18 CA                   JR ENDMID
                              ;
                              ;
                              ;
3F4B  EF               ASC:   RST JT
3F4C  B9                      DB 0B9H
                       ;CALL FIND1$
3F4D  ED 53 FA81              LD (CALCST),DE
3F51  20 04                   JR NZ,ASC1
3F53  3D                      DEC A
3F54  C3 1317                 JP STAC
3F57  1A               ASC1:  LD A,(DE)
3F58  C3 1311                 JP STA
                              ;
                              ;
3F5B  EF               COMP$: RST JT
3F5C  B9                      DB 0B9H
```

```
                        ;CALL FIND1$
3F5D  D5                        PUSH DE         ;Save start of string 2
3F5E  C5                        PUSH BC         ;Save length of string 2
3F5F  EB                        EX DE,HL                ;HL = start of 2
3F60  CD 3FEC                   CALL FIND$              ;BC=length of 1, DE=start of 1
3F63  EB                        EX DE,HL                ;HL=start of 1
3F64  22 FAB1                   LD (CALCST),HL  ;Reset calc stack to start of string 1
3F67  E3                        EX (SP),HL              ;HL=length of 2, (SP)=start of 1
3F68  54                        LD D,H
3F69  5D                        LD E,L          ;DE=length of 2, BC=length of 1
                        ;
3F6A  A7                        AND A
3F6B  ED 42                     SBC HL,BC               ;HL=length of 2 - length of 1
3F6D  20 07                     JR NZ,COMP2             ;Jump if lengths not equal
                        ;
3F6F  78                        LD A,B
3F70  B1                        OR C
3F71  20 15                     JR NZ,COMP4             ;Jump if lengths > 0
                        ;
3F73  D1                COMP1:  POP DE
3F74  E1                COMP1P1:        POP HL
3F75  C9                        RET                     ;Clear stack then return
                        ;
3F76  38 02             COMP2:  JR C,COMP3              ;Jump if length of 2 < length of 1
                        ;
                        ;Length of 2 > length of 1
                        ;
                        ;       LD A,B          ;BC=length of 1
                        ;       OR C
                        ;       JR NZ,COMP4             ;Jump if length of 1 > 0
                        ;
                        ;       INC A                   ;Reset zero flag (A=1)
                        ;       SCF                     ;Shows that string 1 < string 2
                        ;       JR COMP1
                        ;
                        ;Length of 2 < length of 1
                        ;
                        ;COMP3: LD B,D
                        ;       LD C,E          ;BC=length of 2
                        ;
                        ;       LD A,D          ;DE=length of 2
                        ;       OR E                    ;Carry flag reset
                        ;       JR NZ,COMP4             ;Jump if length of 2 > 0
                        ;
                        ;       INC A                   ;Reset zero flag (A=1)
                        ;       JR COMP1                ;Carry reset, i.e. string 1 > string 2
                        ;
                        ;BC=shortest of the two string lengths
                        ;
3F78  42                        LD B,D
3F79  4B                        LD C,E
3F7A  E1                COMP3:  POP HL
3F7B  D1                        POP DE
3F7C  F5                        PUSH AF
3F7D  78                        LD A,B
3F7E  B1                        OR C
```

```
3F7F  28 05              JR Z,LONGEST
3F81  CD 3F8A            CALL COMP5
3F84  20 EE              JR NZ,COMP1P1
3F86  F1          LONGEST:  POP AF
3F87  C9                 RET
                  ;
                  ;
                  ;
3F88  E1          COMP4:  POP HL          ;DE=start of string 1
3F89  D1                  POP DE          ;HL=start of string 2
                  ;
3F8A  1A          COMP5:  LD A,(DE)             ;Get char from string 1
3F8B  BE                  CP (HL)         ;Compare with char from string 2
3F8C  C0                  RET NZ          ;Return if mismatch
3F8D  13                  INC DE
3F8E  23                  INC HL          ;Get next chars
3F8F  0B                  DEC BC          ;Decrement counter
3F90  78                  LD A,B
3F91  B1                  OR C
3F92  C8                  RET Z                 ;Strings are equal
3F93  18 F5               JR COMP5              ;Continue until BC = 0
                  ;
                  ;
                  ;
3F95  EF          LEN:    RST JT
3F96  B9                  DB 0B9H
                  ;CALL FIND1$
3F97  ED 53 FA81          LD (CALCST),DE
3F9B  C3 1348             JP STBC
                  ;
                  ;
                  ;
3F9E  EF          VAL:    RST JT
3F9F  B9                  DB 0B9H
                  ;CALL FIND1$
3FA0  36 FF               LD (HL),0FFH
3FA2  EB                  EX DE,HL
3FA3  22 FA81             LD (CALCST),HL
3FA6  E5                  PUSH HL
3FA7  CD 38BB             CALL LEX
3FAA  E1                  POP HL
3FAB  E5                  PUSH HL
3FAC  CD 399E             CALL ONESP
3FAF  D1                  POP DE
3FB0  CD 124F             CALL TOKMIN
3FB3  F5                  PUSH AF
3FB4  D5                  PUSH DE
3FB5  EB                  EX DE,HL
3FB6  CD 3B9E             CALL NUMIN1
3FB9  D1                  POP DE
3FBA  C1                  POP BC
3FBB  F5                  PUSH AF
3FBC  C5                  PUSH BC
3FBD  CD 224C             CALL MOVEACC
3FC0  F1                  POP AF
3FC1  D4 0FEA             CALL NC,COMPACC
```

```
3FC4  F1                    POP AF
3FC5  C9                    RET
                      ;
                      ;
                      ;
3FC6  EF              STR$:  RST JT
3FC7  81                     DB 81H
                      ;CALL ACCSTACK
3FC8  CD 20FD                CALL BINTOASC
                      ;
                      ;This puts the string pointed to by DE onto
                      ;the calculator stack ended in a 0FFH,this is
                      ;not put onto the stack.
