```
2427  20 EA          JR NZ,NEWMEM
2429  E1             POP HL
242A  7D             LD A,L
242B  E6 07          AND 7
242D  6F             LD L,A
242E  87             ADD A,A
242F  85             ADD A,L
2430  C6 05          ADD A,5
2432  4F             LD C,A
2433  06 02          LD B,2
2435  D7             RST SCRRST
2436  65             DB SDISP+CONTINUE+40H
2437  23             DB 3+CONTINUE
2438  E0             DB 0C0H+CONTINUE
2439  81 3E          DB 80H+1,'>'
243B  C9             RET
                     ;
                     ;
                     ;
                     ;*** DISAM.INC ****
                     ;
                     ;
                     ;this dissasembles the bytes pointed
                     ;to by DE to the screen
                     ;DE points at new start A has the
                     ;number of bytes read.
                     ;
243C  AF             NOTPC:  XOR A
243D  32 FDBB                LD (COMMENT),A
2440  32 FDB0                LD (DTYPE),A
2443  32 FDB7                LD (ITYPE),A
2446  CD 1B50                CALL OUTVBC
2449  C5                    PUSH BC
244A  50                    LD D,B
244B  59                    LD E,C
244C  D7                    RST SCRRST
244D  81                    DB 80H+1
244E  20                    DB ' '
244F  CD 2BBA                CALL RELDE      ;Real address value
2452  CD 29B6                CALL SRCADD ;Look for labels
2455  06 08                 LD B,8
2457  20 0F                 JR NZ,NOADLAB
2459  7E                    LD A,(HL)
245A  FE 03                 CP 3  ;Comment or defb
245C  30 0A                 JR NC,NOADLAB
245E  CD 29F4                CALL MOVESTR
2461  CD 1A70                CALL COUNTHL
2464  05                    DEC B
2465  D7                    RST SCRRST
2466  81 3A                 DB 80H+1,':'
2468  05             NOADLAB:        DEC B
2469  FA 2471                JP M,FINTAB
246C  D7                    RST SCRRST
246D  81                    DB 80H+1
246E  20                    DB ' '
246F  18 F7                 JR NOADLAB
```

```
2471  2A FDC9          FINTAB: LD HL,(CURLAB) ;Get position
2474  7E                       LD A,(HL)
2475  FE FF                    CP 0FFH ;End
2477  28 09                    JR Z,DODIS
2479  FE 03            ONEDI:  CP 3
247B  30 20                    JR NC,PRDEFB
247D  CD 29BA            LOPPP:  CALL LOOPSRC ;Get next one
2480  28 F7                    JR Z,ONEDI
2482  04               DODIS:  INC B
2483  05                       DEC B
2484  20 08                    JR NZ,NORETD
2486  79                       LD A,C
2487  E1                       POP HL
2488  06 00                    LD B,0
248A  09                       ADD HL,BC
248B  44                       LD B,H
248C  4D                       LD C,L
248D  C9                       RET
248E  C1               NORETD: POP BC
248F  C5                       PUSH BC
2490  D9                       EXX
2491  CD 24EB                  CALL DISAM
2494  D9                       EXX
2495  E1                       POP HL
2496  A7                       AND A
2497  ED 42                    SBC HL,BC
2499  7D                       LD A,L
249A  ED 44                    NEG
249C  C9                       RET
                       ;
                       ;This checks for a comment or defb
                       ;if a comment then flags (COMMENT)
                       ;if a DEFB then B=0 and C the length
                       ;
249D  FE 08            PRDEFB: CP 8  ;Comment
249F  38 08                    JR C,ISDFG
24A1  32 FDBB                  LD (COMMENT),A ;NZ
24A4  22 FDBC                  LD (COMAD),HL
24A7  18 D4                    JR LOPPP
24A9  7E               ISDFG:  LD A,(HL)
24AA  23                       INC HL
24AB  23                       INC HL ;Address
24AC  23                       INC HL
24AD  4E                       LD C,(HL) ;Length
24AE  06 00                    LD B,0
24B0  23                       INC HL
24B1  23                       INC HL
24B2  C5                       PUSH BC ;Save BC
                       ;       PUSH AF ;Type
24B3  D7                       RST SCRRST
24B4  81 44                    DB 81H,'D'
                       ;       POP AF
24B6  D6 04                    SUB 4
24B8  3E 57                    LD A,'W'
24BA  28 08                    JR Z,GOTW1
24BC  3E 42                    LD A,'B'
```

```
24BE  38 06                   JR C,GOTW
24C0  3E 53                   LD A,'S'
24C2  0E 02                   LD C,2
24C4  CB 39             GOTW1: SRL C
24C6  EF                GOTW:  RST JT
24C7  AC                       DB 0ACH
                        ;CALL PRINTX
24C8  D7                       RST SCRRST
24C9  81 20                    DB 81H,' '
24CB  CD 24DC                  CALL OUTDEF
24CE  0D                PR3:   DEC C
24CF  28 08                    JR Z,PRDL
24D1  D7                       RST SCRRST
24D2  81 2C                    DB 81H,','
24D4  CD 24DC                  CALL OUTDEF
24D7  18 F5                    JR PR3
24D9  C1                PRDL:  POP BC
24DA  18 A1                    JR LOPPP
                        ;This outputs defb strings
24DC  7E                OUTDEF: LD A,(HL)
24DD  FE 22                    CP '"'
24DF  C2 1907                  JP NZ,OUTSTR
24E2  41                       LD B,C
24E3  CD 1A70                  CALL COUNTHL
24E6  48                       LD C,B
24E7  0C                       INC C
24E8  0C                       INC C
24E9  0C                       INC C
24EA  C9                       RET
                        ;
                        ;
                        ;B WILL CONTAIN G,C WILL HAVE
                        ;THE VALUE OF H BC' HAS POINTER
                        ;TO THE BYTE,HL' HAS POINTER
                        ;TO THE DATA,D' HAS INDEX
                        ;E' HAS CLASS D HAS BYTE
                        ;TO DISEMBLE HL HAS NO SPECIFIC USE
                        ;A AND A' ALSO ARE USED FOR UTILITY
                        ;E HAS THE DATA BYTE THIS IS ROTATED
                        ;SO THE ROUTINE CALLED IS LOST AND THE
                        ;DATA IS MORE ACCESSIBLE
                        ;E IS SET UP WITH THE DATA IN THE LEAST
                        ;SIG 3 BYTES,THE ROUTINE NEEDED IN NEXT 3
                        ;BIT 7 IS SET FOR END OF DATA
                        ;
                        ;CONVERT THIS CONVERTS TH BYTE
                        ;IN D TO BC F GOES TO E
                        ;
                        ;
                        ;THIS IS THE START OF THE DISAMBELLY
                        ;ROUTINE EVERYTHING IS INITIALLISED
                        ;BC' POINTS TO THE BYTE TO DISASEMBLE
                        ;
24EB  21 254E           DISAM:  LD HL,MASTRADS
24EE  22 FDA3                   LD (WCHJUM),HL
24F1  D9                        EXX
```

```
24F2  11 0000          LD DE,0
24F5  0A               LD A,(BC);GET BYTE
24F6  03               INC BC
24F7  FE 76            CP 76H;IS IT HALT
24F9  20 07            JR NZ,NOTHALT
24FB  D9               EXX      ;Send it the length
24FC  21 2E67          LD HL,ADHALT
24FF  C3 1907          JP OUTSTR

2502  FE DD      NOTHALT:    CP 0DDH
2504  28 05                 JR Z,DDFD
2506  FE FD                 CP 0FDH
2508  20 0C                 JR NZ,NOTDDFD
250A  14                    INC D
250B  14               DDFD:  INC D
250C  0A                    LD A,(BC)
250D  03                    INC BC
250E  FE CB                 CP 0CBH
2510  20 10                 JR NZ,NORM
2512  03                    INC BC
2513  1C                    INC E
2514  0A                    LD A,(BC);BC POINTS 1
2515  0B                    DEC BC ;TO DATA
2516  FE CB          NOTDDFD:    CP 0CBH
2518  28 05                 JR Z,CBED
251A  FE ED                 CP 0EDH
251C  20 04                 JR NZ,NORM
251E  1C                    INC E
251F  1C               CBED:  INC E
2520  0A                    LD A,(BC)
2521  03                    INC BC
2522  D9               NORM:  EXX
2523  57                    LD D,A;SAVE BYTE
2524  E6 07                 AND 7
2526  4F                    LD C,A
2527  7A                    LD A,D
2528  1F                    RRA
2529  1F                    RRA
252A  1F                    RRA
252B  E6 07                 AND 7
252D  47                    LD B,A
252E  7A                    LD A,D
252F  07                    RLCA
2530  07                    RLCA
2531  E6 03                 AND 3
2533  5F                    LD E,A
2534  D9                    EXX
2535  7B                    LD A,E;GET CLASS
2536  D9                    EXX
2537  A7                    AND A
2538  17                    RLA
2539  17                    RLA
253A  B3                    OR E;AND IN F
253B  17                    RLA
253C  CD 2542               CALL CALTABLE
253F  C3 1B09               JP MASTER;GET DIS

                ;
```

Test for HALT and send 'HALT' if so

Test for index register opcodes.
D's 0 → HL and jump.
D's 1 → IX
D's 2 → IY

Get main op code

Jump if not but instr

E = 1
A = Sub code
BC = Index byte

E' = 0 → main code
E' = 1 → CB code
E' = 2 → ED code
A = code or sub code

D = opcode
C = Bits 0-2
B = Bits 3-5
E = Bits 6->

A = 8E' + 2E

- Set source pointer for code table

```
                        ;This returns the value of HL
                        ;for the pointer number in A
                        ;
2542  D9                CALTABLE:     EXX
2543  21 2BF2                         LD HL,TABBYT;POINTER FOR TABLE
2546  E6 1E                           AND 1EH
2548  1F                              RRA
2549  CD 0689                         CALL FINDJP
254C  D9                              EXX
254D  C9                              RET
                        ;
                        ;
                        ;THIS IS THE JUMP TABLE FOR
                        ;THE ROUTINES
                        ;
                        ;
254E  258F              MASTRADS:     DW SPLIT
2550  25B2                          DW LITERAL
2552  2575                          DW LISTG
2554  2578                          DW LISTH
2556  259A                          DW SELECG
2558  259D                          DW SELECH
255A  256F                          DW SKIP
255C  255E                          DW KSKIP
                        ;
                        ;
                        ;THIS ROTATES G SO THAT IT IS
                        ;EQUAL TO J THEN IF K WAS ONE
                        ;IT JUMPS
                        ;
                        ;
255E  CB 38             KSKIP:  SRL B;LET G=J
                        ;
                        ;
                        ;THIS SKIPS THE NUMBER OF BYTES
                        ;POINTED TO BY HL OR DEPENDING
                        ;ON ZERO FLAG IF IT IS SET IT
                        ;DOES NOT SKIP.FLAGS ARE UNAFFECTED
                        ;
                        ;
2560  9F                SKIPHC: SBC A,A
2561  F5                SKIPHL: PUSH AF
2562  D9                        EXX
2563  28 06                     JR Z,NOADD
2565  D5                        PUSH DE
2566  5E                        LD E,(HL)
2567  16 00                     LD D,0
2569  19                        ADD HL,DE
256A  D1                        POP DE
256B  23                NOADD:  INC HL
256C  D9                        EXX
256D  F1                        POP AF
256E  C9                        RET
                        ;
                        ;
                        ;THIS SKIPS IF G>3 IF IT IS IT
```

HL' set to address in TABBYT pointed to by bits 0-1 of A.

Test bit 1 & 0 of B

C → NZ

If NZ then

HL' = HL' + (HL') + 1

otherwise unchanged HL' incremented

```
                        ;JUMPS AND THEN RESETS BIT 2
                        ;SO G=G-4
                        ;
                        ;
256F  CB 50             SKIP:   BIT 2,B;IS G>3 IF SO SKIP
2571  CB 90                     RES 2,B;LET B=B-4
2573  18 EC                     JR SKIPHL
                        ;
                        ;THIS WILL LIST THE Gth STRING
                        ;AND THEN MOVE TO THE END OF THE
                        ;STRING OF BYTES
                        ;
                        ;
2575  68               LISTG:  LD L,B;PUT IN BYTE
2576  18 01                    JR LISTL
                        ;
                        ;
                        ;THIS IS THE SAME AS ABOVE
                        ;BUT H INSTEAD
                        ;
                        ;
2578  69               LISTH:  LD L,C
                        ;
                        ;
                        ;THIS LISTS THE Lth STRING
                        ;THEN MOVES TO END
                        ;ASSUMED TO BE IN E ONLY L AND A
                        ;ARE DEYSTROYED
                        ;
                        ;
2579  7D               LISTL:  LD A,L
257A  CD 2589                  CALL FINDEX;FIND THIS BYTE
257D  D5                       PUSH DE
257E  CB 83                    RES 0,E
2580  CD 25B2                  CALL LITERAL
2583  D1                       POP DE
2584  7B                       LD A,E
2585  E6 07                    AND 7
2587  95                       SUB L
2588  D8                       RET C;PAST END OF LIST
                        ;
                        ;THIS FINDS THE Ath STRING IN A SEQUENCE
                        ;A IS DEYSTROYED
                        ;
                        ;
2589  D9               FINDEX: EXX
258A  CD 19DD                  CALL FIND
258D  D9                       EXX
258E  C9                       RET
                        ;
                        ;
                        ;
                        ;THIS RECALCULATES HL' BY TAKING
                        ;THE FOLLOWING 8 BYTES AS DATA
```

```
                        ;FOR JUMPS THERE WILL BE
                        ;8 OF THESE.THE Hth ONE IS TAKEN
                        ;
                        ;
258F    79              SPLIT:  LD A,C
2590    E6 07           SPLIT1: AND 7
2592    3C              SSPLIT: INC A
2593    3D              TOPSPLIT:       DEC A
2594    CD 2561                 CALL SKIPHL
2597    20 FA                   JR NZ,TOPSPLIT
2599    C9                      RET
                        ;
                        ;
                        ;THIS SELECTS THE Lth ONE OF THE
                        ;REGISTERS THE LIST IS GIVEN BY
                        ;E
                        ;
259A    68              SELECG: LD L,B
259B    18 01                   JR SELECL
                        ;
259D    69              SELECH: LD L,C
                        ;
                        ;
                        ;THIS SELECTS THE Lth ONE
                        ;OF THE VARIABLE LIST DONE BY
                        ;READING FIRST BYTE AND THEN SKIPPING
                        ;DEPENDING ON E
                        ;
259E    D9              SELECL: EXX;SAVE HL'
259F    E5                      PUSH HL
25A0    21 2E04                 LD HL,TABLE2
25A3    D9                      EXX
25A4    7B                      LD A,E
25A5    CD 2590                 CALL SPLIT1
25A8    CD 2579                 CALL LISTL
25AB    7B                      LD A,E
25AC    E6 07                   AND 7
25AE    D9                      EXX
25AF    E1                      POP HL
25B0    D9                      EXX
25B1    C9                      RET
                        ;
                        ;
                        ;THIS PRINTS OUT THE STRING
                        ;POINTED TO BY HL'
                        ;IF BIT 0 OF E IS SET IT ALSO PRINTS
                        ;A SPACE A IS DEYSTROYED
                        ;
25B2    E5              LITERAL:        PUSH HL
25B3    D5                      PUSH DE
25B4    1E 06           LIT1:   LD E,6
25B6    D9                      EXX
25B7    7A                      LD A,D;GET INDEX
25B8    D9                      EXX
25B9    6F                      LD L,A
25BA    2C                      INC L ;SKIP 0
```

