

Language Forth Z80FIG-FORTH1_1G_FILES

Z80FORTH

TITLE < Z80 fig-FORTH 1.1 g >

SUBTTL Adaptive version by EHR

```
;
;
; Modified frm FIG document keyed by Dennis L. Wilson 800907
; Converted frm "8080 FIG-FORTH VERSION A0 15SEP79"
;
; fig-FORTH release 1.1 for the 8080 processor.
;
; ALL PUBLICATIONS OF THE FORTH INTEREST GROUP
; ARE PUBLIC DOMAIN. THEY MAY BE FURTHER
; DISTRIBUTED BY THE INCLUSION OF THIS CREDIT NOTICE:
;
; This publication has been made available by the
;   Forth Interest Group
;   P.O.Box 1105
;   San Carlos, CA 94070
;   U.S.A.
;
; Implementation on 8080 by:
;   John Cassady
;   339 15th Street
;   Oakland, CA 94612
;   U.S.A
;   on 790528
; Modified by:
;   Kim Harris
; Acknowledgements:
;   George Flammer
;   Robt. D. Villwock
; -----
; Z80 Version for Cromemco CDOS & Digital Research CP/M by:
;   Dennis Lee Wilson c/o
;   Aristotelian Logicians
;   2631 East Pinchot Avenue
;   Phoenix, AZ 85016
;   U.S.A.
; -----
; The 2 byte Z80 code for Jump Relative (JR) has been substituted for
; the 3 byte Jump (JP) wherever practical. The port I/O words P@ & P!
; have been made ROMable by use of Z80 instructions.
; -----
; Further modifications (marked ;/) by:
;   Edmund Ramm
;   Anderheitsallee 24
;   2000 Hamburg 71
;   Fed. Rep. of Germany      840418
; -----
; Disc I/O has been modified a la Albert van der Horst (HCCH) to employ
; CP/M 2.x's random access feature.
; -----
;
; Release & Version numbers
;
; FIGREL EQU      1          ;FIG RELEASE #
; FIGREV EQU      1          ;FIG REVISION #
```

;

```

; * FOLLOWING USED BY COLD; MUST BE IN SAME ORDER AS USER VARIABLES *
;
S0INIT: DEFW      0          ;/ INIT (S0)
R0INIT: DEFW      0          ;/ INIT (R0)
TIBINI: DEFW      0          ;/ INIT (TIB)
      DEFW      1FH          ; INIT (WIDTH)
      DEFW      0           ; INIT (WARNING)
      DEFW      INITDP       ; INIT (FENCE)
      DEFW      INITDP       ; INIT (DP)
      DEFW      FORTH+8      ; INIT (VOC-LINK)
;
; * END DATA USED BY COLD *
;
      DEFW      0H,0B250H      ;Z80 CPU NAME (HW,LW)
                                ;(32 BIT BASE 36 INTEGER)
      FORM
      REGISTERS
;
;
;      FORTH      Z80      FORTH PRESERVATION RULES
;      ----      ---      -----
;      IP         BC       should be preserved
;                               accross FORTH words.
;      W          DE       sometimes output from
;                               NEXT, may be altered
;                               b4 JP'ing to NEXT,
;                               input only when
;                               "DPUSH" called.
;      SP         SP       should be used only as
;                               Data Stack accross
;                               FORTH words, may be
;                               used within FORTH
;                               words if restored
;                               b4 "NEXT"
;                               HL       Never output frm NEXT
;                               input only when
;                               "HPUSH" called
;
;
;
UP:    DEFW      0          ;/ USER AREA PTR
RPP:   DEFW      0          ;/ RETURN STACK PTR
BUF1:  DEFW      0          ;/ address of 1st disc buffer
;
;
;      COMMENT CONVENTIONS:
;
;      =          MEANS "IS EQUAL TO"
;      <--        MEANS ASSIGNMENT
;      NAME      =      ADDR OF NAME
;      (NAME)    =      CONTENTS @ NAME
;      ((NAME)) =      INDIRECT CONTENTS
;      CFA      =      CODE FIELD ADDR
;      LFA      =      LINK FIELD ADDR
;      NFA      =      NAME FIELD ADDR
;      PFA      =      PARAMETER FIELD ADDR
;      S1       =      ADDR OF 1st WORD OF PARAMETER STACK
;      S2       =      "- OF 2nd -" OF  "- -"
;      R1       =      "- OF 1st -" OF RETURN STACK
;      R2       =      "- OF 2nd -" OF  "- -"
; ( above Stack posn. valid b4 & after execution of any word, not during)
;
;      LSB      =      LEAST SIGNIFICANT BIT
;      MSB      =      MOST SIGNIFICANT BIT
;      LB       =      LOW BYTE
;      HB       =      HIGH BYTE

```

```

;      LW      =      LOW WORD
;      HW      =      HIGH WORD
; (May be used as suffix to above names)
FORM
;      FORTH ADDRESS INTERPRETER
;      POST INCREMENTING VERSION
;
DPUSH:  PUSH    DE
HPUSH:  PUSH    HL      ;      IY points here
NEXT:   LD      A,(BC)   ; (W)<--((IP))  IX points here
        LD      L,A
        INC     BC      ;INC IP
        LD      A,(BC)
        LD      H,A     ; (HL)<--CFA
        INC     BC      ;INC IP
NEXT1:  LD      E,(HL)   ; (PC)<--((W))
        INC     HL
        LD      D,(HL)
        EX      DE,HL
        JP      (HL)    ;NOTE: (DE)=CFA+1
;
JNEXT   MACRO
        JP      (IX)
        ENDM
;
JHPUSH  MACRO
        JP      (IY)
        ENDM
FORM
;      FORTH DICTIONARY
;      DICTIONARY FORMAT:
;
;      ADDRESS NAME      BYTE
;      -----
;
;      (MSB=1
;      (P=PRECEDENCE BIT
;      (S=SMUDGE BIT
;      <NAME LENGTH
;      MSB=0, NAME'S 1st CHAR
;
;      NFA      NAME FIELD      1PS<LEN>
;      0<1CHAR>
;      0<2CHAR>
;      ...
;      1<LCHAR>      MSB=1, NAME'S LAST CHAR
;      LFA      LINK FIELD      <LINKLB>
;      <LINKHB>      =PREVIOUS WORD'S NFA
;
; LABEL: CFA      CODE FIELD      <CODELB>
;      <CODEHB>      =ADDR CPU CODE
;
;      PFA      PARAMETER
;      FIELD      <1PARAM>      1st PARAMETER BYTE
;      <2PARAM>
;      ...
;
;
DP0:    DEFB     83H      ;LIT
        DM      'LIT'
        DEFW     0      ; (LFA)=0 MARKS END OF DICTIONARY
LIT:    DEFW     $+2      ; (S1)<--((IP))
        LD      A,(BC)   ; (HL)<--((IP))=LITERAL
        INC     BC      ; (IP)<--(IP)+2
        LD      L,A
        LD      A,(BC)   ;LB
        INC     BC      ;HB
        LD      H,A
        JHPUSH      ; (S1)<--(HL)

```

```

;
    DEFB    87H                ;EXECUTE
    DM      'EXECUTE'
    DEFW    LIT-6
EXEC:    DEFW    $+2
    POP     HL
    JP      NEXT1

;
    DEFB    86H                ;BRANCH
    DM      'BRANCH'
    DEFW    EXEC-0AH
BRAN:    DEFW    $+2            ;(IP)<--(IP)+((IP))
BRAN1:   LD     H,B            ;(HL)<--(IP)
    LD      L,C
    LD      E,(HL)            ;(DE)<--((IP))=BRANCH OFFSET
    INC     HL
    LD      D,(HL)
    DEC     HL
    ADD     HL,DE            ;(HL)<--(HL)+((IP))
    LD      C,L              ;(IP)<--(HL)
    LD      B,H
    JNEXT

;
    DEFB    87H                ;0BRANCH
    DM      '0BRANCH'
    DEFW    BRAN-9
ZBRAN:   DEFW    $+2
    POP     HL
    LD      A,L
    OR      H
    JR      Z,BRAN1          ;IF (S1)=0 THEN BRANCH
    INC     BC                ;ELSE SKIP BRANCH OFFSET
    INC     BC
    JNEXT

;
    DEFB    86H                ;(LOOP)
    DM      '(LOOP)'
    DEFW    ZBRAN-0AH
XLOOP:   DEFW    $+2
    LD      HL,(RPP)          ; ((HL))=INDEX=(R1)
    inc     (hl)              ;/ index(lb) += 1
    LD      E,(HL)            ;/
    INC     HL                ;/ (hl)-->index(hb)
    jr      nz,xloop1         ;/ jump if ((hl)) < 256
    inc     (hl)              ;/ else index(hb) += 1
xloop1:  LD      D,(HL)        ;/ (DE)<-- new INDEX
    INC     HL                ;/ ((HL))=LIMIT
    LD      A,E
    SUB     (HL)
    LD      A,D
    INC     HL
    SBC     A,(HL)            ; INDEX<LIMIT?
    JP      M,BRAN1           ; YES, LOOP AGAIN
    INC     HL                ; NO, DONE
    LD      (RPP),HL          ; DISCARD R1 & R2
    INC     BC
    INC     BC                ; SKIP BRANCH OFFSET
    JNEXT

;
    DEFB    87H                ;(+LOOP)
    DM      '(+LOOP)'
    DEFW    XLOOP-9
XPL00:   DEFW    $+2
    POP     DE                ;(DE)<-- INCR