                      ;
3FCB  EB              STRSTACK:EX DE,HL
3FCC  CD 1B43                CALL CALLEN
3FCF  0B                     DEC BC ;Includes 0FFh
3FD0  EB                     EX DE,HL ;DE at number
3FD1  2A FA81                LD HL,(CALCST)
3FD4  78                     LD A,B
3FD5  B1                     OR C
3FD6  28 06                  JR Z,STR1
3FD8  EB                     EX DE,HL
3FD9  C5                     PUSH BC
3FDA  ED B0                  LDIR
3FDC  C1                     POP BC
3FDD  EB                     EX DE,HL
3FDE  C3 315E         STR1:  JP STKLN1
                      ;
                      ;
                      ;
3FE1  CD 3E9C         FIND2$: CALL GETST
3FE4  D5                     PUSH DE
3FE5  EF                     RST JT
3FE6  B9                     DB 0B9H
                      ;CALL FIND1$
3FE7  D1                     POP DE
3FE8  C9                     RET
                      ;
3FE9  2A FA81         FIND1$: LD HL,(CALCST)
3FEC  2B              FIND$:  DEC HL
3FED  46                     LD B,(HL)
3FEE  2B                     DEC HL
3FEF  4E                     LD C,(HL)
3FF0  E5                     PUSH HL
3FF1  A7                     AND A
3FF2  ED 42                  SBC HL,BC
3FF4  D1                     POP DE
3FF5  EB                     EX DE,HL
3FF6  78                     LD A,B
3FF7  B1                     OR C
3FF8  C9                     RET            ;DE=Address of first string.
                      ;
3FF9  EF              GETPAR: RST JT
3FFA  9E                     DB 9EH
```

```
                        ;CALL LARGE1
3FFB  20 01             GP1:    JR NZ,ERROR$
3FFD  D0                        RET NC
                        ;
3FFE  EF                ERROR$: RST ERRORS
3FFF  22                        DB 34                   ;"NUMBER TOO BIG"
                        ;
                        ;
                        ;
                        END
```

Macros:

Symbols:

ABS	0E56	ACC1	FDCC	ACCSTA	128F	ADD	0E86
ADDNEG	2415	ADDPOI	2198	ADJDES	35F9I	ADJVAL	0C4F
ADLABE	FDBE	ADOP	3BD8	ADOP1	3BDF	ADX	3590
AE	3C45	AE0	3C46	AE1	3C4D	AE2	3C50
ALIST	1FA5	ALOP1	13F3	ALP	2A00	ALPH2	3B83
ALPHA	3B87	ALPHA1	3B88	ALPHEN	3B91	ANDACC	209A