Move HL' to the start of the ATH list in a table.

```
25BB  D9              EXX;GET DASHED
25BC  7E              LD A,(HL);GET BYTE
25BD  23              INC HL
25BE  CB 7F           BIT 7,A;END YET
25C0  CB BF           RES 7,A
25C2  F5              PUSH AF
25C3  FE 61           CP 'a'
25C5  38 37           JR C,NOTW
25C7  D6 60           SUB 'a'-1
25C9  FE 04           CP 4
25CB  30 0C           JR NC,DGOTIXY
25CD  32 FDB0         LD (DTYPE),A ;NZ
25D0  ED 43 FDB1      LD (DISAD),BC ;Remember
25D4  CD 260F         CALL ASRCPOIN
25D7  18 28           JR MISS
25D9  20 06     DGOTIXY:    JR NZ,GOTBRHL
25DB  D9              EXX
25DC  CD 259E         CALL SELECL
25DF  18 21           JR MISS1
25E1  D7        GOTBRHL:    RST SCRRST
25E2  81 28           DB 80H+1,'('
25E4  D9              EXX
25E5  CD 259E         CALL SELECL
25E8  2D              DEC L
25E9  D9              EXX
25EA  28 10           JR Z,ADDBR
25EC  D7              RST SCRRST
25ED  81 2B           DB 80H+1,'+'
25EF  3E 01           LD A,1 ;data 1 byte
25F1  32 FD87         LD (ITYPE),A ;Flag nz
25F4  CD 260F         CALL ASRCPOIN
25F7  1C              INC E
25F8  1D              DEC E
25F9  28 01           JR Z,ADDBR
25FB  03              INC BC
25FC  3E 29     ADDBR:  LD A,')'
25FE  CD 1B6A   NOTW:   CALL CHOUT;ADD TO STRING
2601  D9        MISS:   EXX
2602  F1        MISS1:  POP AF
2603  28 AF               JR Z,LIT1;NEXT BYTE
2605  D1                  POP DE
2606  E1                  POP HL
2607  CB 43               BIT 0,E
2609  C8                  RET Z
260A  3E 20               LD A,' '
260C  C3 1B6A             JP CHOUT
                ;
                ;This searches for the pointer
                ;in BC,HL returns at the start of
                ;the string if found A has pointer
                ;type.Adds on string if label found
                ;otherwise data byte is added
                ;
260F  E5        ASRCPOIN:   PUSH HL
2610  F5                  PUSH AF
2611  D5                  PUSH DE
```

```
2612  C5                PUSH BC
2613  60                LD H,B
2614  69                LD L,C
2615  ED 4B FDC7        LD BC,(BASEM)
2619  A7                AND A
261A  ED 42             SBC HL,BC
261C  EB                EX DE,HL
261D  CD 29DA           CALL SRCPOIN
2620  C1                POP BC
2621  D1                POP DE
2622  20 0E             JR NZ,DONORM
2624  CD 29F4           CALL MOVESTR
2627  CD 1907           CALL OUTSTR
262A  F1                POP AF
262B  E1                POP HL
262C  03                INC BC
262D  FE 02             CP 2
262F  C0                RET NZ
2630  03                INC BC
2631  C9                RET
2632  F1          DONORM: POP AF
2633  E1                POP HL
2634  3D                DEC A
2635  20 02             JR NZ,DTWOBYTE
2637  18 1D             JR OUTEX
2639  3D          DTWOBYTE:   DEC A
263A  20 0A             JR NZ,NOTSD
263C  0A                LD A,(BC)
263D  03                INC BC
263E  F5                PUSH AF
263F  CD 2656           CALL OUTEX
2642  F1                POP AF
2643  C3 1B55     NOTDD: JP VHEX
2646  0A           NOTSD: LD A,(BC)
2647  03                INC BC
2648  E5                PUSH HL
2649  6F                LD L,A
264A  17                RLA
264B  9F                SBC A,A
264C  67                LD H,A
264D  09                ADD HL,BC
264E  7C                LD A,H
264F  CD 2658           CALL OUTDEX
2652  7D                LD A,L
2653  E1                POP HL
2654  18 ED             JR NOTDD
                        ;
                        ;
                        ;THIS OUTPUTS THE HEX BYTE IN
                        ;(BC)
                        ;
2656  0A          OUTEX: LD A,(BC)
2657  03                INC BC
2658              OUTDEX: ;PUSH AF
2658  D7                RST SCRRST
2659  81 23            DB 81H,'£'
```

```
                        ;       POP AF
265B    C3 1855                 JP VHEX
                        ;
                        ;
                        ;*** ASEM.INC ***
                        ;
                        ;This sets the high bit at the end of an
                        ;address label etc.
                        ;HL returns at end DE at start
                        ;
265E    D5              SETA7:  PUSH DE
265F    CD 29FB                 CALL ENDLABEL
2662    E1                      POP HL
2663    EB                      EX DE,HL
2664    2B                      DEC HL
2665    CB FE                   SET 7,(HL)
2667    23                      INC HL
2668    C9                      RET
                        ;
                        ;B is the flag register
                        ;0,1,2=F,g or h
                        ;3=data Y or N
                        ;4=size set for 2 byte data
                        ;5=ix+d type label
                        ;6=defb,s,w
                        ;7=succesful
                        ;C=Length E= length wihout code
                        ;
                        ;ADLABEL=address of ADDRESS label
                        ;ADLABEL+1=present above
                        ;COMMNET=NZ if comment present(find 0ffh
                        ;DATALABEL has start of data string,or start of defb
                        ;IXLABEL= start of IX+d type string
                        ;
                        ;THIS IS THE ASSEMBLY PROGRAM
                        ;It assembles the string held in (kbdbuf)
                        ;
                        ;THERE ARE SEVERAL BYTES THAT ARE USED
                        ;TO HOLD DATA ETC.
                        ;DBYTE IS USED TO STORE THE BYTE AFTER
                        ; AN INDEXED ADDRESS
                        ;INDEX HAS 0,DD,OR FD REPRESENTING THE INDEX
                        ;ASBYTE IS A 4 BYTE STORE TO STORE THE ASSEMBLED
                        ; BYTES
                        ;NZ if not assembled.
                        ;
2669    21 2809         ASEMBLER:       LD HL,MASTASEM
266C    22 FDA3                 LD (WCHJUM),HL;SET JUMP TABLE
266F    2A FA83                 LD HL,(KBDBUF)
2672    22 FDB4                 LD (ERRPOS),HL
2675    AF                      XOR A
2676    32 FDBB                 LD (COMMENT),A
2679    32 FDBF                 LD (ADLABEL+1),A
267C    E5                      PUSH HL
267D    CD 1B43                 CALL CALLEN     BC = Line length
2680    3E 3B                   LD A,';'
```

```
2682  ED B1                 CPIR        - Locate comment
2684  20 06                 JR NZ,NOCOME - Jump if none
2686  2B                    DEC HL
2687  36 FF                 LD (HL),0FFH ;New end of line   } Replace ; by end of line, and flag comment
2689  32 FDBB               LD (COMMENT),A ;Flag NZ
268C  E1              NOCOME: POP HL
268D  E5                    PUSH HL
268E  CD 1A7E               CALL ALLCAP - Convert line to caps
2691  D1                    POP DE
2692  DA 2791               JP C,BYTNTFND - Jump if unterminated string
2695  CD 1AD3               CALL READ1HEX - Read address
2698  C2 1E21               JP NZ,LINEINV - Jump if invalid
269B  CD 1EBF               CALL TESTPROG } Check legal address
269E  C2 1E21               JP NZ,LINEINV } Jump if not
26A1  ED 43 FDB7            LD (APROG),BC - Set current address
26A5  CD 1A19               CALL FINDNSPC - Find next character
26A8  ED 53 FD84            LD (ERRPOS),DE - Update Error position
26AC  CD 1AF5               CALL SETCHL - Find length remaining
26AF  0D                    DEC C                          } Exit if zero
26B0  20 02                 JR NZ,NOTDEL                   } length line
26B2  0D                    DEC C                          }
26B3  C9                    RET     ;No characters         }
26B4  0D              NOTDEL: DEC C  } Exit if only one
26B5  C8                    RET Z  } character in line
26B6  0C                    INC C
26B7  7E                    LD A,(HL)                      } Jump if
26B8  FE 5B                 CP 'Z'+1                       } not letter
26BA  D2 2791               JP NC,BYTNTFND                 }
26BD  FE 41                 CP 'A'                         }
26BF  DA 2791               JP C,BYTNTFND                  }
26C2  3E 3A                 LD A,':'  } Search for :
26C4  ED B1                 CPIR
26C6  20 29                 JR NZ,NOLABEL - Jump if not found
26C8  ED 53 FDBE            LD (ADLABEL),DE ;NZ
26CC  CD 265E               CALL SETA7 - Set high bit at end of label
26CF  7E                    LD A,(HL)                      } Check label termination.
26D0  FE 3A                 CP ':'                         } Jump if not found
26D2  C2 2791               JP NZ,BYTNTFND                 }
26D5  EB                    EX DE,HL
26D6  13                    INC DE
26D7  CD 1A19               CALL FINDNSPC - Find start of op-code
26DA  ED 53 FD84            LD (ERRPOS),DE - Update error pointer
26DE  3A FDAD               LD A,(EDIT)                    }
26E1  A7                    AND A                          } Jump if EDIT mode
26E2  20 0D                 JR NZ,NOLABEL ;Overwrite       }
26E4  D5                    PUSH DE
26E5  EB                    EX DE,HL        ;DE at start
26E6  CD 2A4C               CALL TESTLAB - Test for existing label
26E9  20 05                 JR NZ,TESTL1 - Jump if none
26EB  7E                    LD A,(HL)                      } Fail with multiple definition
26EC  A7                    AND A                          } if not [illegible] type #.
26ED  C2 1F12               JP NZ,MULTDEF                  }
26F0  D1              TESTL1: POP DE
                      ;Now check for defbs defws or defs
26F1  1A              NOLABEL:      LD A,(DE)  } Jump if first
26F2  FE 44                 CP 'D'             } character not D
```

```
26F4  20 5C           JR NZ,NODF
26F6  13              INC DE        } Get next character
26F7  1A              LD A,(DE)
26F8  F5              PUSH AF
26F9  13              INC DE
26FA  CD 1A19         CALL FINDNSPC } Jump if next character is space
26FD  28 03           JR Z,GETREAD
26FF  F1              POP AF
2700  18 50           JR NODF
2702  F1        GETREAD:    POP AF
2703  ED 53 FDC3           LD (DATALAB),DE ;Save address
2707  32 FDC6              LD (DBLABEL),A ;Flag NZ
270A  FE 42              CP 'B'        } Jump if not DB
270C  20 09              JR NZ,DEFWW
270E  CD 27DF            CALL DEFLEN   - C = Number of bytes found
2711  C2 2791            JP NZ,BYTNTFND - Jump if none
2714  06 03              LD B,3        } Exit with B=3
2716  C9                 RET
2717  FE 57        DEFWW: CP 'W'        } Jump if not DW
2719  20 1A              JR NZ,DEFSS
271B  CD 27DF            CALL DEFLEN   C = Number of words defined
271E  C2 2791            JP NZ,BYTNTFND Jump if none.
2721  CB 21              SLA C         Double C
2723  DA 2791            JP C,BYTNTFND - Jump if too many words
2726  C5                 PUSH BC
2727  CD 1AF5            CALL SETCHL
272A  3E 22              LD A,'"' ;No quotes
272C  ED B1              CPIR          } Jump if definition includes strings
272E  C1                 POP BC
272F  CA 2791            JP Z,BYTNTFND
2732  06 04              LD B,4        } Exit with B=4
2734  C9                 RET
2735  FE 53        DEFSS: CP 'S'        } Jump if not DS
2737  C2 2752            JP NZ,NODF
273A  CD 1AB1            CALL READNUM  - Read hex or dec number (or string)
273D  C2 2791            JP NZ,BYTNTFND - Jump if not valid
2740  CD 1A56            CALL ONEVAL   } Check <256
2743  C2 2791            JP NZ,BYTNTFND
2746  0C                 INC C
2747  C8                 RET Z
2748  0D                 DEC C
2749  CD 27EC            CALL ENDLLL   } Check for END LINE
274C  C2 2791            JP NZ,BYTNTFND Jump if not
274F  06 05              LD B,5        } Exit with B=5
2751  C9                 RET
2752  AF           NODF: XOR A         } Zero definition type
2753  32 FDC6            LD (DBLABEL),A
2756  CD 1AF5            CALL SETCHL
2759  7E                 LD A,(HL)
275A  23                 INC HL
275B  FE 48              CP 'H'
275D  20 11              JR NZ,DONEXIT
275F  7E                 LD A,(HL)
2760  FE 41              CP 'A'
2762  20 0C              JR NZ,DONEXIT
2764  3E 76              LD A,76H
```