```

;

```

DEFB 85H ;DIGIT
DM 'DIGIT'
DEFW ID0-4
DIGIT: DEFW $+2
POP HL ;(L)<--(S1)LB = BASE VALUE
POP DE ;(E)<--(S2)LB = ASCII CHR TO BE CONVERTED
LD A,E ;ACCU<--CHR
SUB '0' ;>=0?
JR C,DIGI2 ;/ <0 IS INVALID
CP 0AH ;>9?
JR C,DIGI1 ;/ NO, TEST BASE VALUE
SUB 07H ;GAP BETWEEN "9" & "A", NW "A"=0AH
CP 0AH ;>="A"?
JR C,DIGI2 ;/ CHRs BETWEEN "9" & "A" ARE INVALID
DIGI1: CP L ;<BASE VALUE?
JR NC,DIGI2 ;/ NO, INVALID
LD E,A ;(S2)<--(DE) = CONVERTED DIGIT
LD HL,0001H ;(S1)<--TRUE
JP DPUSH
DIGI2: LD L,H ;(HL)<--FALSE
JHPUSH ;(S1)<--FALSE
;
DEFB 86H ;(FIND) (2-1)FAILURE
DM '(FIND)' ; (2-3)SUCCESS
DEFW DIGIT-8
PFIND: DEFW $+2
POP DE ;(DE)<--NFA
PFIN1: POP HL ;(HL)<--STRING ADDR
PUSH HL ;SAVE FOR NEXT ITERATION
LD A,(DE)
XOR (HL) ;FILTER DEVIATIONS
AND 3FH ;MASK MSB & PRECEDENCE BIT
JR NZ,PFIN4 ;LENGTHS DIFFER
PFIN2: INC HL ;(HL)<--ADDR NEXT CHR IN STRING
INC DE ;(DE)<--ADDR NEXT CHR IN NF
LD A,(DE)
XOR (HL) ;FILTER DEVIATIONS
ADD A,A
JR NZ,PFIN3 ;NO MATCH
JR NC,PFIN2 ;MATCH SO FAR, LOOP AGN
LD HL,0005H ;STRING MATCHES
ADD HL,DE ;((SP))<--PFA
EX (SP),HL
PFIN6: DEC DE ;POSN DE ON NFA
LD A,(DE)
OR A ;MSB=1? =LENGTH BYTE
JP P,PFIN6 ;NO, TRY NEXT CHR
LD E,A ;(E)<--LENGTH BYTE
LD D,00H
LD HL,0001H ;(HL)<--TRUE
JP DPUSH ;NF FOUND, RETURN
;
;ABOVE NF NOT A MATCH, TRY NEXT ONE
;
PFIN3: JR C,PFIN5 ;CARRY=END OF NF
PFIN4: INC DE ;FIND END OF NF
LD A,(DE)
OR A ;MSB=1?
JP P,PFIN4 ;NO, LOOP
PFIN5: INC DE ;(DE)<--LFA
EX DE,HL
LD E,(HL)
INC HL
LD D,(HL) ;(DE)<--(LFA)

```

```

LD      A,D
OR      E                ;END OF DICTIONARY? (LFA)=0
JR      NZ,PFIN1         ;NO, TRY PREVIOUS DEFINITION
POP     HL               ;DROP STRING ADDR
LD      HL,0             ;(HL)<--FALSE
JHPUSH                ;NO MATCH FOUND, RETURN
;
DEFB    87H              ;ENCLOSE
DM      'ENCLOSE '
DEFW    PFIND-9
ENCL:   DEFW    $+2
POP     DE                ;(DE)<--(S1)=DELIMITER CHR
POP     HL                ;(HL)<--(S2)=ADDR OF TEXT TO SCAN
PUSH    HL                ;(S4)<--ADDR
LD      A,E
LD      D,A              ;(D)<--DELIM CHR
LD      E,-1             ;INIT CHR OFFSET COUNTER
DEC     HL                ;(HL)<--ADDR-1
ENCL1:  INC     HL         ;SKIP OVER LEADING DELIM CHR$
INC     E
CP      (HL)              ;DELIM CHR?
JR      Z,ENCL1           ;YES, LOOP
LD      D,0
PUSH    DE                ;(S3)<--(E)=OFFSET TO 1st NON DELIM
LD      D,A               ;(D)<--DELIM CHR
LD      A,(HL)
AND     A                 ;1st non-DELIM=NULL?
JR      NZ,ENCL2          ;NO
LD      D,0               ;YES
INC     E
PUSH    DE                ;(S2)<--OFFSET TO BYTE FOLLOWING NULL
DEC     E
PUSH    DE                ;(S1)<--OFFSET TO NULL
JNEXT
ENCL2:  LD      A,D         ;(A)<--DELIM CHR
INC     HL                ;(HL)<--ADDR NEXT CHR
INC     E                 ;(E)<--OFFSET TO NEXT CHR
CP      (HL)              ;DELIM CHR?
JR      Z,ENCL4           ;YES
LD      A,(HL)
AND     A                 ;NULL?
JR      NZ,ENCL2          ;NO, CONT SCAN
ENCL3:  LD      D,0
PUSH    DE                ;(S2)<--OFFSET TO NULL
PUSH    DE                ;(S1)<--OFFSET TO NULL
JNEXT
ENCL4:  LD      D,0
PUSH    DE                ;(S2)<--OFFSET TO BYTE FOLLOWING TEXT
INC     E
PUSH    DE                ;(S1)<--OFFSET TO 2 BYTES AFTER END OF WORD
JNEXT
;
DEFB    84H              ;EMIT
DM      'EMIT '
DEFW    ENCL-0AH
EMIT:   DEFW    DOCOL
DEFW    Pemit
DEFW    ONE,OUTT
DEFW    PSTOR,SEMIS
;
DEFB    83H              ;KEY
DM      'KEY '
DEFW    EMIT-7
KEY:    DEFW    $+2