ANDT	20A1	APROG	FDB7	ARANG	2413	ARITHO	3BE1
AROUND	21BC	ARRERR	0027	ARRTOP	FACC	ARRVAR	325C
ASBYTE	FDA8	ASC	3F4B	ASC1	3F57	ASCCPR	37D9
ASCINO	0030	ASCMAP	37C3	ASCMAT	37E0	ASCTES	37D0
ASTACK	FD6D	ATN	241D	ATNT1	2438	ATNT2	2452
AUTOCT	FD7B	AUTOIN	FD77	AUTOST	FD79	BADSUB	33A3
BASBOT	FAAA	BASE	3721	BASELI	FACF	BASEM	FDC7
BASENT	025D	BASIC	0259	BASICO	0244	BASIC1	0244
BASIC2	0250	BASIC4	024A	BASICS	0205	BASLNP	FAD1
BASTOP	FAA7	BASTPO	FAAC	BASTP6	FAA9	BAUDST	0D3C
BCDEC	0DD0	BEGIN	1C33	BELL	0953	BIGSUB	33D8
BIND1	0DE4	BINTOA	20FD	BLOCK	35C9I	BLOCK1	35D6
BOO	3D54	BOOO	3D5B	BOO2	3D64	BOO3	3D65
BOO4	3D6D	BOOBOT	00DC	BOOLEA	3D53	BOOPR	3D70
BOOTOP	00DA	BRANCH	2FB1	BREAIO	0CF3	BREAK	FD81
BREAKK	09B0	BREAKM	09F2	BRLOOP	2FB8	BSLOAD	3A6F
BSSTR	FE3F	BSTPHL	0679	C121	3B02	C12END	3B0F
C12SE	3B0D	C91	3AC8	C92	3AD4	CALCBO	FA7F
CALCST	FA81	CALCX	23DF	CALLEN	1B43	CAP3	3981
CAPITA	3980	CAR1	3D12	CARAC	3D0B	CASBAU	FD5F
CCCOMP	2FA1	CH10	36BD	CH11	36C7	CH12	36CF
CH13	36D0	CH14	36D2	CH15	36D9	CHAN	FE14
CHAR	326C	CHARO	327D	CHAR1	328B	CHECK1	36BC
CHECK2	36E7	CHKS30	3296	CHKST1	32A1	CHKSTK	3299
CHOUT	1B6A	CHPTR	FF59	CHR$	0A3A	CHRO	0A43
CHR1	0A41	CHRDS	153A	CLO1	3A2C	CLOERR	3A31
CL2ENT	3A3F	CL8ERR	3AB5	CLA881	3A9C	CLAS10	3AD9
CLAS11	3AE5	CLAS12	3AF4	CLAS13	3B13	CLAS14	3B23
CLAS15	3B33	CLAS16	3B3D	CLAS17	3B47	CLAS33	3A48
CLAS34	3A58	CLAS41	3A74	CLAS42	3A6D	CLAS81	3AA9
CLAS90	3AC1	CLASSO	3A26	CLASS1	3A33	CLASS2	3A3C
CLASS3	3A45	CLASS4	3A66	CLASS5	3A81	CLASS6	3A89
CLASS7	3A94	CLASS8	3A98	CLASS9	3AB7	CLOCK	FD57
CLRSTK	2F70	CLS	0008	CNT	0020	CNTRL	3636
CODE	00C2	CODETE	1BE3	COMAD	FDBC	COMMAN	FD82
COMMEN	FDBB	COMP	00F4	COMP$	3F5B	COMP1	3F73
COMP1P	3F74	COMP2	3F76	COMP3	3F7A	COMP4	3F88
COMP5	3F8A	COMPAC	0FEA	COMPAO	115D	COMPAR	1163
COMPBC	11EC	COMPST	1BC4	CONOK	27C0	CONOUT	0002
CONT	3E25	CONTAD	FD6A	CONTFL	FD69	CONTON	2B85
CONTPG	FD6C	CONTRY	37E6	CONTST	27A3	COPY	FDDE
COPYAC	12F3	COPYDO	04B0	CORECS	0EE3	COSINE	137D
CPVAL	2F38	CR	000D	CRLF	001B	CRNTPG	FAD3
CTABLE	3D1D	CTRBAD	FA52	CTYSLT	FB45	CURLAB	FDC9
CURSCR	FF5B	DADD	FDA6	DATA	008F	DATAAD	FB46
DATALA	FDC3	DATAPG	FB48	DBLABE	FDC6	DBYTE	FDAA
DDBCM	05D1	DEB1	3701	DEB2	3709	DECBIN	2309
DECFRA	3BB0	DECPG	06ED	DEHL	0008	DEL	39E1