Jump if hex has bytes not HA

```
2766  06 00              LD B,0
2768  1E 00              LD E,0
276A  21 FDA8            LD HL,ASBYTE
276D  C3 27CD            JP SAHALT
2770  11 0000    DONEXIT:    LD DE,0
2773  01 0000            LD BC,0
2776  21 2BC5            LD HL,MATABLE
2779  7E          TOPAS:  LD A,(HL)
277A  23                  INC HL
277B  CB 7F               BIT 7,A
277D  CB BF               RES 7,A
277F  57                  LD D,A
2780  F5                  PUSH AF
2781  A7                  AND A
2782  1F                  RRA
2783  1F                  RRA
2784  CD 2542             CALL CALTABL
2787  E5                  PUSH HL
2788  CD 1B01             CALL MAST1
278B  E1                  POP HL
278C  28 06               JR Z,FNDBYT
278E  F1                  POP AF
278F  28 E8               JR Z,TOPAS
                  ;
                  ;ROUTINE FOR BYTE NOT FOUND
                  ;
2791  0E 00       BYTNTFND:   LD C,0
2793  C9                RET
                  ;
                  ;ROUTINE FOR BYTE FOUND
                  ;
2794  F1          FNDBYT: POP AF
2795  21 FDA8             LD HL,ASBYTE
2798  1E 00               LD E,0;COUNTER
279A  CB 70               BIT 6,B
279C  28 02               JR Z,NOIND
279E  23                  INC HL
279F  1C                  INC E
27A0  78          NOIND:  LD A,B
27A1  E6 07               AND 7
27A3  17                  RLA
27A4  17                  RLA
27A5  17                  RLA
27A6  4F                  LD C,A
27A7  7A                  LD A,D
27A8  E6 07               AND 7
27AA  B1                  OR C
27AB  4F                  LD C,A
27AC  7A                  LD A,D
27AD  17                  RLA
27AE  17                  RLA
27AF  30 08               JR NC,NOTAED
27B1  CB 70               BIT 6,B
27B3  20 DC               JR NZ,BYTNTFND
27B5  36 ED               LD (HL),0EDH
27B7  23                  INC HL
```

```
27B8  1C                      INC E
27B9  17              NOTAED: RLA
27BA  30 09                   JR NC,NOTCB
27BC  36 CB                   LD (HL),0CBH
27BE  23                      INC HL
27BF  1C                      INC E
27C0  CB 70                   BIT 6,B
27C2  28 01                   JR Z,NOTCB
27C4  23                      INC HL ;JUMP OVER DATA
27C5  E6 C0           NOTCB:  AND 0C0H
27C7  B1                      OR C
27CB  FE 76                   CP 76H
27CA  CA 2791                 JP Z,BYTNTFND
27CD  77              SAHALT: LD (HL),A;PUT THE BYTE AWAY
27CE  23                      INC HL
27CF  1C                      INC E
27D0  4B                      LD C,E
27D1  CB 68                   BIT 5,B
27D3  28 01                   JR Z,NOIXD
27D5  0C                      INC C
27D6  CB 60           NOIXD:  BIT 4,B
27D8  C8                      RET Z
27D9  0C                      INC C
27DA  CB 58                   BIT 3,B
27DC  C8                      RET Z
27DD  0C                      INC C
27DE  C9                      RET
                      ;
                      ;This calculates the length of
                      ;a defb or defw string
                      ;
27DF  0E 00           DEFLEN: LD C,0 ;Counter
27E1  CD 29FB                 CALL ENDLABEL
27E4  9F                      SBC A,A
27E5  C0                      RET NZ ;No labels
27E6  0C              SLEB:   INC C
27E7  1A                      LD A,(DE)
27E8  FE 2C                   CP ','
27EA  28 07                   JR Z,DOLLL
27EC  CD 1A19         ENDLLL: CALL FINDNSPC
27EF  1A                      LD A,(DE)
27F0  FE FF                   CP 0FFH
27F2  C9                      RET
27F3  13              DOLLL:  INC DE
27F4  CD 29FB                 CALL ENDLABEL
27F7  30 ED                   JR NC,SLEB
27F9  9F                      SBC A,A ;Set NZ
27FA  C9                      RET
                      ;
                      ;C set if not found
                      ;
27FB  1A              LOOKNUM:        LD A,(DE)
27FC  FE 30                   CP '0'
27FE  D8                      RET C
27FF  FE 3A                   CP '9'+1
2801  3F                      CCF
```

```
2802  C9                    RET
                      ;
                      ;
                      ;
2803  CD 28B0         ASELH: CALL ASELECC
2806  82                    ADD A,D
2807  57                    LD D,A
2808  C9                    RET
                      ;
                      ;
2809  2819            MASTASEM:   DW ASPLIT ;0
280B  28DD                DW ALIT ;8
280D  2859                DW ALISTG ;16
280F  285F                DW ALISTH ;24
2811  288A                DW ASELG ;32
2813  2803                DW ASELH ;40
2815  282F                DW ASKI ;48
2817  2848                DW AKSKIP ;56
                      ;
                      ;
                      ;
2819  7A              ASPLIT: LD A,D
281A  CD 2590             CALL SPLIT1  ;Move HL' along
281D  CD 1B01         AS1PLIT:    CALL MAST1    ;HL' been incremented
2820  C8                  RET Z
2821  14                  INC D
2822  7A                  LD A,D
2823  E6 07               AND 7
2825  D9                  EXX
2826  23                  INC HL
2827  D9                  EXX
2828  20 F3               JR NZ,AS1PLIT
282A  7A                  LD A,D
282B  D6 08               SUB 8
282D  57                  LD D,A
282E  C9                  RET
                      ;
                      ;
282F  CD 283D         ASKI:  CALL ASKIPA
2832  C0                  RET NZ;NOT FOUND
2833  D0                  RET NC
2834  CB 50               BIT 2,B
2836  28 02               JR Z,BLESS4
2838  CB F8               SET 7,B
283A  CB D0           BLESS4: SET 2,B;LET G=G+4
283C  C9                  RET
                      ;
                      ;
                      ;Z FLAG IS SET IF THE COMPARE WAS
                      ;TRUE,C IS THEN SET IF FOUND ON SECOND ONE
                      ;RESET ON THE FIRST ONE
                      ;
                      ;
283D  CD 1AFE         ASKIPA: CALL MA1
2840  37                  SCF
2841  3F                  CCF;SET CARRY TO ZERO
```

```
2842  C8                    RET Z
2843  CD 1B01               CALL MAST1;NO DATA TO SKIP
2846  37                    SCF ;SET THE CARRY
2847  C9                    RET
                      ;
                      ;
2848  CD 283D         AKSKIP: CALL ASKIPA
284B  C0                    RET NZ
284C  78                    LD A,B
284D  88                    ADC A,B
284E  E6 07                 AND 7
2850  F5                    PUSH AF
2851  78                    LD A,B
2852  E6 F8                 AND 0F8H
2854  47                    LD B,A
2855  F1                    POP AF
2856  80                    ADD A,B
2857  47                    LD B,A
2858  C9                    RET
                      ;
                      ;
2859  CD 2865         ALISTG: CALL ALISTC
285C  80                    ADD A,B ;IN
285D  47                    LD B,A
285E  C9                    RET
                      ;
                      ;
285F  CD 2865         ALISTH: CALL ALISTC
2862  82                    ADD A,D
2863  57                    LD D,A;
2864  C9                    RET;
                      ;
                      ;
                      ;THIS ROUTINE REQURIES THAT THE NUMBER
                      ;IN THE STRING IS HELD IN E.C RETURNS
                      ;WITH 0 A WITH E+1 IF STRING NOT FOUND
                      ;OTHERWISE
                      ;A RETURNS WITH THE NUMBER OF WHERE IT
                      ;WAS FOUND.
                      ;
                      ;
2865  7B              ALISTC: LD A,E;GET NUMBER IN
2866  E6 07                 AND 7;LIST
2868  3C                    INC A
2869  4F                    LD C,A;COUNTER
286A  CB 78                 BIT 7,B
286C  C2 2589               JP NZ,FINDEX
286F  D5              STALI:  PUSH DE
2870  CB 83                 RES 0,E
2872  CB B8                 RES 7,B
2874  CD 28DD               CALL ALIT
2877  D1                    POP DE ;get it back
2878  20 07                 JR NZ,CONLIS
287A  0D                    DEC C;MOVE TO THE END
287B  79                    LD A,C;OF THE STRINGS
287C  CD 2589               CALL FINDEX
```

Shift 3 lsb of B left 1 bit, and carry flag into lsb. (Bit 2 lost).

B = B + Position of matching string in list.

D = D + Position of matching string in list.

C = A = Number of strings in list

Jump if compare has already failed

Reset further flag

Compare strings

Jump if strings differ

Skip remainder of string list

```
287F  18 04              JR FOUND
2881  0D           CONLIS: DEC C      - Decrement Counter
2882  20 EB              JR NZ,STALI - Jump if another string to try
2884  4B                 LD C,E;THIS WILL GIVE 0
2885  7B           FOUND: LD A,E
2886  91                 SUB C
2887  E6 07              AND 7
2889  C9                 RET
                         ;
                         ;
288A  CD 2880      ASELG: CALL ASELECC
288D  80                  ADD A,B
288E  47                  LD B,A
288F  C9                  RET
                         ;
                         ;
2890  D5      Main ASELHIXY:    PUSH DE
2891  C5                  PUSH BC
2892  1E 06               LD E,6
2894  CD 28B0             CALL ASELECC
2897  C1                  POP BC
2898  D1                  POP DE
2899  A7                  AND A
289A  C8                  RET Z
289B  2E 00               LD L,0
289D  3D                  DEC A
289E  28 09               JR Z,SETB
28A0  CB F0               SET 6,B
28A2  2E DD               LD L,0DDH
28A4  3D                  DEC A
28A5  28 02               JR Z,SETB
28A7  2E FD               LD L,0FDH
28A9  7D           SETB:  LD A,L
28AA  32 FDA8             LD (INDEX),A
28AD  37                  SCF
28AE  9F                  SBC A,A
28AF  C9                  RET
                         ;
                         ;
28B0  D9      Dark ASELECC:     EXX
28B1  E5                  PUSH HL
28B2  21 2E04             LD HL,TABLE2
28B5  D9      Main        EXX
28B6  7B                  LD A,E
28B7  E6 07               AND 7
28B9  CB 7B               BIT 7,E;IS IT SET
28BB  D5                  PUSH DE;SAVE IT
28BC  F5                  PUSH AF;FLAGS
28BD  1E 07               LD E,7;NUMBER IN LIST
28BF  FE 06               CP 6
28C1  28 08               JR Z,THER4
28C3  FE 01               CP 1;IS THERE 4 IN LIST
28C5  28 04               JR Z,THER4
28C7  FE 02               CP 2
28C9  20 02               JR NZ,THER8
28CB  1E 03        THER4: LD E,3
```

A = String number

B = B + Position of match in Eth list in TABLE2

Compare input string against HL, IX, IY

Exit if no match to all.

Load (IX+dx) with 00, or FD depending on HL, IX, IY. Bit 6 of B set for index register

Ret with NZ

HL' = Start of parameter table

A = 3 lsb of E. Section of parameter table.

Test and save msb of E

E = 3 if A = 1, 2, or 6 else E = 7

Search for a match in the Eth list of TABLE2. A returns with position in list.

```
28CD  F1          THER8:  POP AF
28CE  28 02               JR Z,NOTESET
28D0  CB FB               SET 7,E
28D2  CD 2590     NOTESET:        CALL SPLIT1;FIND START OF
                                  ;STRINGS
28D5  CD 2865             CALL ALISTC
28D8  D1                  POP DE
28D9  D9                  EXX
28DA  E1                  POP HL
28DB  D9                  EXX
28DC  C9                  RET
                        ;
                        ;
                        ;THIS LOOKS FOR THE CHARACTER POINTED
                        ;TO BY HL IF IT IS FOUND THEN HL IS LEFT
                        ;POINTING TO THE CHARACTER OTHERWISE IT
                        ;IS PUT TO ITS ORIGINAL VALUE RETURNS ZERO
                        ;IF A TRUE COMPARE HL IS LEFT AT END OF STRING
                        ;BIT 7 OF B IS SET IF THERE IS AN UNTRUE
                        ;COMPARE
                        ;
                        ;
28DD  CB 78       ALIT:   BIT 7,B
28DF  D9                  EXX
28E0  C2 2977             JP NZ,NOBOTHER
28E3  D5                  PUSH DE
28E4  7E          COM1:   LD A,(HL) ;Get data
28E5  23                  INC HL
28E6  CB 7F               BIT 7,A ;End yet
                                  ;Dash regs
28E8  F5                  PUSH AF
28E9  CB BF               RES 7,A
28EB  FE 20               CP ' '
28ED  20 07               JR NZ,TESTDATA
28EF  CD 1A19             CALL FINDNSPC
28F2  28 60               JR Z,AF1
28F4  18 7E               JR NOTF1
28F6  FE 61       TESTDATA:       CP 'a'
28F8  38 54               JR C,ANORM
28FA  D9                  EXX
28FB  D6 60               SUB 'a'-1
28FD  FE 04               CP 4
28FF  30 17               JR NC,GOTHLIXY
2901  FE 02               CP 2
2903  20 02               JR NZ,NOT2BYTE
2905  CB D8               SET 3,B
2907  32 FDC3   NOT2BYTE:       LD (DATALABEL),A
290A  CB E0               SET 4,B
290C  D9                  EXX
290D  ED 53 FDC4          LD (DATALABEL+1),DE
2911  CD 29FB             CALL ENDLABEL
2914  38 5E               JR C,NOTF1
2916  18 3C               JR AF1 ;On dashed
2918  28 2C     GOTHLIXY:       JR Z,NODATA ;not (HL)
291A  D9                  EXX
291B  1A                  LD A,(DE)
```

```
291C  13              INC DE
291D  D9        Main  EXX
291E  FE 28           CP '('
2920  20 51           JR NZ,NOTFOUN
2922  CD 2890         CALL ASELHIXY
2925  28 4C           JR Z,NOTFOUN
2927  A5              AND L
2928  28 13           JR Z,AHLIX
292A  CB E8           SET 5,B
292C  D9        Dash  EXX
292D  1A              LD A,(DE)
292E  13              INC DE
292F  FE 2B           CP '+'
2931  20 41           JR NZ,NOTF1 ;On dashed
2933  ED 53 FDC1      LD (INDEXLAB),DE
2937  CD 29FB         CALL ENDLABEL
293A  D9        Main  EXX
293B  38 36           JR C,NOTFOUN
293D  D9        Dash  AHLIX: EXX
293E  1A              LD A,(DE)
293F  13              INC DE
2940  FE 29           CP ')'
2942  28 10           JR Z,AF1
2944  18 2E           JR NOTF1
2946  CD 2890   Main  NODATA: CALL ASELHIXY
2949  28 28           JR Z,NOTFOUN
294B  D9        Dash  EXX
294C  18 06           JR AF1
294E  EB        Dash  ANORM: EX DE,HL
294F  BE              CP (HL)
2950  23              INC HL
2951  EB              EX DE,HL
2952  20 20           JR NZ,NOTF1
2954  F1              AF1: POP AF
2955  28 8D           JR Z,COM1
2957  D9        Main  EXX
2958  CB 43           BIT 0,E
295A  28 07           JR Z,MFOUN
295C  D9        Dash  EXX
295D  CD 1A19         CALL FINDNSPC
2960  20 13           JR NZ,NOTF2
2962  D9        Main  EXX
2963  CB 7B           MFOUN: BIT 7,E
2965  28 09           JR Z,FOUN
2967  D9        Dash  EXX
2968  CD 1A19         CALL FINDNSPC
296B  1A              LD A,(DE)
296C  3C              INC A
296D  20 06           JR NZ,NOTF2
296F  D9        Main  EXX
2970  33              FOUN: INC SP
2971  33              INC SP
2972  C9              RET
                      ;
2973  D9              NOTFOUND:       EXX
2974  F1              NOTF1:  POP AF
```

Test for open bracket

Main

Test for HL, IX, IY - Fail if not found

Jump if HL

Dash

Fail if next byte not '+'

- Store start of index label
- Find end

Main

- Fail if not valid

Dash

Fail if next byte not ')'

Main

Test for HL, IX, IY - Fail if not found.