```



```

JP      PKEY

;
DEFB    89H          ;?TERMINAL
DM      '?TERMINAL'
DEFW    KEY-6
QTERM:  DEFW    $+2
LD      HL,0
JP      PQTER

;
DEFB    82H          ;CR
DM      'CR'
DEFW    QTERM-0CH
CR:     DEFW    $+2
JP      PCR

;
DEFB    85H          ;CMOVE
DM      'CMOVE'
DEFW    CR-5
CMOVE:  DEFW    $+2
EXX
POP      BC          ; / SAVE IP
POP      DE          ; (BC)<--(S1)= #CHRs
POP      DE          ; (DE)<--(S2)= DEST ADDR
POP      HL          ; / (HL)<--(S3)= SOURCE ADDR
LD      A,B
OR      C            ; BC=0?
JR      Z,EXCMOV     ; YES, DON'T MOVE ANYTHING
LDIR
EXCMOV: EXX          ; / XFER STRING
JNEXT   ; / RESTORE IP

;
DEFB    82H          ;U* 16*16 unsigned multiply
DM      'U*'         ;994 T cycles average (8080)
DEFW    CMOVE-8
USTAR:  DEFW    $+2
POP      DE          ;(DE)<--MPLIER
POP      HL          ;(HL)<--MPCAND
PUSH     BC          ;SAVE IP
LD      B,H
LD      A,L          ;(BA)<--MPCAND
CALL     MPYX        ;(AHL)1<--MPCAND.LB*MPLIER
;                  ; 1st PARTIAL PRODUCT
PUSH     HL          ;SAVE (HL)1
LD      H,A
LD      A,B
LD      B,H          ;SAVE (A)1
CALL     MPYX        ;(AHL)2<--MPCAND.HB*MPLIER
;                  ; 2nd PARTIAL PRODUCT
POP      DE          ;(DE)<--(HL)1
LD      C,D          ;(BC)<--(AH)1

;
; FORM SUM OF PARTIALS:
;
;
;
;
;
ADD      HL,BC        ;(HL)<--(HL)2+(AH)1
ADC      A,00H        ;(AHL)<--(BA)*(DE)
LD      D,L
LD      L,H
LD      H,A          ;(HLDE)<--MPLIER*MPCAND
POP      BC          ;RESTORE IP
PUSH     DE          ;(S2)<--PRODUCT.LW
JHPUSH   ;(S1)<--PRODUCT.HW

;
; MULTIPLY PRIMITIVE

```

```

;      (AHL)<--(A)*(DE)
;      #BITS:      24      8      16
;
MPYX:  LD      HL,0          ;(HL)<--0=PARTIAL PRODUCT.LW
      LD      C,08H        ;LOOP COUNTER
MPYX1: ADD      HL,HL        ;LEFT SHIFT (AHL) 24 BITS
      RLA
      JR      NC,MPYX2      ;IF NEXT MPLIER BIT = 1
      ADD      HL,DE        ;THEN ADD MPCAND
      ADC      A,0
MPYX2: DEC      C          ;LAST MPLIER BIT?
      JR      NZ,MPYX1      ;NO, LOOP AGN
      RET                ;YES, DONE
;
      DEFB     82H          ;U/
      DM      'U/'
      DEFW     USTAR-5
USLAS: DEFW     $+2
      LD      HL,0004H
      ADD      HL,SP        ;((HL))<--NUMERATOR.LW
      LD      E,(HL)        ;(DE)<--NUMER.LW
      LD      (HL),C        ;SAVE IP ON STACK
      INC      HL
      LD      D,(HL)
      LD      (HL),B
      POP      BC          ;(BC)<--DENOMINATOR
      POP      HL          ;(HL)<--NUMER.HW
      LD      A,L
      SUB      C
      LD      A,H
      SBC      A,B          ;NUMER >= DENOM?
      JR      C,USLA1       ;NO, GO AHEAD
      LD      HL,0FFFFH     ;YES, OVERFLOW
      LD      D,H
      LD      E,L          ;/ SET REM & QUOT TO MAX
      JP      USLA7
USLA1: LD      A,10H        ;LOOP COUNTER
USLA2: ADD      HL,HL        ;LEFT SHIFT (HLDE) THRU CARRY
      RLA
      EX      DE,HL
      ADD      HL,HL
      JR      NC,USLA3
      INC      DE          ;ADD CARRY
      AND      A            ;RESET CARRY
USLA3: EX      DE,HL        ;SHIFT DONE
      RRA                ;RESTORE 1st CARRY & COUNTER
      JR      NC,USLA4      ;IF CARRY=1
      OR      A            ;/ RESET CARRY
      SBC      HL,BC        ;/ THEN (HL)<--(HL)-(BC)
      JP      USLA5
USLA4: SBC      HL,BC        ;/ (HL)<--PARTIAL REMAINDER
      JR      NC,USLA5
      ADD      HL,BC        ;UNDERFLOW, RESTORE
      DEC      DE
USLA5: INC      DE          ;INC QUOT
      DEC      A            ;COUNTER=0?
      JP      NZ,USLA2      ;NO, LOOP AGN
USLA7: POP      BC          ;RESTORE IP
      PUSH     HL           ;(S2)<--REMAINDER
      PUSH     DE           ;(S1)<--QUOTIENT
      JNEXT
;
      DEFB     83H          ;AND
      DM      'AND'

```

```

DEFW USLAS-5
ANDD: DEFW $+2 ;(S1)<--(S1) AND (S2)
      POP DE
      POP HL
      LD A,E
      AND L
      LD L,A
      LD A,D
      AND H
      LD H,A
      JHPUSH
;
      DEFB 82H ;OR
      DM 'OR'
      DEFW ANDD-6
ORR: DEFW $+2 ;(S1)<--(S1) OR (S2)
     POP DE
     POP HL
     LD A,E
     OR L
     LD L,A
     LD A,D
     OR H
     LD H,A
     JHPUSH
;
      DEFB 83H ;XOR
      DM 'XOR'
      DEFW ORR-5
XORR: DEFW $+2 ;(S1)<--(S1) XOR (S2)
     POP DE
     POP HL
     LD A,E
     XOR L
     LD L,A
     LD A,D
     XOR H
     LD H,A
     JHPUSH
;
      DEFB 83H ;SP@
      DM 'SP@'
      DEFW XORR-6
SPAT: DEFW $+2 ;(S1)<--(SP)
     LD HL,0
     ADD HL,SP ;(HL)<--(SP)
     JHPUSH
;
      DEFB 83H ;SP!
      DM 'SP!'
      DEFW SPAT-6
SPSTO: DEFW $+2 ;(SP)<--(S0) (USER VARIABLE)
      LD HL,(UP) ;(HL)<--USER VAR BASE ADDR
      LD DE,0006H
      ADD HL,DE ;(HL)<--S0
      LD E,(HL)
      INC HL
      LD D,(HL) ;(DE)<--(S0)
      EX DE,HL
      LD SP,HL ;(SP)<--(S0)
      JNEXT
;
      DEFB 83H ;RP@
      DM 'RP@'

```

```

      DEFW      SPSTO-6
RPAT:  DEFW      $+2          ;(S1)<--(RP)
      LD        HL,(RPP)
      JHPUSH
;
      DEFB      83H          ;RP!
      DM        'RP!'
      DEFW      RPAT-6
RPSTO: DEFW      $+2          ;(RP)<--(R0) (USER VARIABLE)
      LD        HL,(UP)      ;(HL)<--USER VAR BASE ADDR
      LD        DE,0008H
      ADD       HL,DE        ;(HL)<--R0
      LD        E,(HL)
      INC       HL
      LD        D,(HL)      ;(DE)<--(R0)
      LD        (RPP),DE    ;/ (RP)<--(R0)
      JNEXT
;
      DEFB      82H          ; ;S
      DM        ';S'
      DEFW      RPSTO-6
SEMIS: DEFW      $+2          ;(IP)<--(R1)
      LD        HL,(RPP)
      LD        C,(HL)
      INC       HL
      LD        B,(HL)      ;(BC)<--(R1)
      INC       HL
      LD        (RPP),HL    ;(RP)<--(RP)+2
      JNEXT
;
      DEFB      85H          ;LEAVE
      DM        'LEAVE'
      DEFW      SEMIS-5
LEAVE: DEFW      $+2          ;LIMIT<--INDEX
      LD        HL,(RPP)
      LD        E,(HL)
      INC       HL
      LD        D,(HL)      ;(DE)<--(R1)=INDEX
      INC       HL
      LD        (HL),E
      INC       HL
      LD        (HL),D      ;(R2)<--(DE)=LIMIT
      JNEXT
;
      DEFB      82H          ;>R
      DM        '>R'
      DEFW      LEAVE-8
TOR:   DEFW      $+2
      POP       DE
      LD        HL,(RPP)
      DEC       HL
      LD        (HL),D
      DEC       HL
      LD        (HL),E      ;/ (R1)<--(DE)
      LD        (RPP),HL    ; (RP)<--(RP)-2
      JNEXT
;
      DEFB      82H          ;R>
      DM        'R>'
      DEFW      TOR-5
FROMR: DEFW      $+2
      LD        HL,(RPP)
      LD        E,(HL)
      INC       HL