DEL1	39E4	DEL2	39E7	DELBOT	00C8	DELET1	06CF
DELETE	06C7	DELSPR	FE52	DELTOP	00CE	DESAVE	FB49
DESTAC	1274	DETPN	06E7	DETYPE	FDAF	DF1	3BB6
DIGIT	3B96	DIGIT1	3B97	DIM1	34C1	DIMERR	332F
DIMLP	3440	DIR1	39EE	DIR111	2EA2	DIR2	39F4
DIR3	39F6	DIRECT	39DE	DIRIO	0006	DIROUT	08EB
DIRRRR	2E9A	DIRTRY	2E9D	DISAD	FDB1	DISP1	2F75
DIV10	0E3D	DIVACC	21ED	DIVBY1	0E2B	DIVIDE	109E
DIVOP	22CE	DJVL4	0C9D	DJVLA	3611	DJVLB	3615
DLSPND	FE4F	DODIM	331F	DOJ1	3E3B	DOJMP1	3E33
DOJUMP	3E38	DOLAR	2267	DOLAST	2031	DOLONG	2376
DON	2A7B	DONEXT	2AAD	DONIB	0FBE	DONS5	334C
DONSIZ	3350	DOSIGN	203A	DOWNST	3E2B	DOZE	336F
DPROG	FDB3	DR	0005	DTYPE	FDB0	EDEND	1448
EDENT	02D9	EDIT	FDAD	EDIT1	1424	EDITOR	1429
EFFENT	02CB	EL1	3A1A	ELSE	00C3	EMB	0080
END1	35F4	ENDAR	00FF	ENDAUT	3716	ENDC3	3A62
ENDCAP	398A	ENDFUN	277D	END61	24ED	END6ET	24EA
ENDLI1	2CFF	ENDLIN	2CFEI	ENDLLL	3A16	ENDMAR	00FF
ENDMID	3F15	ENDSCA	36F3	ENDTAB	FDB9	ENDVAR	29ED
ENGLAN	37F0I	ENTER3	1C03	ENTERI	1C11	ENUMBE	2499
EQUAL	20C0	EREPBI	22E8	ERR34	197B	ERROR	0028
ERROR$	3FFE	ERROR1	2F99	ERROR2	3005	ERRORR	0028
ERRORS	0028	ERRPOS	FD84	ERRRST	0028	ESHIFT	2245
EV	3051	EV0	3D89	EV01	3D91	EV1	3D92
EV11	3D98	EV2	3DA0	EV3	3DA4	EV4	3DB1
EV5	3DB8	EV6	3DBF	EV65	3DC3	EV7	3DE3
EV8	3DEE	EVAL1	3D87	EVAL2	3D88	EVALAB	3DB4
EVALFN	3E5B	EVALSE	3E7E	EVFN0	3E60	EVFN1	3E63
EVFN3	3E6E	EXCHAN	1183	EXP	23EC	EXP01	1372
EXP2X	23F2	EXPON	3BA4	EXXXX	2F03	FAST	3712
FE	29DF	FEXPAN	FA9E	FFFND	2F31	FFFOUN	395C
FINIMU	22D3	FINASC	211F	FIND	19DD	FIND$	3FEC
FIND1$	3FE9	FIND2$	3FE1	FINDAD	39F8	FINDDA	2C2C
FINDIN	2F1A	FINDIT	301C	FINDJP	0689	FINDLE	316E
FINDLI	061A	FINDNO	2E39I	FINDNS	1A19	FINDVA	3171
FINFF	0BFE	FINMUL	22D5	FINSIG	22D6	FIRST	3045
FIXED	3BBB	FLAGS1	FD86	FLOOP	301E	FN1	2E3E
FND1ST	2C4B	FNDEND	2C6B	FNDERR	3177	FNDINI	313A
FNDLP	2C4E	FNDNXT	2C45	FNDTOP	2E5A	FNDTPX	2E64
FORCOU	FB44	FOUG2	2B7D	FOUND	3179	FOUND0	396C
FOUND1	396E	FOUND2	3976	FOUNDA	3969	FOUNDG	2B6C
FOUNEN	397C	FRAC	112B	FRANCE	37F1I	FREESP	033C
FREQ	FE16	FRONT	0030	FULL	0005	FUNC	3DEB
GASH	FE02	GERMAN	382CI	GET5VA	351A	GETBAS	0BC5
GETCHA	2FF3I	GETDE1	2350	GETDEC	22DF	GETDIM	3464
GETINP	2927	GETLEN	34AD	GETLIN	06DB	GETLN1	06DE
GETLNE	19E5	GETLP	3521	GETNEX	24CC	GETNVA	351D