Dash

Compare byte in input string with that from assembly table. Jump if different.

Main

Continue comparison unless bit 7 of assembly table byte was set

Dash

If bit 0 of E set then skip spaces in input string, failing if current byte is not a space.

Main

If bit 7 of E set then fail if next non blank character is not END LINE

Discard old input string pointer

```
2975  D1              NOTF2:  POP DE
2976  2B                      DEC HL
2977  CD 1BDD         NOBOTHER:       CALL MOVHL
297A  D9                      EXX
297B  CB F8                   SET 7,B
297D  C9                      RET
                      ;
                      ;
                      ;
                      ;KBDTEST        EQU     0FFD6H
0003                  CNTLC   EQU     3
                      ;FLAGS  EQU     0
                      ;READF  EQU     20
                      ;OPEN   EQU     15
                      ;FCB    EQU     5CH
                      ;FCBDN  EQU     FCB+0
                      ;FCBFN  EQU     FCB+1
                      ;FCBFT  EQU     FCB+9
                      ;FCBRL  EQU     FCB+12
                      ;FCBRC  EQU     FCB+15
                      ;FCBCR  EQU     FCB+32
                      ;FCBLN  EQU     FCB+33
                      ;BIOS   EQU     5
                      ;
                      ;
                      ;THIS CAN BE CALLED AFTER A SUCCESSFUL
                      ;OPEN
                      ;
                      ;
                      ;READFIL:       LD DE,100H
                      ;MORFIL:        LD HL,80H
                      ;       LD BC,128
                      ;       CALL DISKR
                      ;
                      ;SETUP FILE
                      ;
                      ;SETUP: LD C,OPEN
                      ;       LD DE,FCB
                      ;       CALL BIOS
                      ;CHECK ERRORS
                      ;       CP 255
                      ;       RET Z
                      ;
                      ;OPNOK: LD A,0
                      ;       LD (FCBCR),A
                      ;       RET
                      ;
                      ;DISKR: PUSH HL
                      ;       PUSH DE
                      ;       PUSH BC
                      ;       LD DE,FCB
                      ;       LD C,READF
                      ;       CALL BIOS
                      ;       POP BC
                      ;       POP DE
                      ;       POP HL
```

Restore DE' and advance HL' to start of next string. Set first bit (7) in B

```
                        ;       RET
                        ;
                        ;
                        ;
                        ;
                        ;***THESE ARE THE INPUT OUTPUT ROUTINES***
                        ;
                        ;
                        ;THIS GETS A 4 BYTE HEX NUMBER FROM
                        ;THE KEYBOARD NUMBER IS TERMINATED
                        ;BY A NON HEX CHARACTER NUMBER
                        ;IS RETURNED IN DE LAST CHARACTER
                        ;INPUTTED IN A B=0 IF NOTHING WAS
                        ;INPUT
                        ;
                        ;
                        ;
                        ;
                        ;
297E    E1              MOREHEX:        POP HL ;Return address
297F    D7                      RST SCRRST
2980    4F                      DB 40H+SEDIT+CLSE
2981    CD 1907                 CALL OUTSTR
2984    E5                      PUSH HL ;Return
2985    D7                      RST SCRRST
2986    81                      DB 80H+1
2987    3E                      DB ')'
                        ;
                        ;This gets a string from the keyboard
                        ;Tests for a dot.If nothing input then
                        ;Z returned otherwise if a value input
                        ;then NZ returned.Returned in BC.C set
                        ;if first character was a dot.
                        ;
2988    D7              INHEX:  RST SCRRST
2989    1E                      DB CURSON
298A    CD 1B23                 CALL GETCHAR
298D    FE 2E                   CP '.'
298F    20 05                   JR NZ,TRY23
2991    D7                      RST SCRRST
2992    1F                      DB CURSOFF
2993    AF              EXDOT:  XOR A
2994    37                      SCF
2995    C9                      RET
                        ;
                        ;
2996    CD 1B6F         TRY23:  CALL INSTR1
2999    38 0A                   JR C,FBE1
299B    C8                      RET Z  ;No chars
299C    CD 1AE0                 CALL READHEX ;C=no good
299F    44                      LD B,H
29A0    4D                      LD C,L
29A1    3F                      CCF
29A2    9F                      SBC A,A
29A3    3F                      CCF
29A4    C0                      RET NZ ;some
```

```
29A5  C3 1C33          FBE1:   JP FBEGIN
                       ;
                       ;
                       ;OUTDE OUTPUTS THE HEX BYTE IN DE
                       ;TO THE SCREEN
                       ;
                       ;
                       ;OUTDE: LD A,D
                       ;       CALL OUTHEX
                       ;       LD A,E
                       ;       CALL HINIB
                       ;       POP AF
                       ;       CALL LONIB
                       ;       RET
                       ;HEXHL: LD A,(HL);GET LS NIBBLE
                       ;       INC HL ;NEXT LOCATI  BIT 7,(IX+FLAGS)
                       ;       JR Z,NOTREG
                       ;       LD A,(MAFD)
                       ;NOTREG:        LD B,8
                       ;       OR 28H
                        ;       CALL OUTHEX ;NEXT ONE
                       ;       RET
                       ;
                       ;
                       ;THIS OUTPUTS A NIBBLE IN A
                       ;ONLY A AFFECTED
                       ;
                       ;
                       ;HINIB: RLCA
                       ;       RLCA
                       ;       RLCA
                       ;       RLCA
                       ;LONIB: AND 0FH
                       ;       CP 0AH
                       ;       JR C,LEST9
                       ;       ADD A,7
                       ;LEST9: ADD A,'0' ;PUT TO ASCII
                       ;       CALL OUTCHAR
                       ;       RET
                       ;
                       ;
                       ;
                       ;THIS CHANGES THE COLOUR FOR OUTPUTTING
                       ;MEMORY
                       ;
                       ;
                       ;OUTHCOL:       NOP
                       ;OUTHNCOL:      BIT 1,(IX+FLAGS)
                       ;       CALL NZ,READVID
                       ;       BIT 0,(IX+FLAGS)
                       ;       JR Z,OUTHHEX
                       ;       PUSH HL
                       ;       LD HL,(PPOS)
                       ;       CALL OUTVID
                       ;       LD (PPOS),HL
                       ;       POP HL
```

```
;       CALL OUTCALL
;       DC ' '
;       JR RESCOL
;OUTHHEX:       CALL OUTHEX
;RESCOL:        RET
;;
;;
;;This reads the video ram pointed
;;to by HL
;;
;READVID:       LD A,H
;       AND 7
;       OUT (11H),A
;       LD A,L
;       OUT (10H),A
;       IN A,(12H)
;       RET
;
;OUTSCREEN:PUSH DE
;       PUSH HL
;       PUSH AF
;       PUSH BC
;       LD C,A
;       CALL 0F65DH
;       LD A,(0F40EH)
;       CALL 0F7C4H
;       CALL OUTCALL
;       DB FWD
;       DC ' '
;       POP BC
;       POP AF
;       POP HL
;       POP DE
;       RET
;
;
;THIS CHANGES THE COLOUR TO
;THAT IN A
;
;
;OUTCOL:        LD (ATTRIB),A
;       RET
;;
;
;This outputs a string until a high
;bit is set pointed to by HL.
;
;OUTDC: LD A,(HL)
;       INC HL
;       BIT 7,A
;       RES 7,A
;       CALL OUTCHAR
;       JR Z,OUTDC
;       RET
;
```

```
                ;
                ;KBTEST:        JP OFFD6H
                ;
                ;This outputs a character in A
                ;to the screen,at the printposition
                ;this is updated CR and LF are tested
                ;for.
                ;
                ;
                ;OUTCHAR:       PUSH AF
                ;       PUSH HL
                ;       LD HL,(PPOS)
                ;       CP LF
                ;       JR Z,DONEWLINE
                ;       CP CR
                ;       JR Z,DOCR
                ;       CP BS
                ;       JR Z,DOBS
                ;       CP EOL
                ;       JR Z,DOEOL
                ;       CP 12
                ;       JR Z,DOCLS
                ;       CALL OUTVID
                ;DOCONT:        LD (PPOS),HL
                ;DCONT: POP HL
                ;       POP AF
                ;       RET
                ;
                ;
                ;DONEWLINE:PUSH DE
                ;       LD DE,80
                ;       LD HL,(NEWPOS)
                ;       ADD HL,DE
                ;       LD (NEWPOS),HL
                ;       POP DE
                ;       JR DOCONT
                ;;
                ;
                ;DOCR:  LD HL,(NEWPOS)
                ;       JR DOCONT
                ;;
                ;DOBS:  DEC HL
                ;       JR DOCONT
                ;
                ;
                ;DOEOL: PUSH BC
                ;       LD BC,18
                ;       CALL ERASE
                ;       POP BC
                ;       JR DCONT
                ;
                ;DOCLS: PUSH BC
                ;       LD BC,2*1024
                ;       CALL ERASE
                ;       POP BC
                ;       LD HL,0
```

```
;       LD (NEWPOS),HL
;       JR DOCONT
;
;
;This erases BC caharacters from
;HL
;
;
;ERASE: LD A,' '
;       OUT (12H),A
;LERASE:        LD A,B
;       OR C
;       RET Z
;       LD A,H
;       AND 7
;       OR 0E0H
;       OUT (11H),A
;       LD A,L
;       OUT (10H),A
;       DEC BC
;       INC HL
;       JR LERASE
;
;This outputs the character in A
;to HL
;
;OUTVID:        PUSH AF
;       OUT (12H),A
;       LD A,(ATTRIB)
;       OUT (13H),A
;       LD A,H
;       AND 7
;       OR 0E0H
;       OUT (11H),A
;       LD A,L
;       OUT (10H),A
;       INC HL
;       POP AF
;       RET
;
;
;*** LABEL.INC ***
;
;
;All values are calculated relative to the
;start address of the program.
;The label table is set up as a big string
;with the labels stored inside the labels
;
; 0=an undefined number,This has its variable
;       name and then its pointers
;
; 1=a defined address label
;       the following bytes give its value
;       Then its name (starts with letter)
;       Then the number of pointers
```

```
                        ;       followed by the pointers themselves
                        ;
                        ; 2=an equate,which will need to be calculated
                        ;       The first bytes are a space for
                        ;       its value. The next giv where it was
                        ;       defined. Next its name and then
                        ;       The next ones give its string
                        ;       definition.Finally the pointers
                        ;
                        ; 3=a DEFB ,followed by where it belongs
                        ;       then how many bytes it occupies,
                        ;       a 1 byte value. Then comes the string
                        ;
                        ; 4=a DEFC ,same as above ?????
                        ;
                        ; 5=a DEFW, this has the same format as above
                        ;
                        ; 6=a DEFS, where it belongs then how many bytes
                        ;       (2 byte)then the the string that defined
                        ;       it.
                        ; 8=a comment this is followed by where it belongs
                        ;       then the string
                        ;
                        ;
                        ;
                        ;       Values for pointers.
                        ;       ------------------
                        ;The pointers are divided into three lots of
                        ;256 pointers i.e. the first byte specifies
                        ;how many one byte pointers there are then the
                        ;pointers then how many two byte pointers
                        ;and finally how many JR
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;This moves HL to the type in A
                        ;HL points to first data byte
                        ;
29A8    3E 04           ADDHL4: LD A,4 ;4 skips
29AA    3D              ADDHL:  DEC A
29AB    C8                      RET Z ;already there
29AC    C5                      PUSH BC
29AD    4E                      LD C,(HL)
29AE    23                      INC HL
29AF    06 00                   LD B,0
29B1    09                      ADD HL,BC
29B2    09                      ADD HL,BC
29B3    C1                      POP BC
29B4    18 F4                   JR ADDHL
                        ;
                        ;
                        ;
                        ;This searches through the label table
                        ;for the value in DE, NZ if notfound
```

```
                        ;This is for address pointers.
                        ;
                        ;
29B6  EF                SRCADD: RST JT
29B7  B4                        DB 0B4H
                        ;       CALL TEST1
29B8  18 02                     JR STRTSRC
29BA  EF                LOOPSRC:        RST JT
29BB  B5                        DB 0B5H
                        ;CALL GETLNEXTL
29BC  C0                STRTSRC:        RET NZ
29BD  A7                        AND A
29BE  28 FA                     JR Z,LOOPSRC ;No address pointer
29C0  3D                        DEC A
29C1  28 F7                     JR Z,LOOPSRC
29C3  3D                        DEC A
29C4  23                        INC HL
29C5  20 0A                     JR NZ,DODEFB
29C7  CD 1BB6           GET2HL: CALL TESTHLDE
29CA  20 EE             GET1HL: JR NZ,LOOPSRC
29CC  2A FDC9           GETHL:  LD HL,(CURLAB)
29CF  7E                        LD A,(HL)
29D0  C9                        RET
29D1  FE 06             DODEFB: CP 6
29D3  30 F2                     JR NC,GET2HL
29D5  CD 2ABE                   CALL DOESDELIE
29D8  18 F0                     JR GET1HL
                        ;
                        ;
                        ;This searches for data pointers
                        ;Value wanted is held in DE Tests
                        ;Type wanted is in A
                        ;Hl returns at the start of label
                        ;A has data type
                        ;
29DA  47                SRCPOIN:        LD B,A ;Save type wanted
29DB  EF                        RST JT
29DC  B4                        DB 0B4H
                        ;       CALL TEST1
29DD  18 02                     JR STRPOIN
29DF  EF                LOOPADD:        RST JT
29E0  B5                        DB 0B5H
                        ;?CALL GETLNEXTL
29E1  C0                STRPOIN:        RET NZ
29E2  FE 03                     CP 3
29E4  D2 2A63                   JP NC,SETNZ
29E7  CD 1BD7                   CALL FINPOIN ;Move to pointers
29EA  78                        LD A,B
29EB  CD 29AA                   CALL ADDHL
29EE  EF                        RST JT
29EF  BC                        DB 0BCH
                        ;       CALL TEST2BY
29F0  28 DA                     JR Z,GETHL
29F2  18 EB                     JR LOOPADD
                        ;
                        ;This moves HL to point to the start of
```

```
                        ;the label name assiming that it has one
                        ;
29F4  7E                MOVESTR:        LD A,(HL)
29F5  23                        INC HL
29F6  A7                        AND A
29F7  C8                        RET Z  ;Already there
29F8  23                        INC HL
29F9  23                        INC HL
29FA  C9                        RET
                        ;
                        ;This finds the end of a label
                        ;only numbers and letters are allowed
                        ;returns DE at end,C is set if any  no
                        ;characters.
                        ;
29FB  1A                ENDLABEL:       LD A,(DE)
29FC  FE 23                     CP '£'
29FE  20 0D                     JR NZ,LOKSTR
2A00  13                        INC DE
2A01  CD 1A2A                   CALL LKHEX
2A04  D8                        RET C
2A05  13                HEXL:   INC DE
2A06  CD 1A2A                   CALL LKHEX
2A09  30 FA                     JR NC,HEXL
2A0B  3F                        CCF
2A0C  C9                        RET
2A0D  FE 22             LOKSTR: CP '"'
2A0F  20 13                     JR NZ,LOKNUM
2A11  13                        INC DE
2A12  1A                        LD A,(DE)
2A13  FE 22                     CP '"'
2A15  20 03                     JR NZ,SRCQUO
2A17  13                        INC DE
2A18  37                        SCF
2A19  C9                        RET
2A1A  1B                SRCQUO: DEC DE
2A1B  EB                        EX DE,HL
2A1C  CD 1A95                   CALL MOVQUO
2A1F  EB                        EX DE,HL
2A20  0D                        DEC C
2A21  D8                        RET C
2A22  13                        INC DE
2A23  C9                        RET
2A24  CD 27FB           LOKNUM: CALL LOOKNUM ;Number
2A27  38 0B                     JR C,LOCAR
2A29  13                LKNUM:  INC DE
2A2A  CD 27FB                   CALL LOOKNUM
2A2D  30 FA                     JR NC,LKNUM
2A2F  3F                        CCF
2A30  C9                        RET
2A31  CD 2A3E           LOCAR:  CALL CHARAC
2A34  D8                        RET C ;Not valid
2A35  13                MORC:   INC DE
2A36  1A                        LD A,(DE)
2A37  CD 2A3E                   CALL CHARAC
2A3A  30 F9                     JR NC,MORC
```