```

```

LD      D, (HL)
INC     HL
LD      (RPP), HL
PUSH    DE                ; (S1) <-- (R1)
JNEXT

;

DEFB    81H                ; R
DM      'R'
DEFW    FROMR-5
RR:     DEFW    ID0+2

;

DEFB    82H                ; 0=
DM      '0='
DEFW    RR-4
Z EQU:  DEFW    $+2
POP     HL
LD      A, L
OR      H
LD      HL, 0
JR      NZ, Z EQU1
INC     L                ; (HL) <-- TRUE
Z EQU1: JHPUSH

;

DEFB    82H                ; 0<
DM      '0<'
DEFW    Z EQU-5
Z LESS: DEFW    $+2
POP     AF                ; / (A) <-- (S1)H
RLA                    ; / (CARRY) <-- BIT 7
LD      HL, 0                ; (HL) <-- FALSE
JR      NC, Z LES1
INC     L                ; (HL) <-- TRUE
Z LES1: JHPUSH

;

DEFB    81H                ; +
DM      '+'
DEFW    Z LESS-5
PLUS:   DEFW    $+2
POP     DE
POP     HL
ADD     HL, DE
JHPUSH

;

DEFB    82H                ; D+ ( d1L d1H d2L d2h -- d3L d3H)
DM      'D+'
DEFW    PLUS-4
D PLUS: DEFW    $+2
EXX                    ; / SAVE IP
POP     BC                ; (BC) <-- d2H
POP     HL                ; (HL) <-- d2L
POP     AF                ; d (AF) <-- d1H
POP     DE                ; (DE) <-- d1L
PUSH    AF                ; / (S1) <-- d1H
ADD     HL, DE                ; (HL) <-- d2L+d1L=d3L
EX      DE, HL                ; (DE) <-- d3L
POP     HL                ; (HL) <-- d1H
ADC     HL, BC                ; / (HL) <-- d1H+d2H+CARRY=d3H
PUSH    DE                ; (S2) <-- d3L
PUSH    HL                ; / (S1) <-- d3H
EXX                    ; / RESTORE IP
JNEXT

;

DEFB    85H                ; MINUS
DM      'MINUS'

```

```

MINUS:  DEFW    DPLUS-5
        DEFW    $+2
        POP     DE                ;/
        XOR     A                ;/ RESET CARRY, (A)<--0
        LD      H,A              ;/
        LD      L,A              ;/ LD HL,0
        SBC     HL,DE            ;/ (HL)<--(DE)2's COMPL.
        JHPUSH

;

        DEFB    86H              ;DMINUS
        DM      'DMINUS'
        DEFW    MINUS-8
DMINU:  DEFW    $+2
        POP     HL                ;(HL)<--d1H
        POP     DE                ;(DE)<--d1L
        SUB     A                ;(A)<--0
        SUB     E
        LD      E,A              ;(E)<--NEG(E)
        LD      A,00H
        SBC     A,D
        LD      D,A              ;(D)<--NEG(D)
        LD      A,00H
        SBC     A,L
        LD      L,A              ;(L)<--NEG(L)
        LD      A,00H
        SBC     A,H
        LD      H,A              ;(H)<--NEG(H)
        JP      DPUSH            ;(S2)<--d2L, (S1)<--d2H

;

        DEFB    84H              ;OVER
        DM      'OVER'
        DEFW    DMINU-9
OVER:   DEFW    $+2
        POP     DE
        POP     HL
        PUSH    HL
        JP      DPUSH

;

        DEFB    84H              ;DROP
        DM      'DROP'
        DEFW    OVER-7
DROP:   DEFW    $+2
        POP     HL
        JNEXT

;

        DEFB    84H              ;SWAP
        DM      'SWAP'
        DEFW    DROP-7
SWAP:   DEFW    $+2
        POP     HL
        EX      (SP),HL
        JHPUSH

;

        DEFB    83H              ;DUP
        DM      'DUP'
        DEFW    SWAP-7
DUP:    DEFW    $+2
        POP     HL
        PUSH    HL
        JHPUSH

;

        DEFB    84H              ;2DUP
        DM      '2DUP'
        DEFW    DUP-6

```

```

TDUP:  DEFW  $+2
        POP   HL
        POP   DE
        PUSH  DE
        PUSH  HL
        JP    DPUSH
;
        DEFB  82H          ;+!
        DM    '+!'
        DEFW  TDUP-7
PSTOR:  DEFW  $+2
        POP   HL          ;(HL)<--VAR ADDR
        POP   DE          ;(DE)<--NUMBER
        LD    A,(HL)
        ADD   A,E
        LD    (HL),A
        INC   HL
        LD    A,(HL)
        ADC   A,D
        LD    (HL),A      ;((HL)<--((HL))+NUMBER
        JNEXT
;
        DEFB  86H          ;TOGGLE
        DM    'TOGGLE'
        DEFW  PSTOR-5
TOGGL:  DEFW  $+2
        POP   DE          ;(E)<--BIT PATTERN
        POP   HL          ;(HL)<--ADDR
        LD    A,(HL)
        XOR   E
        LD    (HL),A
        JNEXT
;
        DEFB  81H          ;@
        DM    '@'
        DEFW  TOGGL-9
AT:     DEFW  $+2
        POP   HL
        LD    E,(HL)
        INC   HL
        LD    D,(HL)
        PUSH  DE
        JNEXT
;
        DEFB  82H          ;C@
        DM    'C@'
        DEFW  AT-4
CAT:    DEFW  $+2
        POP   HL
        LD    L,(HL)
        LD    H,0
        JHPUSH
;
        DEFB  82H          ;2@
        DM    '2@'
        DEFW  CAT-5
TAT:    DEFW  $+2
        EXX          ;/ SAVE IP
        POP   HL          ; (HL)<--ADDR
        LD    C,(HL)
        INC   HL
        LD    B,(HL)      ;/ (BC)<--dH
        INC   HL
        LD    E,(HL)

```

```

INC      HL
LD       D,(HL)      ; (DE)<--dL
PUSH     DE           ; (S2)<--dL
PUSH     BC           ;/ (S1)<--dH
EXX      ;/ RESTORE IP
JNEXT

;
DEFB     81H          ;!
DM       '! '
DEFW     TAT-5
STORE:   DEFW         $+2
POP      HL
POP      DE
LD       (HL),E
INC      HL
LD       (HL),D
JNEXT

;
DEFB     82H          ;C!
DM       'C! '
DEFW     STORE-4
CSTOR:   DEFW         $+2
POP      HL
POP      DE
LD       (HL),E
JNEXT

;
DEFB     82H          ;2!
DM       '2! '
DEFW     CSTOR-5
TSTOR:   DEFW         $+2
POP      HL
POP      DE
LD       (HL),E
INC      HL
LD       (HL),D
INC      HL
POP      DE
LD       (HL),E
INC      HL
LD       (HL),D
JNEXT

;
DEFB     0C1H         ; :
DM       ': '
DEFW     TSTOR-5
COLON:   DEFW         DOCOL
DEFW     QEXEC
DEFW     SCSP
DEFW     CURR
DEFW     AT
DEFW     CONT
DEFW     STORE
DEFW     CREAT
DEFW     RBRAC
DEFW     PSCOD
DOCOL:   LD          HL,(RPP)
DEC      HL
LD       (HL),B
DEC      HL
LD       (HL),C
LD       (RPP),HL
INC      DE
LD       C,E

```



```

LD      B,D
JNEXT

;
DEFB    0C1H      ; ;
DM      ';'
DEFW    COLON-4
SEMI:   DEFW      DOCOL
DEFW    QCSP
DEFW    COMP
DEFW    SEMIS
DEFW    SMUDG
DEFW    LBRAC
DEFW    SEMIS

;
DEFB    84H      ;NOOP
DM      'NOOP'
DEFW    SEMI-4
NOOP:   DEFW      DOCOL
DEFW    SEMIS

;
DEFB    88H      ;CONSTANT
DM      'CONSTANT'
DEFW    NOOP-7
CON:    DEFW      DOCOL
DEFW    CREAT
DEFW    SMUDG
DEFW    COMMA
DEFW    PSCOD
DOCON:  INC      DE
EX      DE,HL
LD      E,(HL)
INC     HL
LD      D,(HL)
PUSH    DE
JNEXT

;
DEFB    88H      ;VARIABLE
DM      'VARIABLE'
DEFW    CON-0BH
VAR:    DEFW      DOCOL
DEFW    CON
DEFW    PSCOD
DOVAR:  INC      DE
PUSH    DE
JNEXT

;
DEFB    84H      ;USER
DM      'USER'
DEFW    VAR-0BH
USER:   DEFW      DOCOL
DEFW    CON
DEFW    PSCOD
DOUSE:  INC      DE
EX      DE,HL
LD      E,(HL)
LD      D,00H
LD      HL,(UP)
ADD     HL,DE
JHPUSH

;
DEFB    81H      ;0
DM      '0'
DEFW    USER-7
ZERO:   DEFW      $+2      ;/