GETPAR	3FF9	GETRET	3531	GETRST	0030.	GETSPA	2FE9I
GETST	3E9C	GETSTA	2FC5	GETSTR	00BC	GETSUB	3009
GETVA1	307C	GETVAL	3066	GGETST	2945	GG6OTO	2EE9
GN1	24E2	GOLABE	2EF7	GOPTR	FB41	GOSERR	28C1
GOSNUM	FB43	GOSTAC	FAD8	GOSUB	0097	GOSUB2	28B1
GOTMIN	20B7	GOTO	0096	GOTO1	2896	GOTP	21B9
GOTZER	10A6	GP1	3FFB	GRE1	20CD	GREATE	20C8
GREEQU	20DB	GRRDS	1541	GST5R	34DF	GST5VA	3505

GSTNSR	34E2	GSTNVA	3508	HIGH	0A8C	ID1	3B4F
IDENT	3BF6	IDENT0	3BF5	IDENT1	3C0C	IDENT2	3C0D
IDLIST	3B4E	IF1	296E	IF2	2977	IFTHEN	39EA
IJINIT	097F	IJTABL	FFF0	IMOVE	1C73	INBIT	0A6B
INBLK2	0ACB	INBLOC	0ABA	INBYTE	0A9A	INCL1	35C0
INCLR1	35AD	INCLRA	35AC	INCPG	06F3	INDEX	FDA8
INDEXL	FDC1	INKEY	2F61	INKEY$	3ECE	INOUT	0AAE
INPUT	00FF	INPUTC	28FE	INPUTD	290D	INPUTE	2917
INSERT	02FE	INSTR	1B85	INT	200A	INT125	0996
INT64	3D7D	INT64K	121D	INTE6	3BCE	INTE61	3BD1
INTEGE	119D	INTFFF	FD5E	INTTAB	FDF2	IOPL	FABF
IOPR	FD76	ISARR	30B8	ISCHR	3099	ITYPE	FD87
JPHL	07B8	JT	0028	K	0400	KB8	365B
KBCHK	28D2	KBD	0079	KBDBUF	FA83	KBDP0	3618
KBFLAG	FA91	KBN999	3664	KBNORM	3643	KEY1	3622I
KEYPOR	0005	KILLIN	098A	KILLVA	27FBI	KILSND	093A
LABTAB	FDB5	LARGE1	11C5	LARGEN	11C8	LASTAS	FD7D
LASTDR	FD7E	LASTKY	FD7C	LDBSBT	06AC	LEFT$	3F20
LEFTRI	3F27	LEN	3F95	LENGTH	FDAE	LENHI	FF57
LENLO	FF56	LESEQU	20E2	LESS1	20D7	LESSTH	20D2
LET1	2A29	LET2	2A2B	LET3	2A2C	LEX	38BB
LEX1	3935	LEX2	3936	LEX3	3941	LEX7	394D
LEX8	394F	LEX9	3957	LEXFUN	2680	LF	000A
LINKER	FDAC	LIST1	2992	LISTEN	29AC	LLIST	297E
LNUMBE	2461	LOAD	0B40	LOAD2	0B9E	LOADED	2D53
LOADER	0891	LOADSN	2D0CI	LOG	2398	LOG12	23CF
LOG2E	248F	LOG2X	23A5	LOGE2	2494	LOGNX	23BC
LONGES	3F86	LOOK	3718	LOOK1S	30E2	LOOP	2D12
LOW	0AB2	LP	3042	LR1	3F36	LRATOV	3593
LSTPG	FA7A	MAF	FD91	MAFD	FD89	MAINLP	2ECD
MAKESP	0351	MATHSR	0038	MBC	FD93	MBCD	FD8B
MDE	FD95	MDED	FD8D	MDELET	21D2	MEM1	FDD9
MEMPOI	FDA1	MESS	0007	MFBOT	00F4	MFTOP	00DD
MFUNC	2688	MHL	FD97	MHLD	FD8F	MID$	3EDD
MID$ER	3F18	MID1	3EF6	MID11	3EF7	MID2	3EFC
MID3	3F0A	MID4	3F0D	MID5	3F11	MIDVAL	FD60
MIX	FD99	MIY	FD9B	MORE	2275	MORE0	2174
MORE10	2294	MOREDA	2C58	MOREHE	297E	MORES0	2024
MORROU	21C3	MOVDE0	12B9	MOVE1	217B	MOVEAC	224C
MOVED	30CF	MOVEDE	12A7	MOVHL	1BDD	MPC	FD9F
MSP	FD9D	MSSS	2D2B	MUL10	222E	MULACC	220A
MULLP	330C	MULPLY	100B	MULT16	3307	MULTER	331D