Max number

Quoted string

Decimal Number

Alpha-Numeric String

```
2A3C  3F                    CCF
2A3D  C9                    RET
                    ;
                    ;This tests A to see if it is a valid
                    ;label character
                    ;
2A3E  FE 5B         CHARAC: CP 'Z'+1
2A40  3F                    CCF
2A41  D8                    RET C
2A42  FE 41                 CP 'A'
2A44  D0                    RET NC
2A45  FE 30                 CP '0'
2A47  D8                    RET C
2A48  FE 3A                 CP '9'+1
2A4A  3F                    CCF
2A4B  C9                    RET
                    ;
                    ;
                    ;
                    ;This searches through the label table
                    ;for the string pointed to by DE if a true
                    ;compare then HL returns at the start of the
                    ;label with a NZ
                    ;
2A4C  EF            TESTLAB:    RST JT
2A4D  B4                    DB 0B4H
                    ;CALL TEST1 ;Initialise
2A4E  18 02                 JR MOVTEST
2A50  EF            LOOK1STR:   RST JT
2A51  B5                    DB 0B5H
                    ;CALL GETLNEXT
2A52  C0            MOVTEST:    RET NZ
2A53  FE 03                 CP 3
2A55  30 0C                 JR NC,SETNZ
2A57  CD 29F4               CALL MOVESTR
2A5A  CD 1BC4               CALL COMPSTR
2A5D  2A FDC9               LD HL,(CURLAB);Start of label
2A60  20 EE                 JR NZ,LOOK1STR
2A62  C9                    RET
2A63  A7            SETNZ:  AND A
2A64  C9                    RET
                    ;
                    ;This deletes BC spaces from the labels
                    ;
2A65  F5            TDELETE:    PUSH AF
2A66  D5                    PUSH DE
2A67  E5                    PUSH HL
2A68  C5                    PUSH BC
2A69  CD 04B0               CALL COPYDOWN
2A6C  C1                    POP BC
2A6D  2A FDB9               LD HL,(ENDTABL)
2A70  A7                    AND A
2A71  ED 42                 SBC HL,BC
2A73  22 FDB9               LD (ENDTAB),HL
2A76  E1                    POP HL
2A77  D1                    POP DE
```

```
2A78  F1                    POP AF
2A79  C9                    RET
                    ;
                    ;This inserts BC spaces into the labels and
                    ;the at HL does not update labtabl
                    ;ENDTABL,HI returns pointing at start
                    ;BC and DE are unaffected
                    ;
2A7A  D5            TINSERT:    PUSH DE
2A7B  E5                    PUSH HL
2A7C  F5                    PUSH AF
2A7D  C5                    PUSH BC
2A7E  CD 0351               CALL MAKESPACE
2A81  C1                    POP BC
2A82  2A FDB9               LD HL,(ENDTABL)
2A85  09                    ADD HL,BC
2A86  22 FDB9               LD (ENDTABL),HL
2A89  F1                    POP AF
2A8A  E1                    POP HL
2A8B  D1                    POP DE
2A8C  C9                    RET
                    ;
                    ;This tests a string of two byte numbers
                    ;pointed to by HL for the value in DE
                    ;HL is moved to end of count
                    ;
2A8D  7E            TEST2BY:    LD A,(HL)
2A8E  23                    INC HL
2A8F  3C                    INC A
2A90  3D            LOOPTEST:   DEC A
2A91  28 06                 JR Z,EXTEST
2A93  CD 1BB6               CALL TESTHLDE
2A96  20 F8                 JR NZ,LOOPTEST
2A98  C9                    RET
2A99  3C            EXTEST: INC A
2A9A  C9                    RET
                    ;
                    ;This chugs along the numbers pointed to
                    ;by HL for values bigger or equal
                    ;to DE and adds BC to them
                    ;
2A9B  7E            DOADDS: LD A,(HL)
2A9C  23                    INC HL
2A9D  3C                    INC A
2A9E  3D            ADLO:   DEC A
2A9F  C8                    RET Z
2AA0  CD 2AA8               CALL TESTADD
2AA3  18 F9                 JR ADLO
                    ;
                    ;This tests the bytes pointed to by
                    ;HL to see if they are less than or
                    ;equal to DE if they are then BC is
                    ;added to HL
                    ;
2AA5  E1            POHL:   POP HL
2AA6  23                    INC HL
```

```
2AA7  C9                    RET
                    ;
2AA8  D5            TESTADD:    PUSH DE
2AA9  5E                    LD E,(HL)
2AAA  23                    INC HL
2AAB  56                    LD D,(HL)
2AAC  E3                    EX (SP),HL ;HL=DE
2AAD  A7                    AND A
2AAE  EB                    EX DE,HL ;At address- DE
2AAF  ED 52                 SBC HL,DE
2AB1  38 F2                 JR C,POHL
2AB3  19                    ADD HL,DE
2AB4  09                    ADD HL,BC ;New value
2AB5  EB                    EX DE,HL
2AB6  E3                    EX (SP),HL ;On the stack
2AB7  72                    LD (HL),D
2AB8  2B                    DEC HL
2AB9  73                    LD (HL),E
2ABA  23                    INC HL
2ABB  D1                    POP DE
2ABC  23                    INC HL
2ABD  C9                    RET
                    ;
                    ;
                    ;This tests to see if DE lies in the
                    ;DEFB or DEFW pointed to by HL
                    ;
2ABE  C5            DOESDELIE:PUSH BC
2ABF  42                    LD B,D
2AC0  4B                    LD C,E
2AC1  CF                    RST DEHLRST
2AC2  7E                    LD A,(HL)
2AC3  EB                    EX DE,HL
2AC4  37                    SCF
2AC5  ED 42                 SBC HL,BC ;HL>BC Then stop
2AC7  50                    LD D,B
2AC8  59                    LD E,C ;Restore DE
2AC9  30 04                 JR NC,EXDE
2ACB  4F                    LD C,A
2ACC  06 00                 LD B,0
2ACE  09                    ADD HL,BC ;HL<BC? C=yes
2ACF  3F            EXDE:   CCF
2AD0  9F                    SBC A,A
2AD1  C1                    POP BC
2AD2  C9                    RET
                    ;
                    ;
                    ;
                    ;This looks for the label pointed to
                    ;by DE A has how long it is
                    ;if it is found then it returns,
                    ;HL pointing to the start of the definition.
                    ;If one is not found then a new label is
                    ;created.DE points to start of string
                    ;
2AD3  EB            CRELAB: EX DE,HL
```

```
2AD4  CD 265E                 CALL SETA7 ;SetbitsDE start hl end
2AD7  CD 2A4C          CREAD: CALL TESTLAB
2ADA  C8                      RET Z
2ADB  EB                      EX DE,HL
2ADC  CD 1CB9                 CALL FINDEND
2ADF  EB                      EX DE,HL
2AE0  E5                      PUSH HL
2AE1  C5                      PUSH BC
2AE2  C6 04                   ADD A,4 ;3 pointers+type
2AE4  4F                      LD C,A
2AE5  06 00                   LD B,0
2AE7  EF                      RST JT
2AE8  BA                      DB 0BAH
                      ;       CALL TINSERT
2AE9  D6 04                   SUB 4
2AEB  4F                      LD C,A
2AEC  70                      LD (HL),B ;Undefined
2AED  23                      INC HL
2AEE  D5                      PUSH DE
2AEF  EB                      EX DE,HL
2AF0  ED B0                   LDIR    ;copy string
2AF2  EB                      EX DE,HL
2AF3  70                      LD (HL),B
2AF4  23                      INC HL
2AF5  70                      LD (HL),B
2AF6  23                      INC HL
2AF7  70                      LD (HL),B
2AF8  D1                      POP DE
2AF9  C1                      POP BC
2AFA  E1                      POP HL
2AFB  C9                      RET
                      ;
                      ;
                      ;This adds a pointer to any labels.The value
                      ;to be added is held in BC.A has the type.
                      ;creating a label if it does not already exist.
                      ;
                      ;
2AFC  C5              POINLAB:        PUSH BC
2AFD  06 00                   LD B,0
2AFF  CD 1AA1                 CALL RELBC
2B02  C5                      PUSH BC
2B03  F5                      PUSH AF
2B04  CD 2AD3                 CALL CRELAB
2B07  CD 1BD7                 CALL FINPOIN
2B0A  F1                      POP AF
2B0B  CD 29AA                 CALL ADDHL ;Move to pointers
2B0E  34                      INC (HL)
2B0F  28 0C                   JR Z,TOOMPOIN
2B11  23                      INC HL
2B12  01 0002                 LD BC,2
2B15  EF                      RST JT
2B16  BA                      DB 0BAH
                      ;       CALL TINSERT
2B17  C1                      POP BC
2B18  71                      LD (HL),C
```

```
2B19  23                    INC HL
2B1A  70                    LD (HL),B
2B1B  C1                    POP BC
2B1C  C9                    RET
                        ;
2B1D  35                TOOMPOIN:   DEC (HL)
2B1E  C1                    POP BC
2B1F  C1                    POP BC
2B20  C9                    RET
                        ;
                        ;This creates the address label pointed
                        ;to by DE a high bit is already put in
                        ;
2B21  C5                VALLAB: PUSH BC
2B22  CD 2AD7               CALL CREAD
2B25  7E                    LD A,(HL)
2B26  36 02                 LD (HL),2
2B28  23                    INC HL
2B29  A7                    AND A
2B2A  20 05                 JR NZ,DOSH
2B2C  01 0002               LD BC,2
2B2F  EF                    RST JT
2B30  BA                    DB 0BAH
                        ;       CALL TINSERT
2B31  C1                DOSH:   POP BC
2B32  71                    LD (HL),C
2B33  23                    INC HL
2B34  70                    LD (HL),B
2B35  C9                    RET
                        ;
                        ;This creates a comment (APRO6) has
                        ;The address value
                        ;
2B36  CD 2B4C           DOCOM:  CALL PUTLAB
2B39  C8                    RET Z   ;No comment
2B3A  01 0003               LD BC,3
2B3D  EF                    RST JT
2B3E  BA                    DB 0BAH
                        ;       CALL TINSERT
2B3F  36 08                 LD (HL),8
2B41  23                    INC HL
2B42  01 0000               LD BC,0
2B45  CD 1AA1               CALL RELBC
2B48  71                    LD (HL),C
2B49  23                    INC HL
2B4A  70                    LD (HL),B
2B4B  C9                    RET
                        ;
                        ;This puts in the label table the string
                        ;pointed to by HL ended with 0ffh
                        ;HI points start space created,Removes
                        ;any spaces
                        ;
2B4C  E5                PUTLAB: PUSH HL
2B4D  CD 0A15               CALL SQUASH ;HL start DE end
```

```
2B50  E1                  POP HL
2B51  EB                  EX DE,HL
2B52  ED 52               SBC HL,DE ;NC
2B54  C8                  RET Z  ;No characters
2B55  44                  LD B,H
2B56  4D                  LD C,L
2B57  2A FDB9             LD HL,(ENDTAB)
2B5A  03                  INC BC
2B5B  03                  INC BC
2B5C  EF                  RST JT
2B5D  BA                  DB 0BAH
                  ;       CALL TINSERT
2B5E  0B                  DEC BC
2B5F  71                  LD (HL),C ;Length
2B60  EB                  EX DE,HL
2B61  D5                  PUSH DE
2B62  13                  INC DE
2B63  ED B0               LDIR     ;Copy string
2B65  E1                  POP HL ;Flags not destroyed
2B66  C9                  RET
                  ;
                  ;This routine needs to be called to update
                  ;all the pointers to the labels
                  ;
2B67  EF          UPDL:   RST JT
2B68  B4                  DB 0B4H
                  ;CALL TEST1     ;Setup HL
2B69  18 02               JR UPDX
2B6B  EF          LOOPUP: RST JT
2B6C  B5                  DB 0B5H
                  ;CALL GETLNEXT
2B6D  C0          UPDX:   RET NZ ;Finished
2B6E  23                  INC HL
2B6F  A7                  AND A
2B70  28 07               JR Z,NOUPDAT
2B72  FE 01               CP 1
2B74  28 25               JR Z,NOUP1
2B76  CD 2AA8             CALL TESTADD ;Update it
2B79  FE 03       NOUPDAT:        CP 3
2B7B  30 EE               JR NC,LOOPUP
2B7D  EF                  RST JT
2B7E  B3                  DB 0B3H
                  ;       CALL MOVHL
2B7F  B6                  OR (HL) ;Any pointers left
2B80  23                  INC HL
2B81  B6                  OR (HL)
2B82  23                  INC HL
2B83  B6                  OR (HL)
2B84  20 05               JR NZ,NTAKOUT
2B86  CD 2B9F             CALL TAKOUT
2B89  18 E2               JR UPDX
2B8B  2B          NTAKOUT:        DEC HL
2B8C  2B                  DEC HL
2B8D  CD 2A9B             CALL DOADDS
2B90  CD 2A9B             CALL DOADDS
2B93  CD 2A9B             CALL DOADDS
```

```
2B96  CD 1A0F                 CALL TESTCUR
2B99  18 D2                   JR UPDX
2B9B  23              NOUP1:  INC HL
2B9C  23                      INC HL
2B9D  18 DA                   JR NOUPDAT
                      ;
                      ;This takes out the label pointed to by
                      ;CURLAB this is then reset to its previous
                      ;value
                      ;
2B9F  C5              TAKOUT: PUSH BC
2BA0  2A FDC9                 LD HL,(CURLAB)
2BA3  E5                      PUSH HL ;Start of length
2BA4  EF                      RST JT
2BA5  B5                      DB 0B5H
                      ;       CALL GETLNEXT
2BA6  C1                      POP BC
2BA7  ED 43 FDC9              LD (CURLAB),BC
2BAB  A7                      AND A
2BAC  ED 42                   SBC HL,BC
2BAE  44                      LD B,H
2BAF  4D                      LD C,L
2BB0  2A FDC9                 LD HL,(CURLAB)
2BB3  CD 2A65                 CALL TDELETE
2BB6  C1                      POP BC
2BB7  C3 1A12                 JP TESTEND
                      ;
                      ;Given a value in DE this recalculates
                      ;its relative value from basem
                      ;
2BBA  E5              RELDE:  PUSH HL
2BBB  2A FDC7                 LD HL,(BASEM)
2BBE  EB                      EX DE,HL
2BBF  A7                      AND A
2BC0  ED 52                   SBC HL,DE
2BC2  EB                      EX DE,HL
2BC3  E1                      POP HL
2BC4  C9                      RET
                      ;
                      ;
                      ;
                      ;(HL) is represented as e
                      ;HL is represented as d
                      ;JR is   as c
                      ;LD REG,n is     a
                      ;LD RREG,NN is  b
                      ;
                      ;
0010                  .RADIX 16
2BC5  08 10 4F        MATABLE:        DB 8,10,4F ;This is for LD A,I,etc.
2BC8  00 48 18                DB 0,48,18
2BCB  20 28 30 38             DB 20,28,30,38
2BCF  D0                      DB 50+80
2BD0  48 65 6C 6C     SECTABLE:       DB 48,65,6C,6C,6F,20
2BD4  6F 20
2BD6  43 61 72 6F             DB 43,61,72,6F
```