```

```

LD      HL,0      ;/
JHPUSH      ;/
;
DEFB     81H      ;1
DM       '1'
DEFW     ZERO-4
ONE:     DEFW     $+2      ;/
LD       HL,1      ;/
JHPUSH      ;/
;
DEFB     81H      ;2
DM       '2'
DEFW     ONE-4
TWO:     DEFW     $+2      ;/
LD       HL,2      ;/
JHPUSH      ;/
;
DEFB     81H      ;3
DM       '3'
DEFW     TWO-4
THREE:   DEFW     $+2      ;/
LD       HL,3      ;/
JHPUSH      ;/
;
DEFB     82H      ;BL
DM       'BL'
DEFW     THREE-4
BL:      DEFW     DOCON
DEFW     20H
;
DEFB     83H      ;C/L
DM       'C/L'
DEFW     BL-5
CSLL:    DEFW     DOCON
DEFW     64
;
DEFB     85H      ;FIRST
DM       'FIRST'
DEFW     CSLL-6
FIRST:   DEFW     DOCON
DEFW     0      ;/ set by CLD
;
DEFB     85H      ;LIMIT
DM       'LIMIT'
DEFW     FIRST-8
LIMIT:   DEFW     DOCON
DEFW     0      ;/ set by CLD
;
DEFB     85H      ;B/BUF
DM       'B/BUF'
DEFW     LIMIT-8
BBUF:    DEFW     DOCON
DEFW     KBBUF
;
DEFB     85H      ;B/SCR
DM       'B/SCR'
DEFW     BBUF-8
BSCR:    DEFW     DOCON
DEFW     400H/KBBUF
;
DEFB     87H      ;+ORIGIN
DM       '+ORIGIN'
DEFW     BSCR-8
PORIG:   DEFW     D0COL

```

```

      DEFW      LIT
      DEFW      ORIG
      DEFW      PLUS
      DEFW      SEMIS
;
;      USER VARIABLES
;
      DEFB      82H          ;S0
      DM        'S0'
      DEFW      PORIG-0AH
SZERO:  DEFW      DOUSE
      DEFW      6
;
      DEFB      82H          ;R0
      DM        'R0'
      DEFW      SZERO-5
RZERO:  DEFW      DOUSE
      DEFW      8
;
      DEFB      83H          ;TIB
      DM        'TIB'
      DEFW      RZERO-5
TIB:    DEFW      DOUSE
      DEFB      0AH
;
      DEFB      85H          ;WIDTH
      DM        'WIDTH'
      DEFW      TIB-6
WIDTH:  DEFW      DOUSE
      DEFB      0CH
;
      DEFB      87H          ;WARNING
      DM        'WARNING'
      DEFW      WIDTH-8
WARN:   DEFW      DOUSE
      DEFB      0EH
;
      DEFB      85H          ;FENCE
      DM        'FENCE'
      DEFW      WARN-0AH
FENCE:  DEFW      DOUSE
      DEFB      10H
;
      DEFB      82H          ;DP
      DM        'DP'
      DEFW      FENCE-8
DP:     DEFW      DOUSE
      DEFB      12H
;
      DEFB      88H          ;VOC-LINK
      DM        'VOC-LINK'
      DEFW      DP-5
VOCL:   DEFW      DOUSE
      DEFW      14H
;
      DEFB      83H          ;BLK
      DM        'BLK'
      DEFW      VOCL-0BH
BLK:    DEFW      DOUSE
      DEFB      16H
;
      DEFB      82H          ;IN
      DM        'IN'
      DEFW      BLK-6

```

```

INN:    DEFW    DOUSE
        DEFB    18H
;
        DEFB    83H            ;OUT
        DM      'OUT'
        DEFW    INN-5
OUTT:   DEFW    DOUSE
        DEFB    1AH
;
        DEFB    83H            ;SCR
        DM      'SCR'
        DEFW    OUTT-6
SCR:    DEFW    DOUSE
        DEFB    1CH
;
        DEFB    86H            ;OFFSET
        DM      'OFFSET'
        DEFW    SCR-6
OFSET:  DEFW    DOUSE
        DEFB    1EH
;
        DEFB    87H            ;CONTEXT
        DM      'CONTEXT'
        DEFW    OFSET-9
CONT:   DEFW    DOUSE
        DEFB    20H
;
        DEFB    87H            ;CURRENT
        DM      'CURRENT'
        DEFW    CONT-0AH
CURR:   DEFW    DOUSE
        DEFB    22H
;
        DEFB    85H            ;STATE
        DM      'STATE'
        DEFW    CURR-0AH
STATE:  DEFW    DOUSE
        DEFB    24H
;
        DEFB    84H            ;BASE
        DM      'BASE'
        DEFW    STATE-8
BASE:   DEFW    DOUSE
        DEFB    26H
;
        DEFB    83H            ;DPL
        DM      'DPL'
        DEFW    BASE-7
DPL:    DEFW    DOUSE
        DEFB    28H
;
        DEFB    83H            ;FLD
        DM      'FLD'
        DEFW    DPL-6
FLD:    DEFW    DOUSE
        DEFB    2AH
;
        DEFB    83H            ;CSP
        DM      'CSP'
        DEFW    FLD-6
CSPP:   DEFW    DOUSE
        DEFB    2CH
;
        DEFB    82H            ;R#

```

```

      DM      'R#'
      DEFW    CSPP-6
RNUM:  DEFW    DOUSE
      DEFB    2EH
;
      DEFB    83H          ;HLD
      DM      'HLD'
      DEFW    RNUM-5
HLD:   DEFW    DOUSE
      DEFW    30H
;
;      END OF USER VARIABLES
;
      DEFB    82H          ;1+
      DM      '1+'
      DEFW    HLD-6
ONEP:  DEFW    $+2          ;/
      POP     HL           ;/
      INC     HL           ;/
      JHPUSH           ;/
;
      DEFB    82H          ;2+
      DM      '2+'
      DEFW    ONEP-5
TWOP:  DEFW    $+2          ;/
      POP     HL           ;/
      INC     HL           ;/
      INC     HL           ;/
      JHPUSH           ;/
;
      DEFB    82H          ;/ 1-
      DM      '1-'          ;/
      DEFW    TWOP-5        ;/
ONEMIN: DEFW    $+2          ;/
      POP     HL           ;/
      DEC     HL           ;/
      JHPUSH           ;/
;
      DEFB    82H          ;/ 2-
      DM      '2-'          ;/
      DEFW    ONEMIN-5      ;/
TWOMIN: DEFW    $+2          ;/
      POP     HL           ;/
      DEC     HL           ;/
      DEC     HL           ;/
      JHPUSH           ;/
;
      DEFB    82H          ;/ 2*
      DM      '2*'          ;/
      DEFW    TWOMIN-5      ;/
TWOSTA: DEFW    $+2          ;/
      POP     HL           ;/
      ADD     HL,HL         ;/ ASL HL
      JHPUSH           ;/
;
      DEFB    82H          ;/ 2/
      DM      '2/'          ;/
      DEFW    TWOSTA-5      ;/
TWOSLA: DEFW    $+2          ;/
      POP     HL           ;/
      BIT     7,H           ;/
      JR      Z,TWOSL1      ;/
      INC     HL           ;/
TWOSL1: SRA     H           ;/