MVDIST	FE4D	MVN0	FE51	NA	2A1E	NAMELE	3152
NAND	00DA	NAR	3062	NBTOP	FAA4	NBTPG	FAA6
NCONT	3684	NCP1	06F7	ND	3059	NDBS1	2E7D
NDDDD	29F8	NDISPL	2F73	NDSP1	2E23	NDSP2	2E2B
NEEDEX	2142	NEEDP0	216C	NEGACC	0F13	NEWENT	0221
NEWERR	320A	NEWLIN	2BC9I	NEWS0	31F2	NEWS1	3202
NEWS2	320E	NEWSC1	2F91	NEWSCR	2F90I	NEWSTR	31C4
NEWSX	31F8	NEWVAR	30EE	NEXCH	38FB	NEXCH0	3929
NEXCH1	390A	NEXCH4	3917	NEXCH5	391C	NEXFOU	3914
NEXON1	2B60	NEXPR	3E84	NEXT	39CE	NEXTON	2B5D
NEXTST	2AC9	NEXWIO	2A60	NF1	38F8	NLB	2A0D
NNOT	00DC	NO	20C5	NO10	22B4	NO1EXX	2226
NOADD	3317	NOADJ	3591	NOAF	230F	NOCH	3948
NODBAS	2E67	NODD2	2D83	NODD33	2DA0	NODD4	2DB7

NODD5	2DBD	NODD55	2DDD	NODD6	2DE1	NODDSP	2E0E
NODDY	2D7CI	NODENT	2EB8	NODER2	3034	NODERR	302B
NODLIS	2ED9	NODLOC	FA9B	NOGOOD	02C0	NOMINU	2186
NOMORE	22AA	NONAU	308D	NONZER	2386	NOOVER	21CC
NOPAR	2A73	NOPARA	3E73	NOPOIN	2282	NOR	00DB
NORM1	3685	NORM11	3689	NORM2	3693	NORM3	369F
NORM4	36A4	NORMAL	2823	NOROOM	32D8	NOSPAC	32AB
NOSPR	FE4E	NOT	00DC	NOT10N	22B0	NOTACC	20AF
NOTARR	3193	NOTCOL	3A24	NOTEQU	20E9	NOTFND	3149
NOTHIN	217F	NOTNUM	2937	NOTPRE	3331	NPAR	2A3C
NQT	2A17	NSH1	3677	NSH2	367E	NSHIFT	367A
NTLAST	35A4	NULL$	3EB9	NUMARR	328D	NUMBER	3B9F
NUMELS	343D	NUMEND	3097	NUMI2	345D	NUMIN1	3B9E
NUMIND	345B	NUML1	3B63	NUML2	3B70	NUMLIS	3B5C
NXTND	2E88	OFF	2F57	OFREQ	08C6	OL1	24F3
OL2	24FA	OL3	2510	OLDQT	325A	OLDST1	3257
OLDSTR	322B	ONELIN	29AFI	ONESP	399E	ONESP0	39A8
ONESP1	39B0	ONESP2	39BC	ONESP3	39BE	ONESPA	399F
OP	3E07	OP11	FDD2	OP1STA	129E	OPBASE	00CF
OPBOT	00D9	OPCON	3BF1	OPS	2672	OPST	3DF9
OPTABL	0D72	OPTOP	00CF	ORACC	20A6	OSPRST	0018
OTBLK2	0AFC	OUT2	39C9	OUTBIT	0A7B	OUTBLO	0AE6
OUTBYT	0AA4	OUTFUN	2514	OUTR	27BE	OUTRAN	1219
OUTRS	2519	OUTSP	39C1	OUTSTR	1907	OUTWOR	252D
OVERAL	38F0	OVERFL	121B	OVERHL	3B73	OVERSP	3B74
OVOL	08F9	OVRLAY	FFED	P1JTBL	2E6B	PA1	3CE3
PA2	3CE9	PA3	3CF5	PA4	3D03	PAGE	FAD2
PAR0	3E4E	PAR1	3E4C	PAR2	3E44	PARAMS	3CDC
PAREND	3CCA	PARERR	3D09	PARRRR	3E40	PARSSS	3E3C
PAU0	2BE6	PAU1	2BE8	PAU2	2BEF	PAU3	2BF6
PAUSE	2F13	PGCHK	0671	PGCHK1	066E	PGN1	FAD4
PGN2	FAD5	PGPORT	0000	PGSLC1	0647	PGSLC2	0644
PGSLCT	064A	PGTOP	FAD6	PIOP1	120B	PLSPN0	FE50
PLUS1	33F5	POELOP	2356	POINTE	FDA5	POLY2	13C1
POLYNO	13D2	POPVA	3483	POPVA1	3486	POSI	2156