DE : DE - (BASEM)

```
000A                        .RADIX 10
                            ;
2BDA  53 5A 08 48           FLAGTABLE:DEFB 'SZ',BS,'H',BS,'VNC'
2BDE  08 56 4E 43
2BE2  41 46                         DEFB 'AF'
2BE4  42 43 44 45                   DEFB 'BC','DE','HL'
2BE8  48 4C
2BEA  49 58 49 59                   DEFB 'IX','IY'
2BEE  53 50 50 43                   DEFB 'SP','PC'
2BF2  2C0A 2CAC             TABBYT: DW C0F0,C0F1,C0F2,C0F3
2BF6  2CB3 2CB5
2BFA  2D3F 2D57                     DW C1F0,C1F1,C1F2,C1F3
2BFE  2D5D 2D63
2C02  2D69 2D68                     DW C2F0,C2F1,C2F2,C2F3
2C06  2DEA 2E02
00B0                        ZER     EQU     '0'+80H
00B1                        ONE     EQU     '1'+80H
00B2                        TWO     EQU     '2'+80H
00B8                        EIG     EQU     '8'+80H
00A0                        SPA     EQU     ' '+80H
                            ;
00C2                        BB      EQU     'B'+80H
00C7                        GG      EQU     'G'+80H
00CF                        OO      EQU     'O'+80H
00C9                        II      EQU     'I'+80H
00CC                        LL      EQU     'L'+80H
00D8                        XX      EQU     'X'+80H
00C8                        HH      EQU     'H'+80H
00C3                        CC      EQU     'C'+80H
00D2                        RR      EQU     'R'+80H
00D3                        SS      EQU     'S'+80H
00CD                        MM      EQU     'M'+80H
00CE                        NN      EQU     'N'+80H
00C6                        FF      EQU     'F'+80H
00DA                        ZZ      EQU     'Z'+80H
00C1                        AA      EQU     'A'+80H
00C4                        DD      EQU     'D'+80H
00D4                        TT      EQU     'T'+80H
00D0                        PP      EQU     'P'+80H
00C5                        EE      EQU     'E'+80H
00A9                        BR      EQU     ')'+80H
00E1                        VV      EQU     'a'+80H
00AC                        CO      EQU     ','+80H
00E2                        QQ      EQU     'b'+80H
00D9                        YY      EQU     'Y'+80H
00E4                        @@      EQU     'd'+80H
2C0A  00 20 30 17           C0F0:   DB 0,32,48,23;H=0**+
2C0E  93                            DB 16+3+80H
2C0F  4E 4F D0                      DB 'NO',PP
2C12  45 58 20 41                   DB 'EX AF,AF',27H+80H
2C16  46 2C 41 46
2C1A  A7
2C1B  44 4A 4E 5A                   DC 'DJNZ c'
2C1F  20 E3
2C21  4A 52 20 E3                   DC 'JR c'
2C25  09 4A D2                      DB 8+1,'J',RR
```

a = 1 byte value

b = 2 byte value

c = relative address

d = HL, IX, IY

e = (HL), (IX+dd), (IY+dd)

```
2C28  24 88          2        DB 32+4,8+80H
2C2A  2C E3                   DC ',c'
2C2C  11 38 07       4        DB 17,56,7;H=1‡‡
2C2F  09                      DB 8+1
2C30  4C C4                   DB 'L',DD
2C32  21             2        DB 32+1
2C33  88                      DB 8+80H
2C34  2C E2                   DC ',b'
2C36  08             1        DB 8
2C37  41 44 44 20             DC 'ADD d,'
2C3B  64 AC
2C3D  A1             3        DB 32+1+80H
2C3E  30 09                   DB 48,8+1;H=2‡‡
2C40  4C C4                   DB 'L',DD
2C42  97             1        DB 16+7+80H
2C43  28 42 43 29             DB '(BC),',AA
2C47  2C C1
2C49  41 2C 28 42             DB 'A,(BC',BR
2C4D  43 A9
2C4F  28 44 45 29             DB '(DE),',AA
2C53  2C C1
2C55  41 2C 28 44             DB 'A,(DE',BR
2C59  45 A9
2C5B  28 62 29 2C             DC '(b),d'
2C5F  E4
2C60  64 2C 28 62             DC 'd,(b)'
2C64  A9
2C65  28 62 29 2C             DC '(b),A'
2C69  C1
2C6A  41 2C 28 62             DC 'A,(b)'
2C6E  A9
2C6F  0C 38 05       4        DB 12,56,5;H=3‡‡
2C72  09                      DB 8+1
2C73  49 4E C3                DB 'IN',CC
2C76  A1             2        DB 32+80H+1
2C77  09                      DB 8+1
2C78  44 45 C3                DB 'DE',CC
2C7B  A1             3        DB 32+80H+1
2C7C  05 09                   DB 5,8+1;H=4‡‡
2C7E  49 4E C3                DB 'IN',CC
2C81  A0             3        DB 32+80H+0
2C82  05 09                   DB 5,8+1;H=5‡‡
2C84  44 45 C3                DB 'DE',CC
2C87  A0             3        DB 32+80H+0
2C88  07 09 4C C4             DB 7,8+1,'L',DD,32+0,8+80H
2C8C  20 88          2
2C8E  2C E1                   DC ',a'  ;H=6‡‡
2C90  1B 97                   DB 27,16+80H+7;H=7‡‡
2C92  52 4C 43 C1             DB 'RLC',AA
2C96  52 52 43 C1             DB 'RRC',AA
2C9A  52 4C C1                DB 'RL',AA
2C9D  52 52 C1                DB 'RR',AA
2CA0  44 41 C1                DB 'DA',AA
2CA3  43 50 CC                DB 'CP',LL
2CA6  53 43 C6                DB 'SC',FF
2CA9  43 43 C6                DB 'CC',FF
```

```
                        ;
2CAC  09 4C C4          COF1:   DB 8+1,'L',DD
2CAF  20                        DB 32+0
2CB0  08 AC                     DB 8,CO
2CB2  A8                        DB 40+80H+0
                        ;
2CB3  25 A8             COF2:   DB 32+5,40+0+80H
                        ;
2CB5  00 05 09          COF3:   DB 0,5,8+1;H=0‡‡
2CB8  52 45 D4                  DB 'RE',TT
2CBB  A4                        DB 32+4+80H
2CBC  1B 38 05                  DB 27,56,5;H=1‡‡+
2CBF  09                        DB 8+1
2CC0  50 4F D0                  DB 'PO',PP
2CC3  A2                        DB 32+2+80H
2CC4  93                        DB 16+3+80H
2CC5  52 45 D4                  DB 'RE',TT
2CC8  45 58 DB                  DB 'EX',XX
2CCB  4A 50 20 28               DC 'JP (d)'
2CCF  64 A9
2CD1  4C 44 20 53               DC 'LD SP,d'
2CD5  50 2C E4
2CD8  07 09 4A D0               DB 7,8+1,'J',PP;H=2‡‡
2CDC  24 88                     DB 32+4,8+80H
2CDE  2C E2                     DC ',b'
2CE0  2C 97                     DB 44,16+7+80H;H=3‡‡
2CE2  4A 50 20 E2               DC 'JP b'
2CE6  80                        DB 80H;NOTHING
2CE7  4F 55 54 20               DC 'OUT (a),A'
2CEB  28 61 29 2C
2CEF  C1
2CF0  49 4E 20 41               DC 'IN A,(a)'
2CF4  2C 28 61 A9
2CF8  45 5B 20 28               DC 'EX (SP),d'
2CFC  53 50 29 2C
2D00  E4
2D01  45 58 20 44               DB 'EX DE,H',LL
2D05  45 2C 48 CC
2D09  44 C9                     DB 'D',II
2D0B  45 C9                     DB 'E',II
2D0D  09 09                     DB 9,8+1 ;H=4‡‡
2D0F  43 41 4C CC               DB 'CAL',LL
2D13  24                        DB 32+4
2D14  88 2C E2                  DB 8+80H,',',QQ
2D17  0F 38 06                  DB 15,56,6 ;H=5‡‡
2D1A  09 50 55 53               DB 8+1,'PUS',HH
2D1E  C8
2D1F  A2                        DB 32+2+80H
2D20  88 43 41 4C               DB 8+80H,'CALL ',QQ
2D24  4C 20 E2
2D27  03 25                     DB 3,32+5;H=6‡‡
2D29  88 E1                     DB 8+80H,VV
2D2B  15 09                     DB 21,8+1       ;H=7‡‡
2D2D  52 53 D4                  DB 'RS',TT
2D30  97                        DB 16+7+80H
2D31  B0                        DC '0'
```

```
2D32  B8                      DC '8'
2D33  31 B0                   DC '10'
2D35  31 B8                   DC '18'
2D37  32 B0                   DC '20'
2D39  32 B8                   DC '28'
2D3B  33 B0                   DC '30'
2D3D  33 B8                   DC '38'

                              ;
                              ;
2D3F  17                      C1F0:   DB 16+7
2D40  52 4C C3                        DB 'RL',CC
2D43  52 52 C3                        DB 'RR',CC
2D46  52 CC                           DB 'R',LL
2D48  52 D2                           DB 'R',RR
2D4A  53 4C C1                        DB 'SL',AA
2D4D  53 52 C1                        DB 'SR',AA
2D50  80                              DB 80H;NOTHING
2D51  53 52 CC                        DB 'SR',LL
2D54  08 A0                           DB 8,SPA
2D56  A8                              DB 40+80H+0

                              ;
2D57  09                      C1F1:   DB 8+1
2D58  42 49 D4                        DB 'BI',TT
2D5B  23 A8                           DB 32+3,40+80H+0

                              ;
2D5D  09                      C1F2:   DB 8+1
2D5E  52 45 D3                        DB 'RE',SS
2D61  23 A8                           DB 32+3,40+80H+0

                              ;
2D63  09                      C1F3:   DB 8+1
2D64  53 45 D4                        DB 'SE',TT
2D67  23 A8                           DB 32+3,40+80H+0

                              ;
                              ;
2D69  88 80                   C2F0:   DB 8+80H,80H

                              ;
2D6B  00 09 09                C2F1:   DB 0,9,8+1;H=0**
2D6E  49 CE                           DB 'I',NN
2D70  20                              DB 32+0
2D71  88                              DB 8+80H
2D72  2C 28 43 A9                     DB ',(C',BR
2D76  0A 08                           DB 10,8 ;H=1**
2D78  4F 55 54 20             DB 'OUT (C)',CO
2D7C  28 43 29 AC
2D80  A0
2D81  12 38 08                DB 32+0+80H
2D84  08                      DB 18,56,8 ;H=2**
2D85  53 42 43 20             DB 8
2D89  64 AC                   DC 'SBC d,'
2D8B  A1
2D8C  08                      DB 32+1+80H
2D8D  41 44 43 20             DB 8
2D91  64 AC                   DC 'ADC d,'
2D93  A1
2D94  14 38 09                DB 32+1+80H
2D97  08                      DB 20,56,9 ;H=3**
                              DB 8
```

```
2D98  4C 44 20 28          DC 'LD (b),'
2D9C  62 29 AC
2D9F  A1                   DB 32+80H+1
2DA0  09 4C C4             DB 8+1,'L',DD
2DA3  21 88                DB 32+1,8+80H
2DA5  2C 28 62 A9          DC ',(b)'
2DA9  04 88                DB 4,8+80H ;H=4**
2DAB  4E 45 C7             DB 'NE',66
2DAE  0C 38 05             DB 12,56,5;H=5**
2DB1  88                   DB 8+80H
2DB2  52 45 54 CE          DB 'RET',NN
2DB6  88                   DB 8+80H
2DB7  52 45 54 C9          DB 'RET',II
2DBB  0C 09                DB 12,8+1
2DBD  49 CD                DC 'IM'
2DBF  97                   DB 16+7+80H;H=6**
2DC0  B0                   DC '0'
2DC1  80                   DB 80H
2DC2  B1                   DC '1'
2DC3  B2                   DC '2'
2DC4  80 80 80 80          DB 80H,80H,80H,80H
2DC8  1F 97                DB 31,16+7+80H;H=7**
2DCA  4C 44 20 49          DB 'LD I,',AA
2DCE  2C C1
2DD0  4C 44 20 52          DB 'LD R,',AA
2DD4  2C C1
2DD6  4C 44 20 41          DB 'LD A,',II
2DDA  2C C9
2DDC  4C 44 20 41          DB 'LD A,',RR
2DE0  2C D2
2DE2  52 52 C4             DB 'RR',DD
2DE5  52 4C C4             DB 'RL',DD
2DE8  80                   DB 80H
2DE9  80                   DB 80H
                           ;
2DEA  1F                   C2F2:  DB 24+7
2DEB  4C C4                       DB 'L',DD
2DED  43 D0                       DB 'C',PP
2DEF  49 CE                       DB 'I',NN
2DF1  4F D4                       DB 'O',TT
2DF3  80 80 80 80                 DB 80H,80H,80H,80H
2DF7  97                          DB 16+80H+7
2DF8  80 80 80 80                 DB 80H,80H,80H,80H
2DFC  C9                          DB II
2DFD  C4                          DB DD
2DFE  49 D2                       DB 'I',RR
2E00  44 D2                       DB 'D',RR
                           ;
2E02  88 80                C2F3:  8+80H,80H
                           ;
                           ;
2E04  08 C2 C3 C4          TABLE2: DEFB 8,BB,CC,DD
2E08  C5 C8 CC E5                  DB EE,HH,LL,'e'+80H,AA
2E0C  C1
2E0D  07 42 C3 44                  DB 7,'B',CC,'D'
2E11  C5 E4 53 D0                  DB EE,@@,'S',PP
```

```
2E15  07 42 C3 44
2E19  C5 E4 41 C6
2E1D  10
2E1E  30 AC
2E20  31 AC
2E22  32 AC
2E24  33 AC
2E26  34 AC
2E28  35 AC
2E2A  36 AC
2E2C  37 AC
2E2E  0C 4E DA DA
2E32  4E C3 C3 50
2E36  CF 50 C5 D0
2E3A  CD
2E3B  23 41 44 44
2E3F  20 41 AC
2E42  41 44 43 20
2E46  41 AC
2E48  53 55 42 A0
2E4C  53 42 43 20
2E50  41 AC
2E52  41 4E 44 A0
2E56  58 4F 52 A0
2E5A  4F 52 A0
2E5D  43 50 A0
2E60  80
2E61  48 CC
2E63  49 D8
2E65  49 D9
2E67  48 41 4C D4
```

```
DB 7,'B',CC,'D'
DB EE,@@,'A',FF
DB 16
DB '0',CO
DB '1',CO
DB '2',CO
DB '3',CO
DB '4',CO
DB '5',CO
DB '6',CO
DB '7',CO
DB 12,'N',ZZ,ZZ
DB 'N',CC,CC,'P'
DB OO,'P',EE,PP,MM

DB 35,'ADD A',CO

DB 'ADC A',CO

DB 'SUB',SPA
DB 'SBC A',CO

DB 'AND',SPA
DB 'XOR',SPA
DB 'OR',SPA
DB 'CP',SPA
DB 80H
DB 'H',LL
DB 'I',XX
DB 'I',YY
ADHALT: DB 'HAL',TT
;
;
;
END
```