```

```

RR      L      ;/ ASR HL
JHPUSH  ;/

;
DEFB    84H      ;HERE
DM      'HERE'
DEFW    TWOSLA-5
HERE:   DEFW    DOCOL
        DEFW    DP
        DEFW    AT
        DEFW    SEMIS

;
DEFB    85H      ;ALLOT
DM      'ALLOT'
DEFW    HERE-7
ALLOT:  DEFW    DOCOL
        DEFW    DP
        DEFW    PSTOR
        DEFW    SEMIS

;
DEFB    81H      ; ,
DM      ', '
DEFW    ALLOT-8
COMMA:  DEFW    DOCOL
        DEFW    HERE
        DEFW    STORE
        DEFW    TWO
        DEFW    ALLOT
        DEFW    SEMIS

;
DEFB    82H      ;C,
DM      'C, '
DEFW    COMMA-4
CCOMM:  DEFW    DOCOL
        DEFW    HERE
        DEFW    CSTOR
        DEFW    ONE
        DEFW    ALLOT
        DEFW    SEMIS

;
DEFB    81H      ; -
DM      '_ '
DEFW    CCOMM-5
SUBB:   DEFW    $+2
        POP     DE
        POP     HL
        OR      A      ;/ RESET CARRY
        SBC     HL,DE   ;/
        JHPUSH

;
DEFB    81H      ;=
DM      '='
DEFW    SUBB-4
EQUAL:  DEFW    $+2      ;/
        POP     DE      ;/
        POP     HL      ;/
        XOR     A      ;/ RESET CARRY
        SBC     HL,DE   ;/
        LD      H,A     ;/
        LD      L,A     ;/ LD HL,0000H
        JR      NZ,EXEQU ;/ FALSE
        INC     L      ;/ TRUE
EXEQU:  JHPUSH      ;/
;
DEFB    81H      ; <

```

```

DM      '<'
DEFW    EQUAL-4
LESS:   DEFW    $+2
        POP     DE
        POP     HL          ; (HL) (DE) <
        LD      A,D
        XOR     H          ; ONE OF THEM NEGATIVE?
        JP      M,LES1     ; YES, DETERMINE WHICH
        OR      A          ;/ CLR CARRY
        SBC     HL,DE      ;/
LES1:   BIT     7,H        ;/ (H) NEGATIVE?
        LD      HL,0
        JR      Z,EXLESS   ;/ NO, FALSE
        INC     L          ;/ TRUE
EXLESS: JHPUSH
;
        DEFB    82H        ;U<
        DM      'U<'
        DEFW    LESS-4
ULESS:  DEFW    $+2        ;/
        POP     DE
        POP     HL        ;/ (HL) (DE) U<
        XOR     A          ;/
        SBC     HL,DE      ;/
        LD      H,A        ;/
        LD      L,A        ;/ LD HL,0000H
        JR      NC,EXULES  ;/ FALSE
        INC     L          ;/ TRUE
EXULES: JHPUSH
;
        DEFB    81H        ;>
        DM      '>'
        DEFW    ULESS-5
GREAT:  DEFW    $+2
        POP     HL        ;/
        POP     DE        ;/ (HL) (DE) > = (DE) (HL) <
        LD      A,D
        XOR     H          ; ONE OF THEM NEGATIVE?
        JP      M,GREAT1   ; YES, DETERMINE WHICH
        OR      A          ;/ CLR CARRY
        SBC     HL,DE      ;/
GREAT1: BIT     7,H        ;/ (H) NEGATIVE?
        LD      HL,0        ; (HL)<--FALSE
        JR      Z,GREAT2   ;/ NO, FALSE
        INC     L          ;/ (HL)<--TRUE
GREAT2: JHPUSH
;
        DEFB    83H        ;ROT
        DM      'ROT'
        DEFW    GREAT-4
ROT:    DEFW    $+2
        POP     DE
        POP     HL
        EX      (SP),HL
        JP      DPUSH
;
        DEFB    85H        ;SPACE
        DM      'SPACE'
        DEFW    ROT-6
SPACE:  DEFW    DOCOL
        DEFW    BL
        DEFW    EMIT
        DEFW    SEMIS
;

```

```

      DEFB      84H          ; -DUP
      DM        '-DUP'
      DEFW      SPACE-8
DDUP:  DEFW      $+2          ;/
      POP       HL          ;/
      LD        A,H          ;/
      OR        L            ;/ (HL)=0?
      JR        Z,EXDDUP     ;/ YES, DON'T DUP
      PUSH      HL          ;/
EXDDUP: JHPUSH
;
      DEFB      88H          ; TRAVERSE
      DM        'TRAVERSE'
      DEFW      DDUP-7
TRAV:  DEFW      DOCOL
      DEFW      SWAP
TRAV1: DEFW      OVER        ; BEGIN
      DEFW      PLUS
      DEFW      LIT
      DEFW      7FH
      DEFW      OVER
      DEFW      CAT
      DEFW      LESS
      DEFW      ZBRAN        ; UNTIL
      DEFW      TRAV1-$
      DEFW      SWAP
      DEFW      DROP
      DEFW      SEMIS
;
      DEFB      86H          ; LATEST
      DM        'LATEST'
      DEFW      TRAV-0BH
LATES: DEFW      DOCOL
      DEFW      CURR
      DEFW      AT
      DEFW      AT
      DEFW      SEMIS
;
      DEFB      83H          ; LFA
      DM        'LFA'
      DEFW      LATES-9
LFA:  DEFW      $+2          ;/
      POP       HL          ;/ (HL)<--PFA
      DEC       HL          ;/
      DEC       HL          ;/
      DEC       HL          ;/
      DEC       HL          ;/ (HL)<--(HL)-4 = LFA
      JHPUSH
;
      DEFB      83H          ; CFA
      DM        'CFA'
      DEFW      LFA-6
CFA:  DEFW      DOCOL
      DEFW      TWOMIN      ;/
      DEFW      SEMIS
;
      DEFB      83H          ; NFA
      DM        'NFA'
      DEFW      CFA-6
NFA:  DEFW      DOCOL
      DEFW      LIT
      DEFW      5
      DEFW      SUBB
      DEFW      LIT

```



```

DEFW -1
DEFW TRAV
DEFW SEMIS

;
DEFB 83H ;PFA
DM 'PFA'
DEFW NFA-6
PFA: DEFW DOCOL
DEFW ONE
DEFW TRAV
DEFW LIT
DEFW 5
DEFW PLUS
DEFW SEMIS

;
DEFB 84H ;!CSP
DM '!CSP'
SCSP: DEFW PFA-6
DEFW DOCOL
DEFW SPAT
DEFW CSPP
DEFW STORE
DEFW SEMIS

;
DEFB 86H ;?ERROR
DM '?ERROR'
QERR: DEFW SCSP-7
DEFW DOCOL
DEFW SWAP
DEFW ZBRAN ;IF
DEFW QERR1-$
DEFW ERROR
DEFW BRAN ;ELSE
QERR1: DEFW QERR2-$
QERR2: DEFW DROP ;ENDIF
DEFW SEMIS

;
DEFB 85H ;?COMP
DM '?COMP'
QCOMP: DEFW QERR-9
DEFW DOCOL
DEFW STATE
DEFW AT
DEFW ZEQU
DEFW LIT
DEFW 11H
DEFW QERR
DEFW SEMIS

;
DEFB 85H ;?EXEC
DM '?EXEC'
QEXEC: DEFW QCOMP-8
DEFW DOCOL
DEFW STATE
DEFW AT
DEFW LIT
DEFW 12H
DEFW QERR
DEFW SEMIS

;
DEFB 86H ;?PAIRS
DM '?PAIRS'
QPAIR: DEFW QEXEC-8
DEFW DOCOL

```

```

      DEFW      SUBB
      DEFW      LIT
      DEFW      13H
      DEFW      QERR
      DEFW      SEMIS
;
      DEFB      84H          ;?CSP
      DM        '?CSP'
      DEFW      QPAIR-9
QCSP:  DEFW      DOCOL
      DEFW      SPAT
      DEFW      CSPP
      DEFW      AT
      DEFW      SUBB
      DEFW      LIT
      DEFW      14H
      DEFW      QERR
      DEFW      SEMIS
;
      DEFB      88H          ;?LOADING
      DM        '?LOADING'
      DEFW      QCSP-7
QLOAD: DEFW      DOCOL
      DEFW      BLK
      DEFW      AT
      DEFW      ZEQU
      DEFW      LIT
      DEFW      16H
      DEFW      QERR
      DEFW      SEMIS
;
      DEFB      87H          ;COMPILE
      DM        'COMPILE'
      DEFW      QLOAD-0BH
COMP:  DEFW      DOCOL
      DEFW      QCOMP
      DEFW      FROMR
      DEFW      DUP
      DEFW      TWOP
      DEFW      TOR
      DEFW      AT
      DEFW      COMMA
      DEFW      SEMIS
;
      DEFB      0C1H         ;[
      DM        '['
      DEFW      COMP-0AH
LBRAC: DEFW      DOCOL
      DEFW      ZERO
      DEFW      STATE
      DEFW      STORE
      DEFW      SEMIS
;
      DEFB      81H          ;]
      DM        ']'
      DEFW      LBRAC-4
RBRAC: DEFW      DOCOL
      DEFW      LIT,0C0H
      DEFW      STATE,STORE
      DEFW      SEMIS
;
      DEFB      86H          ;SMUDGE
      DM        'SMUDGE'
      DEFW      RBRAC-4