POWER	233F	PREND	2BDE	PRIMAR	3C60	PRIMBO	3C7B
PRIMFU	3C80	PRIMPA	3C88	PRIMVA	3C6C	PRINTX	0CAB
PRINTZ	20F0	PRNNUM	2BB1	PRNTST	2BCD	PRORPL	FD75
PSHNAM	3037	PSHVA	3493	PST5SR	34EE	PST5VA	3534
PSTNSR	34F1	PSTNVA	3537	PUT5VA	3544	PUTLP	354B
PUTNVA	3547	PUTS1	30C6	PUTST1	2FE2	PUTSTA	2FCDI
PUTSTR	30BD	QUITNZ	35D3	R2	2C0C	RAMP6C	34FE
RAND1	2BD7I	RCALC1	349D	RCALCS	316A	READ2	2C7C
READDE	231C	READER	2CB8	REALBY	FA90	RECEP	2332
RELAT	3D2E	RELBOO	3D3A	RELNUM	3D4A	RELOP	3BEA
REM	00B0	REM0	2034	REMAIN	0FFA	REMHL	2195
REMZER	21AD	REP1BI	2303	REPASC	2105	REPBIN	22E9
REPDEC	2316	RESINT	0B06	REST1	2C9F	RESTEN	2C8DI
RET1	2C1A	RET11	2C1C	RETACC	12D1	RETERR	2C2A
RETSAV	FD61	RETT	07B5	RETURN	2FBD	RICHJL	FD4B
RIGHT$	3F1A	RMTPHL	0697	RND	2048	RND1	2059
RND2	207B	RNGERR	0024	RNSEED	FD7F	ROAHL	0FA1
ROMBIT	0000	ROOM	32EE	ROTR	0FBD	ROUNAC	0F2B
ROUND	2189	RUN2	2CBF	RUN3	2CC1	RUN33	2CC0
RUN4	2CFDI	RUN5	2CF0I	RUN55	2CDBI	RUN6	2CDD
RUN8	2CECI	RUN9	2CF6	RWTAB	26F7	SAANGL	15FC

SADJSP	15BB	SARC	16CE	SASSEM	2784	SATTR	15E2
SAUTO	0A07	SAVE	0B90	SAVEA	21E2	SAVEAC	12BF
SAVEP	2364	SAVEPA	3577	SBAUD	0D3B	SCAN	36AA
SCAN2	36AF	SCAN3	36DF	SCAN4	36EE	SCENT	1876
SCENT4	188F	SCENTR	00CB	SCIRCL	180F	SCLEAR	27FB
SCLS	1595	SCODE	00DA	SCOLOU	15B3	SCONT	2787
SCR	0040	SCRN0	FF5D	SCRN1	FF6C	SCRN2	FF7B
SCRN3	FF8A	SCRN4	FF99	SCRN5	FFA8	SCRN6	FFB7
SCRN7	FFC6	SCRRST	0010	SCRVS	15EA	SCTLSP	159B
SCURSO	1583	SDATA	27AA	SDIM	2837	SDRAW	1687
SDSI	157C	SE	3C91	SE2	3C92	SEARCH	30D5
SEDIT	2846	SEDITO	28C5	SEFUN	3EC6	SENSE1	0005
SENSE2	0006	SEQ0	3EA1	SEQ1	3EA4	SEQ2	3EB0
SEQEND	3EB8	SET1AC	2000	SETCAL	FD48	SETINI	27B7I
SETINT	0B10	SETZER	10A9	SEVAR	3EBC	SFBOT	00FD
SFOR	2867I	SFTOP	00F5	SGENPA	15A3	SGOSU3	28A6
SGOSUB	289CI	SGOTO	288FI	SGOTO3	2894	SHIF	00AD
SHIFT4	2211	SHIFTA	2246	SIF	2961	SIGN	FDD8
SIGNAC	208F	SINCOS	161F	SINE	1382	SINED	28FA
SINEE	28FB	SINK	1590	SINP0	28E5	SINP1	28F3
SINP11	28EF	SINP2	28F6	SINPUT	28DC	SINTEG	3BCB
SLET	29DA	SLINE	1574	SLIST	297D	SLLIST	2941
SLOAD	2AEE	SLOOP	27E1	SLPRIN	2A37	SMALL1	11B9
SMALLE	11BB	SMBYTE	FF55	SMVSPR	15D1	SNDBUF	27B3I
SNDERO	2D69	SNDPTI	0003	SNDPTO	0006	SNEW	0205
SNEXT	2A64I	SNODDY	2B44	SNODE	277F	SON	2B4F
SON1	2B54	SOUT	09FD	SP0	2B93	SP1	2B9D
SP2	2BA1	SPAIN	3890I	SPANEL	2BF8	SPAPER	158A