2 BC, DE, d, AF
2 bytes long

3 '1', to '7,'

16 bytes long

4 NZ, Z, NC, C, PO, PE, P, M
12 bytes long

5 ADD A,
ADC A,
SUB
SBC A,
AND
XOR
OR
CP
35 bytes long

6 HL, IX, IY

End of label

Macros:

Symbols:

@@	00E4	AA	00C1	ABS	0E56	ABS1BC	1F92
ABSBC	1F99	ACC1	FDCC	ACCSTA	128F	ACON	1FAB
ADD	0E86	ADD1	212C	ADD8	2127	ADDBR	25FC
ADDHL	29AA	ADDHL4	29A8	ADHALT	2E67	ADJDES	35F9
ADJVAL	0C4F	ADLABE	FDBE	ADLO	2A9E	ADVANC	23F9
AE	3C45	AF1	2954	AHLIX	293D	AKSKIP	2848
AL1	1D63	ALIST	1FA5I	ALISTC	2865	ALISTG	2859
ALISTH	285F	ALIT	28DD	ALL1	1A7D	ALLCAP	1A7E
ALOP1	13F3	ALPHA	3B87	ANDACC	209A	ANORM	294E
APROG	FDB7	ARRTOP	FACC	AS1PLI	281D	ASBYTE	FDA8
ASC	3F4B	ASELEC	28B0	ASEL6	288A	ASELH	2803
ASELHI	2890	ASEMBL	2669	ASKI	282F	ASKIPA	283D
ASPLIT	2819	ASRCPO	260F	ASTACK	FD6D	ATN	241D
ATNT2	2452	AUTOCT	FD7B	AUTOIN	FD77	AUTOST	FD79
BASBOT	FAAA	BASELI	FACF	BASEM	FDC7	BASENT	025D
BASIC	0259	BASIC0	0244	BASIC1	0244	BASIC2	0250
BASIC4	024A	BASICS	0205	BASLNP	FAD1	BASTOP	FAA7
BASTPO	FAAC	BASTPG	FAA9	BAUDST	0D3C	BB	00C2
BCDEC	0DD0	BEDIT	0000	BEENIN	2200	BEGIN	1C33I
BELL	0953	BIGF1	1F06	BIGF2	1F05	BIGL	1EE7
BIGL1	1EE6	BIND1	0DE4	BINTOA	20FD	BLESS4	283A
BLOCK	35C9	BLOOP	2245	BNOINP	2204	BR	00A9
BREAIO	0CF3	BREAK	FD81	BREAKK	09B0	BREAKM	09F2
BRKBIT	0002	BS	0008	BSLOAD	3A6F	BSSTR	FE3F
BSTPHL	0679	BYTNTF	2791	C0F0	2C0A	C0F1	2CAC
C0F2	2CB3	C0F3	2CB5	C1F0	2D3F	C1F1	2D57
C1F2	2D5D	C1F3	2D63	C2F0	2D69	C2F1	2D6B
C2F2	2DEA	C2F3	2E02	CALCB0	FA7F	CALCST	FA81
CALLEN	1B43I	CALTAB	2542	CANA	1DDD	CAPITA	39B0
CARAC	3D0B	CASBAU	FD5F	CBED	251F	CC	00C3
CHAN	FE14	CHANME	23C1	CHARAC	2A3E	CHECKE	1E4E
CHKS30	3296	CHKST1	32A1	CHKSTK	3299	CHOUT	1B6AI
CHPTR	FF59	CHR$	0A3A	CHR0	0A43	CHR1	0A41
CHRDS	153A	CINP	23D4	CINPUT	23D8	CLEARL	2140
CLOCK	FD57	CLSE	0008	CNTLC	0003	CO	00AC
CODE	00C2	CODETE	1BE3I	COM1	28E4	COMAD	FDBC
COMMAN	FD82	COMMEN	FDBB	COMP$	3F5B	COMP5	3FBA
COMPAC	0FEA	COMPAO	115D	COMPAR	1163	COMPBC	11EC
COMPST	1BC4I	CONLIS	2881	CONTAD	FD6A	CONTCO	1BC5
CONTFL	FD69	CONTIN	0020	CONTPG	FD6C	COPY	FDDE
COPYAC	12F3	COPYDO	04B0	CORECS	0EE3	COSINE	137D
COUNTH	1A70	CR	000D	CREAD	2AD7	CRELAB	2AD3
CRLF	001B	CRNTPG	FAD3	CTC3	000A	CTRBAD	FA52
CTYSLT	FB45	CURLAB	FDC9	CURSCR	FF5B	CURSOF	001F
CURSON	001E	D1DELE	20C5	DADD	FDA6	DASH	0027
DATA	2073	DATA1	206F	DATAAD	FB46	DATALA	FDC3
DATAPG	FB48	DBLABE	FDC6	DBYTE	FDAA	DD	00C4
DDBCM	05D1	DDELET	20C3	DDFD	2508	DECPG	06ED
DEFLEN	27DF	DEFSS	2735	DEFWW	2717	DEHLRS	0008
DELET1	06CF	DELETE	2004	DELSPR	FE52	DESAVE	FB49
DESTAC	1274	DETPN	06E7	DETYPE	FDAF	DGOTIX	25D9
DIGIT	3B96	DIROUT	08EB	DISAD	FDB1	DISAM	24EB
DISASC	213C	DISEXX	2132	DISPC	214B	DIV10	0E3D
DIVBY1	0E28	DIVIDE	109E	DJVL4	0C9D	DLSPNO	FE4F

DOADDS	2A9B	DOBACK	1BA9	DOBASI	2118	DOCLF	1FF9
DOCOM	2B36	DOCOMP	1B49	DODEFB	29D1	DODIM	331F
DODIS	2482	DOESDE	2ABE	DOINV	1DCF	DOJUMP	3E38
DOLIN2	2293	DOLIST	1D2B	DOLLL	27F3	DONEXI	2770
DONIB	0FBE	DONORM	2632	DONQ	2257	DONTDE	1FA2
DONUM	1ACD	DOSH	2B31	DOWW	2269	DPROG	FDB3
DTWOBY	2639	DTYPE	FDB0	EDEND	1448	EDENT	02D9
EDIT	FDAD	EDIT1	1424	EDITOR	1429	EE	00C5
EFFENT	02CB	EI6	00B8	ENDEL	203C	ENDLAB	29FB
ENDLIN	2CFE	ENDLLL	27EC	ENDTAB	FDB9	ENLIST	1FC2
ENOU	1DEB	ENSING	218C	ENTER3	1C03I	ENTERI	1C11I
ENUMBE	2499	EOL	0005	EQUAL	20C0	ERR34	197B
ERROR	18AF	ERRPOS	FD84	EVALAB	3D84	EVALFN	3E5B
EVALHI	2302	EVALL	1F66	EVALSE	3E7E	EXASSE	1E97
EXCHAN	1183	EXDE	2ACF	EXDOT	2993	EXIT	1A5E
EXP	23EC	EXP01	1372	EXP2X	23F2	EXTEST	2A99
FBE1	29A5	FBEGIN	1C33	FEXPAN	FA9E	FF	00C6
FIND	19DDI	FIND$	3FEC	FIND1$	3FE9	FINDEN	1CB9
FINDEX	2589	FINDJP	0689	FINDLE	316E	FINDLI	061A
FINDNO	2E39	FINDNS	1A19I	FINDRE	1A3B	FINFF	0BFE
FINPOI	18D7	FINTAB	2471	FLAGS	0000	FLAGS1	FD86
FLAGTA	2BDA	FNDBYT	2794	FORCOU	FB44	FORWAR	0019
FOUN	2970	FOUND	2885	FRAC	112B	FREESP	033C
FREQ	FE16	GASH	FE02	GET1HL	29CA	GET2HL	29C7
GET5VA	351A	GETBAS	0BC5	GETCHA	1B23	GETDE1	2350I
GETHL	29CC	GETHL1	20BD	GETLIN	06DB	GETLN1	06DE
GETLNE	19E5I	GETNEX	24CC	GETNVA	351D	GETREA	2702
GETSPA	2FE9	GETSTR	00BC	GETVAL	3066	GG	00C7
GOPTR	FB41	GOSNUM	FB43	GOSTAC	FAD8	GOTBRH	25E1
GOTCAL	21B7	GOTDEC	1ADC	GOTDEF	1FFF	GOTHLI	2918
GOTLIN	1CEE	GOTW	24C6	GOTW1	24C4	GOTZER	10A6
GREATE	20C8	GREEQU	20DB	GRRDS	1541	GST5VA	3505
GSTNVA	3508	HEXL	2A05	HH	00C8	HIGH	0A8C
HIPUT	1B2F	HIRET	1E15	HLBEYO	1C80	II	00C9
IJINIT	097F	IJTABL	FFF0	IMOVE	1C73I	INBIT	0A6B
INBLK2	0ACB	INBLOC	0ABA	INBYTE	0A9A	INCA	1CBB
INCLRA	35AC	INCPG	06F3	INCREG	2143	INDEX	FDA8
INDEXL	FDC1	INHEX	2988	INKEY$	3ECE	INOUT	0AAE
INSERT	02FE	INSTR	1B85I	INSTR1	1B6F	INSTR2	1B87
INT	200A	INT125	0996	INT64K	121D	INTEGE	119D
INTFFF	FD5E	INTTAB	FDF2	IOPL	FA8F	IOPR	FD76
ISACII	19D7	ISDFG	24A9	ITYPE	FD87	JNOINP	21DB
JPHL	07B8	JROUT	22EB	JT	0028	JUMKEY	20F8
JUMP	21CE	JUMROT	20CE	KBD	0079	KBDBUF	FA83
KBDPO	3618	KBFLAG	FA91	KEY1	3622	KEYROT	20E8
KILLIN	098A	KILLVA	27FB	KILSND	093A	KSKIP	255E
LABTAB	FDB5	LABTOP	1FD6	LARGE1	11C5	LARGEN	11C8
LASTAS	FD7D	LASTDR	FD7E	LASTKY	FD7C	LDBSBT	06AC
LDEL	203A	LEFT$	3F20	LEN	3F95	LENGTH	FDAE
LENHI	FF57	LENLO	0FF5	LESEQU	20E2	LESSLI	2281
LESSTH	1A4E	LEXFUN	2680	LF	000A	LINEIN	1E21
LINK	2277	LINKDE	223C	LINKER	FDAC	LIST	21EF
LISTG	2575	LISTH	2578	LISTL	2579	LISTLA	1FCF
LIT1	25B4	LITERA	25B2	LKHEX	1A2A	LKNUM	2A29
LL	00CC	LOAD	0B40	LOAD2	089E	LOADER	0891
LOADSN	2D0C	LOCAR	2A31	LOG	2398	LOG2X	23A5