```

```

SMUDG:  DEFW  DOCOL
        DEFW  LATES
        DEFW  LIT
        DEFW  20H
        DEFW  TOGGL
        DEFW  SEMIS

;
        DEFB  83H          ;HEX
        DM    'HEX'
        DEFW  SMUDG-9
HEX:    DEFW  DOCOL
        DEFW  LIT
        DEFW  10H
        DEFW  BASE
        DEFW  STORE
        DEFW  SEMIS

;
        DEFB  87H          ;DECIMAL
        DM    'DECIMAL'
        DEFW  HEX-6
DEC:    DEFW  DOCOL
        DEFW  LIT
        DEFW  0AH
        DEFW  BASE
        DEFW  STORE
        DEFW  SEMIS

;
        DEFB  87H          ;(;CODE)
        DM    ' (;CODE) '
        DEFW  DEC-0AH
PSCOD:  DEFW  DOCOL
        DEFW  FROMR
        DEFW  LATES
        DEFW  PFA
        DEFW  CFA
        DEFW  STORE
        DEFW  SEMIS

;
        DEFB  0C5H         ; ;CODE
        DM    ' ;CODE '
        DEFW  PSCOD-0AH
SEMIC:  DEFW  DOCOL
        DEFW  QCSP
        DEFW  COMP
        DEFW  PSCOD
        DEFW  LBRAC
SEMI1:  DEFW  NOOP          ;ASSEMBLER
        DEFW  SEMIS

;
        DEFB  87H          ;<BUILDS
        DM    '<BUILDS'
        DEFW  SEMIC-8
BUILD:  DEFW  DOCOL
        DEFW  ZERO
        DEFW  CON
        DEFW  SEMIS

;
        DEFB  85H          ;DOES>
        DM    'DOES>'
        DEFW  BUILD-0AH
DOES:   DEFW  DOCOL
        DEFW  FROMR
        DEFW  LATES
        DEFW  PFA

```

```

      DEFW    STORE
      DEFW    PSCOD
DODOE: LD      HL,(RPP)
      DEC     HL
      LD      (HL),B
      DEC     HL
      LD      (HL),C
      LD      (RPP),HL
      INC     DE
      EX      DE,HL
      LD      C,(HL)
      INC     HL
      LD      B,(HL)
      INC     HL
      JHPUSH

;
      DEFB    85H          ;COUNT
      DM      'COUNT'
      DEFW    DOES-8
COUNT: DEFW    DOCOL
      DEFW    DUP
      DEFW    ONEP
      DEFW    SWAP
      DEFW    CAT
      DEFW    SEMIS

;
      DEFB    84H          ;TYPE
      DM      'TYPE'
      DEFW    COUNT-8
TYPE:   DEFW    DOCOL
      DEFW    DDUP
      DEFW    ZBRAN        ;IF
      DEFW    TYPE1-$
      DEFW    OVER
      DEFW    PLUS
      DEFW    SWAP
      DEFW    XDO          ;DO
TYPE2:  DEFW    IDO
      DEFW    CAT
      DEFW    EMIT
      DEFW    XLOOP        ;LOOP
      DEFW    TYPE2-$
      DEFW    BRAN         ;ELSE
      DEFW    TYPE3-$
TYPE1:  DEFW    DROP        ;ENDIF
TYPE3:  DEFW    SEMIS

;
      DEFB    89H          ;-TRAILING
      DM      '-TRAILING'
      DEFW    TYPE-7
DTRAI:  DEFW    DOCOL
      DEFW    DUP
      DEFW    ZERO
      DEFW    XDO          ;DO
DTRA1:  DEFW    TDUP        ;/
      DEFW    PLUS
      DEFW    ONEMIN       ;/
      DEFW    CAT
      DEFW    BL
      DEFW    SUBB
      DEFW    ZBRAN        ;IF
      DEFW    DTRA2-$
      DEFW    LEAVE
      DEFW    BRAN         ;ELSE

```

```

      DEFW      DTRA3-$
DTRA2: DEFW      ONEMIN      ;/
DTRA3: DEFW      XLOOP      ;LOOP
      DEFW      DTRA1-$
      DEFW      SEMIS
;
      DEFB      84H          ;(.")
      DM        '(.")'
      DEFW      DTRAI-0CH
PDOTQ: DEFW      DOCOL
      DEFW      RR
      DEFW      COUNT
      DEFW      DUP
      DEFW      ONEP
      DEFW      FROMR
      DEFW      PLUS
      DEFW      TOR
      DEFW      TYPE
      DEFW      SEMIS
;
      DEFB      0C2H          ;. "
      DM        '." '
      DEFW      PDOTQ-7
DOTQ:  DEFW      DOCOL
      DEFW      LIT
      DEFW      22H
      DEFW      STATE
      DEFW      AT
      DEFW      ZBRAN        ;IF
      DEFW      DOTQ1-$
      DEFW      COMP
      DEFW      PDOTQ
      DEFW      WORD
      DEFW      HERE
      DEFW      CAT
      DEFW      ONEP
      DEFW      ALLOT
      DEFW      BRAN        ;ELSE
      DEFW      DOTQ2-$
DOTQ1: DEFW      WORD
      DEFW      HERE
      DEFW      COUNT
      DEFW      TYPE        ;ENDIF
DOTQ2: DEFW      SEMIS
;
      DEFB      86H          ;EXPECT
      DM        'EXPECT'
      DEFW      DOTQ-5
EXPEC: DEFW      DOCOL
      DEFW      OVER
      DEFW      PLUS
      DEFW      OVER
      DEFW      XDO          ;DO
EXPE1: DEFW      KEY
      DEFW      DUP
      DEFW      LIT
      DEFW      0EH
      DEFW      PORIG
      DEFW      AT
      DEFW      EQUAL
      DEFW      ZBRAN        ;IF
      DEFW      EXPE2-$
      DEFW      DROP
      DEFW      DUP

```

```

DEFW  IDO
DEFW  EQUAL
DEFW  DUP
DEFW  FROMR
DEFW  TWOMIN      ;/
DEFW  PLUS
DEFW  TOR
DEFW  ZBRAN      ;IF
DEFW  EXPE6-$
DEFW  LIT
DEFW  BELL
DEFW  BRAN      ;ELSE
DEFW  EXPE7-$
EXPE6: DEFW  LIT
DEFW  BSOUT      ;ENDIF
EXPE7: DEFW  BRAN      ;ELSE
DEFW  EXPE3-$
EXPE2: DEFW  DUP
DEFW  LIT
DEFW  ACR      ;/
DEFW  EQUAL
DEFW  ZBRAN      ;IF
DEFW  EXPE4-$
DEFW  LEAVE
DEFW  DROP
DEFW  BL
DEFW  ZERO
DEFW  BRAN      ;ELSE
DEFW  EXPE5-$
EXPE4: DEFW  DUP      ;ENDIF
EXPE5: DEFW  IDO
DEFW  CSTOR
DEFW  ZERO
DEFW  IDO
DEFW  ONEP
DEFW  STORE      ;ENDIF
EXPE3: DEFW  EMIT
DEFW  XLOOP      ;LOOP
DEFW  EXPE1-$
DEFW  DROP
DEFW  SEMIS
;
DEFB  85H      ;QUERY
DM    'QUERY'
DEFW  EXPEC-9
QUERY: DEFW  DOCOL
DEFW  TIB
DEFW  AT
DEFW  LIT
DEFW  50H
DEFW  EXPEC
DEFW  ZERO
DEFW  INN
DEFW  STORE
DEFW  SEMIS
;
DEFB  0C1H      ;NULL
DEFB  80H
DEFW  QUERY-8
NULL:  DEFW  DOCOL
DEFW  BLK
DEFW  AT
DEFW  ZBRAN      ;IF
DEFW  NULL1-$