SPAUSE	2BE4	SPBASE	FE4C	SPCERR	0023	SPEED	FE4B
SPLOD	2B32	SPLOD1	2B25	SPLOT	1670	SPOKE	2C02
SPR	2BB8	SPR1	2B89	SPRI2	2B8C	SPRIM	3CA0
SPRIMF	3CD2	SPRIML	3CA8	SPRIMP	3CC2	SPRIMV	3CB5
SPRINT	2B88I	SPRLIT	3CA7	SPRTBL	FE55	SQROOT	236E
SQUA	238E	SQUA1	2391	SQUARE	1008	SQUASH	0A15
SR1	294C	SRAMPG	34FF	SRAND	2C09	SRANGL	160A
SREAD	2C74	SREM	27AA	SREND	295B	SRESTO	2C8C
SRETUR	2C11	SROM	009E	SRPTRS	2825	SRUN	2CAFI
SSAVE	2B03	SSCLOC	2AD6	SSOUND	2D0CI	SSPRIT	15C4
SSS	361B	SSSUBS	3C20	SSTACK	FA96	SSTOP	2D55
STA	1311	STAC	1317	STACCT	FD73	STACK	FD48
STACKA	1286	STACKE	300F	STACKO	1298	STACKS	0744
STAHLH	0F58	START	3622	STASEM	1CC3	STBC	1348
STEP	00C6	STEP1	2876	STHLHL	136B	STKDIM	34B7
STKLEN	315B	STKLIM	FA92	STKLN1	315E	STKNOD	2F0A
STKTES	3EC9	STKTOP	FA7A	STOP	2D62I	STOPPP	2D56I
STQU	228D	STR	3CF3	STR$	3FC6	STR1	3FDE
STRHL	02E7	STRING	26CD	STRJP	0DBE	STRNG	0009
STROUT	02E8	STRSTA	3FCB	STRVAR	31A4	SUBCH0	3411
SUBCH1	3415	SUBCHK	3412	SUBEN1	341F	SUBEND	341B
SUBERR	31DB	SUBLP	33B2	SUBRET	342F	SUBS	30B5
SUBS19	3C2F	SUBS2	3C31	SUBSCR	33A5	SUBSTR	3435
SUBTRA	0E83	SVERIF	2B1A	SVS	15F6	SWEDIS	3863I
SWINDO	15DA	SWITCH	00F4	SYFAIL	3C40	SYFL	3BA2
SYFLA	3C01	SYFLB	3BFA	SYFLC	3C96	SYFLD	3D61
SYN	398C	SYN1	39D1	SYNPRO	14B9	SYNRST	0020
SYNT	3A0B	SYNT1	3A04	SYNTA	3A60	SYNTB	3A87

SYNTC	3AF0	SYSCA	3C5E	SYSCB	3CB3	SYSCC	3D42
SYSTOP	FA94	SYSUC1	3C37	SYSUCC	3C38	TAB	0009
TEMP	FE04	TESDIV	2114	TEST0	32D2	TEST1	1A0C
TEST2B	2A8D	TESTBI	2FFE	TESTHL	1BB6	TESTNU	2327
TESTSP	0344	THEN	00C7	TIME$	3F38	TINSER	2A7A
TL1	38C1	TL2	38CD	TL3	38C6	TL4	38E6
TMPA	FD71	TMPHL	FD6F	TO	00C8	TOKMIN	124F
TOP	3E15	TOP1	3E1B	TOPCH	2D6B	TOPMUL	22BC
TOPN	2A46	TOPNOD	2E53	TPCH1	06BB	TSSTSP	32AD
TTTOP	39C4	TYPE	FD68	TYPTBL	FFD5	UNDEFI	0030
UNKERR	0026	UNSTEP	2879	UPDATE	3182	UPPER	3772
USER	FA89	USERIN	FA9B	USERIO	FD51	USERNO	FAA1
USERRO	FD54	USR	203D	USRRST	FD4E	UVITOS	34A0
VAL	3F9E	VALBOT	FA7D	VALID	36D6	VALRA1	35B0I
VARNAM	FA7B	VATOLR	357CI	VAZERO	FD65	VCOUNT	FE53
VDINIT	2E85	VDPR60	0002	VDPR61	0002	VDPSTS	FE54
VERIF	FD67	VINT	3A55	VINTF6	FF58	VJ1	2F42
VJUMP	2F43	VOL	FE18	WCHJUM	FDA3	WDTOP	00C2
WKAREA	FE1A	WORDLS	253B	YES	20CF	YORN	FDD7
ZERACC	0FAF	ZEROLP	3561	ZERONV	355D	ZLP	3123

No Fatal error(s)