LOGE2	2494	LOGNX	23BC	LOKNUM	2A24	LOKSTR	2A0D
LOOK1S	2A50	LOOKNU	27FB	LOOPAD	29DF	LOOPSR	29BA
LOOPTE	2A90	LOOPUP	2B6B	LOPPP	247D	LOPTAK	2079
LOW	0A82	LOWCAL	1F42	LRATOV	3593	LSPOP	20A0
LSTPG	FA7A	MA1	1AFE	MAF	FD91	MAFD	FDB9
MAINSC	2148	MAKESP	0351	MAST1	1B01	MASTAS	2809
MASTER	1B09	MASTRA	254E	MATABL	2BC5	MBC	FD93
MBCD	FD8B	MDE	FD95	MDED	FD8D	MEM1	FDD9
MEMPOI	FDA1	MEMREP	2419	MFOUN	2963	MHL	FD97
MHLD	FD8F	MID$	3EDD	MIDVAL	FD60	MISS	2601
MISS1	2602	MISSON	2211	MIX	FD99	MIY	FD9B
MM	00CD	MORC	2A35	MOREHE	297E1	MORLLI	227F
MOVADR	1A04	MOVDEO	12B9	MOVE	1C60	MOVEAC	224C
MOVEDE	12A7	MOVEST	29F4	MOVHL	1BDDI	MOVPOI	2171
MOVQUO	1A95	MOVTES	2A52	MPC	FD9F	MSP	FD9D
MULPLY	100B	MULTDE	1F12	MVDIST	FE4D	MVNO	FE51
NBTOP	FAA4	NBTPG	FAA6	NCP1	06F7	NEGACC	0F13
NEWENT	0221	NEWLIN	2BC9	NEWMEM	2413	NEWSCR	2F90
NEXT	2134	NN	00CE	NOADD	256B	NOADLA	2468
NOAPD	2033	NOAS11	1D48	NOASCI	19D5	NOASS	1CF7
NOASS1	1D36	NOASS2	1D42	NOASS3	1D55	NOASS4	1D4A
NOBOTH	2977	NOC	19FB	NOC1	1F64	NOCOME	268C
NOCOMP	1BD2	NODATA	2946	NODBYT	1F35	NODDY	2D7C
NODELE	1DCC	NODF	2752	NODLOC	FA9B	NOERRO	1D67
NOGOOD	02C0	NOHLS	1BDD	NOIND	27A0	NOIX	20A3
NOIX1	209F	NOIX2	20A6	NOIXD	27D6	NOIXY1	206C
NOLABE	26F1	NOPUT	1B35	NOQUO	1ABC	NOREL	1FE8
NORETD	248E	NORM	2522	NORST	21E8	NOSETU	1BEA
NOSPR	FE4E	NOT2BY	2907	NOTACC	20AF	NOTAED	27B9
NOTBOT	2163	NOTCB	27C5	NOTCP	20DA	NOTD	2393
NOTDD	2643	NOTDDF	2516	NOTDEL	26B4	NOTEQU	20E9
NOTESE	28D2	NOTF1	2974	NOTF2	2975	NOTFOU	2973
NOTHAL	2502	NOTINS	1C94	NOTPC	243C	NOTREG	236C
NOTSD	2646	NOTSET	2378	NOTW	25FE	NOUP1	2B9B
NOUPDA	2B79	NTAKOU	2B8B	OFREQ	08C6	OL1	24F3
ONE	00B1	ONEDI	2479	ONELIN	29AF	ONESP	399E
ONESPC	1A23	ONEVAL	1A56	ONOUT	2341	OO	00CF
OP11	FDD2	OP1STA	129E	OPS	2672	OPTABL	0D72
ORACC	20A6	OTBLK2	0AFC	OUT4RE	239B	OUTBIT	0A7B
OUTBLO	0AE6	OUTBYT	0AA4	OUTDEF	24DC	OUTDEX	2658
OUTEX	2656	OUTFLA	2359	OUTMEM	2400	OUTRAN	1219
OUTRDE	2331	OUTREG	2359	OUTSP	39C1	OUTSTR	1907
OUTVBC	1B50	OVERFL	121B	OVERHL	3B73	OVERSP	3B74
OVOL	08F9	OVRLAY	FFED	P1JTBL	2E6B	PAGE	FAD2
PAU1	2BE8	PGCHK	0671	PGCHK1	066E	PGN1	FAD4
PGN2	FAD5	PGSLC1	0647	PGSLC2	0644	PGSLCT	064A
PGTOP	FAD6	PIOP1	120B	PLSPND	FE50	POHL	2AA5
POINLA	2AFC	POINTE	FDA5	POLY2	13C1	POLYNO	13D2
POWER	233F	PP	00D0	PR3	24CE	PRCOM	1EA9
PRDEFB	249D	PRDL	24D9	PRE	1DA5	PRE1	1DAB
PRINDF	1EB7	PRINTX	0CAB	PRORPL	FD75	PROUTR	2326
PRUNDE	2313	PSHNAM	3037	PST5VA	3534	PSTNVA	3537
PUT1BC	22A1	PUT2BC	22C0	PUT2JR	22E8	PUT2L	22C3
PUT5VA	3544	PUTINL	1F24	PUTJRB	22CE	PUTJRL	22D1
PUTLAB	2B4C	PUTNVA	3547	PUTOK	22B7	PUTOK1	22B6
PUTSTA	2FCD	QOL	224C	QQ	00E2	QU1HL	1F09

QUHL	1F0F	RAMPGC	34FE	RAND1	2BD7	READ1H	1AD3
READDE	1E80	READHE	1AE0	READNU	1AB1	REALBY	FA90
REENTE	1C4D	RELBC	1AA1	RELDE	2BBA	REM1	20AD
REM2BY	20A9	REMAIN	0FFA	REPDEC	2316	REPFLA	2371
RESINT	0B06	RESTEN	2C8D	RETACC	12D1	RETINP	1B7D
RETSAV	FD61	RETT	07B5	RICHJL	FD4B	RIGHT$	3F1A
RMTPHL	0697	RND	2048	RNSEED	FD7F	ROAHL	0FA1
ROTR	0FBD	ROUNAC	0F2B	RR	00D2	RSTD01	00FF
RUN4	2CFD	RUN5	2CF0	RUN55	2CDB	RUN8	2CEC
RWTAB	26F7	SAANGL	15FC	SADJSP	15BB	SAFB	2264
SAHALT	27CD	SARC	16CE	SATTR	15E2	SAUTO	0A07
SAVE	0B90	SAVEAC	12BF	SAVEAD	2122	SAVEMH	2123
SAVLHL	1A09	SBAUD	0D38	SCENT	1876	SCENT4	188F
SCENTR	00CB	SCIRCL	180F	SCLS	1595	SCODE	00DA
SCOLOU	15B3	SCRN0	FF5D	SCRN1	FF6C	SCRN2	FF7B
SCRN3	FF8A	SCRN4	FF99	SCRN5	FFA8	SCRN6	FFB7
SCRN7	FFC6	SCRRST	0010	SCRVS	15EA	SCTLSP	159B
SCURS0	1583	SDISP	0005	SDRAW	1687	SDSI	157C
SE	3C91	SECTAB	2BD0	SEDIT	0007	SELECG	259A
SELECH	259D	SELECL	259E	SEQ0	3EA1	SET1AC	2000
SETA7	265E	SETB	28A9	SETCAL	FD48	SETCHL	1AF5
SETINI	27B7	SETINT	0B10	SETNZ	2A63	SETZER	10A9
SGENPA	15A3	SHIF	00AD	SIGN	FDD8	SIGNAC	208F
SIMPL	221F	SIMPLE	229C	SINCOS	161F	SINE	13B2
SING	2183	SING1	2183	SINK	1590	SKIP	256F
SKIPHC	2560	SKIPHL	2561	SLEB	27E6	SLET	29DA
SLINE	1574	SLIST	0000	SMALL1	11B9	SMALLE	11BB
SMBYTE	FF55	SMVSPR	15D1	SNDBUF	27B3	SNEW	0205
SOUT	09FD	SPA	00A0	SPAPER	158A	SPAUSE	2BE4
SPBASE	FE4C	SPDEFB	1E27	SPEED	FE4B	SPLIT	258F
SPLIT1	2590	SPLOT	1670	SPRINT	2B88	SPRTBL	FE55
SQROOT	236E	SQUA1	2391	SQUARE	1008	SQUASH	0A15
SRANGL	160A	SRCADD	29B6	SRCPOI	29DA	SRCQUO	2A1A
SRE6	0001	SREM	27AA	SROM	009E	SRUN	2CAF
SS	00D3	SSOUND	2D0C	SSPLIT	2592	SSPRIT	15C4
SSTACK	FA96	STA	1311	STAC	1317	STACCT	FD73
STACK	FD48	STACKA	1286	STACKO	1298	STACKS	0744
STAHLH	0F58	STALI	286F	START	FB4B	STASEM	1CC3I
STBC	1348	STHLHL	136B	STKLEN	315B	STKLIM	FA92
STKTES	3EC9	STKTOP	FA7A	STLIST	1FDB	STOP	2D62
STOPPP	2D56	STR$	3FC6	STREND	1AC8	STRHL	02E7
STRJP	0DBE	STRLP	1ABA	STROUT	02EB	STRPOI	29E1
STRSTA	3FCB	STRTSR	29BC	SUB1	212F	SUB8	211F
SUBTRA	0EB3	SVS	15F6	SWINDO	15DA	SWITCH	00F4
SYFAIL	3C40	SYN	398C	SYNPRO	14B9	SYSTOP	FA94
SYSUCC	3C3B	TAB	0009	TABBYT	2BF2	TABLE2	2E04
TAKEPO	2057	TAKOUT	2B9F	TDELET	2A65	TDIS	220D
TEMP	FE04	TERL	1E43	TERL1	1E31	TERL2	1E3A
TERMIN	1CAE	TEST1	1A0CI	TEST2B	2A8DI	TESTAD	2AA8
TESTBC	1C9F	TESTCU	1A0F	TESTDA	28F6	TESTEN	1A12
TESTHL	1BB6I	TESTL1	26F0	TESTLA	2A4C	TESTPR	1EBF
TESTSP	0344	THER4	28CB	THER8	28CD	TIME$	3F3B
TINSER	2A7AI	TMPA	FD71	TMPHL	FD6F	TOKMIN	124F
TOOMPO	2B1D	TOPAS	2779	TOPFIN	19DE	TOPIN	1B8F
TOPM	1B94	TOPNUM	1AE7	TOPREG	239E	TOPSPL	2593
TPCH1	06BB	TRACE	21A9	TRY23	2996	TT	00D4

```
TWO     00B2    TYPE    FD68    TYPTBL  FFD5    UNSTAC  1F54
UPDATE  3182    UPDL    2B67    UPDLIN  22F6    UPDX    2B6D
USER    FA89    USERIN  FA98    USERIO  FD51    USERNO  FAA1
USERRO  FD54    USR     203D    USRRST  FD4E    UVITOS  34A0
VAL     3F9E    VALBOT  FA7D    VALLAB  2B21    VALRA1  3580
VARNAM  FA7B    VATOLR  357C    VAZERO  FD65    VCOUNT  FE53
VDINIT  2E85    VDPSTS  FE54    VERIF   FD67    VHEX    1B55
VHINIB  1B5C    VINT    3A55    VINTFG  FF58    VLESS   1B68
VLONIB  1B60    VOL     FE18    VOUTHC  19C6    VV      00E1
WCHJUM  FDA3    WHAT    1D67    WHAT1   1DB6    WHAT11  1DCA
WHAT3   1F1D    WKAREA  FE1A    WORDLS  253B    XX      00DB
YORN    FDD7    YY      00D9    ZER     00B0    ZERACC  0FAF
ZERONV  355D    ZZ      00DA
```

No Fatal error(s)

A>