```

```

DEFW ONE
DEFW BLK
DEFW PSTOR
DEFW ZERO
DEFW INN
DEFW STORE
DEFW BLK
DEFW AT
DEFW BSCR
DEFW ONEMIN      ;/
DEFW ANDD
DEFW ZEQU
DEFW ZBRAN      ;IF
DEFW NULL2-$
DEFW QEXEC
DEFW FROMR
DEFW DROP      ;ENDIF
NULL2: DEFW BRAN      ;ELSE
DEFW NULL3-$
NULL1: DEFW FROMR
DEFW DROP      ;ENDIF
NULL3: DEFW SEMIS
;
DEFB 84H      ;FILL
DM 'FILL'
DEFW NULL-4
FILL: DEFW $+2
EXX      ;/ SAVE IP
POP DE      ;/ (E)<--BYTE
POP BC      ; (BC)<--QUANTITY
POP HL      ;/ (HL)<--ADDR
FILL1: LD A,B
OR C      ; QTY=0?
JR Z,FILL2  ; YES
LD (HL),E  ;/ ((HL))<--BYTE
INC HL      ; INC POINTER
DEC BC      ; DEC COUNTER
JP FILL1    ;/
FILL2: EXX      ;/ RESTORE IP
JNEXT
;
DEFB 85H      ;ERASE
DM 'ERASE'
DEFW FILL-7
ERASEE: DEFW DOCOL
DEFW ZERO
DEFW FILL
DEFW SEMIS
;
DEFB 86H      ;BLANKS
DM 'BLANKS'
DEFW ERASEE-8
BLANK: DEFW DOCOL
DEFW BL
DEFW FILL
DEFW SEMIS
;
DEFB 84H      ;HOLD
DM 'HOLD'
DEFW BLANK-9
HOLD: DEFW DOCOL
DEFW LIT
DEFW -1
DEFW HLD

```

```

DEFW PSTOR
DEFW HLD
DEFW AT
DEFW CSTOR
DEFW SEMIS

;
DEFB 83H          ;PAD
DM 'PAD'
DEFW HOLD-7
PAD: DEFW DOCOL
DEFW HERE
DEFW LIT
DEFW 44H
DEFW PLUS
DEFW SEMIS

;
DEFB 84H          ;WORD
DM 'WORD'
DEFW PAD-6
WORD: DEFW DOCOL
DEFW BLK
DEFW AT
DEFW ZBRAN        ;IF
DEFW WORD1-$
DEFW BLK
DEFW AT
DEFW BLOCK
DEFW BRAN         ;ELSE
DEFW WORD2-$
WORD1: DEFW TIB
DEFW AT           ;ENDIF
WORD2: DEFW INN
DEFW AT
DEFW PLUS
DEFW SWAP
DEFW ENCL
DEFW HERE
DEFW LIT
DEFW 22H
DEFW BLANK
DEFW INN
DEFW PSTOR
DEFW OVER
DEFW SUBB
DEFW TOR
DEFW RR
DEFW HERE
DEFW CSTOR
DEFW PLUS
DEFW HERE
DEFW ONEP
DEFW FROMR
DEFW CMOVE
DEFW SEMIS

;
DEFB 88H          ;(NUMBER)
DM '(NUMBER)'
DEFW WORD-7
PNUMB: DEFW DOCOL
PNUM1: DEFW ONEP   ;BEGIN
DEFW DUP
DEFW TOR
DEFW CAT
DEFW BASE

```



```

DEFW AT
DEFW DIGIT
DEFW ZBRAN ;WHILE
DEFW PNUM2-$
DEFW SWAP
DEFW BASE
DEFW AT
DEFW USTAR
DEFW DROP
DEFW ROT
DEFW BASE
DEFW AT
DEFW USTAR
DEFW DPLUS
DEFW DPL
DEFW AT
DEFW ONEP
DEFW ZBRAN ;IF
DEFW PNUM3-$
DEFW ONE
DEFW DPL
DEFW PSTOR ;ENDIF
PNUM3: DEFW FROMR
DEFW BRAN ;REPEAT
DEFW PNUM1-$
PNUM2: DEFW FROMR
DEFW SEMIS
;
DEFB 86H ;NUMBER
DM 'NUMBER'
DEFW PNUMB-0BH
NUMB: DEFW DOCOL
DEFW ZERO
DEFW ZERO
DEFW ROT
DEFW DUP
DEFW ONEP
DEFW CAT
DEFW LIT
DEFW 2DH
DEFW EQUAL
DEFW DUP
DEFW TOR
DEFW PLUS
DEFW LIT
DEFW -1
NUMB1: DEFW DPL ;BEGIN
DEFW STORE
DEFW PNUMB
DEFW DUP
DEFW CAT
DEFW BL
DEFW SUBB
DEFW ZBRAN ;WHILE
DEFW NUMB2-$
DEFW DUP
DEFW CAT
DEFW LIT
DEFW 2EH
DEFW SUBB
DEFW ZERO
DEFW QERR
DEFW ZERO
DEFW BRAN ;REPEAT

```

```

NUMB2: DEFW NUMB1-$
        DEFW DROP
        DEFW FROMR
        DEFW ZBRAN          ;IF
        DEFW NUMB3-$
        DEFW DMINU          ;ENDIF
NUMB3: DEFW SEMIS
;
        DEFB 85H          ; -FIND (0-3) SUCCESS
        DM   '-FIND'      ;      (0-1) FAILURE
        DEFW NUMB-9
DFIND:  DEFW DOCOL
        DEFW BL
        DEFW WORD
        DEFW HERE
        DEFW CONT
        DEFW AT
        DEFW AT
        DEFW PFIND
        DEFW DUP
        DEFW ZEQU
        DEFW ZBRAN          ;IF
        DEFW DFIN1-$
        DEFW DROP
        DEFW HERE
        DEFW LATES
        DEFW PFIND          ;ENDIF
DFIN1:  DEFW SEMIS
;
        DEFB 87H          ; (ABORT)
        DM   '(ABORT)'
        DEFW DFIND-8
PABOR:  DEFW DOCOL
        DEFW ABORT
        DEFW SEMIS
;
        DEFB 85H          ; ERROR
        DM   'ERROR'
        DEFW PABOR-0AH
ERROR:  DEFW DOCOL
        DEFW WARN
        DEFW AT
        DEFW ZLESS
        DEFW ZBRAN          ;IF
        DEFW ERRO1-$
        DEFW PABOR          ;ENDIF
ERRO1:  DEFW HERE
        DEFW COUNT
        DEFW TYPE
        DEFW PDOTQ
        DEFB 2
        DB   '? '
        DEFW MESS
        DEFW SPSTO
;
;      CHANGE FROM fig MODEL
        DEFW INN,AT,BLK,AT
        DEFW BLK,AT
        DEFW DDUP
        DEFW ZBRAN,ERRO2-$  ;IF
        DEFW INN,AT
        DEFW SWAP            ;ENDIF
ERRO2:  DEFW QUIT
;
        DEFB 83H          ; ID.

```

```

DM      'ID.'
DEFW    ERROR-8
IDDOT:  DEFW    DOCOL
        DEFW    PAD
        DEFW    LIT
        DEFW    20H
        DEFW    BLANK      ;/
        DEFW    DUP
        DEFW    PFA
        DEFW    LFA
        DEFW    OVER
        DEFW    SUBB
        DEFW    DUP        ;/ change frm MODEL
        DEFW    TOR        ;/ to suppress BIT 7
        DEFW    PAD
        DEFW    SWAP
        DEFW    CMOVE
        DEFW    PAD
        DEFW    FROMR      ;/ for terminals
        DEFW    PAD        ;/ with an 8 bit
        DEFW    PLUS       ;/ ASCII character set.
        DEFW    ONEMIN     ;/
        DEFW    DUP        ;/
        DEFW    AT         ;/
        DEFW    LIT        ;/
        DEFW    7FH        ;/
        DEFW    ANDD       ;/
        DEFW    SWAP       ;/
        DEFW    STORE      ;/
        DEFW    COUNT
        DEFW    LIT
        DEFW    1FH        ; WIDTH
        DEFW    ANDD
        DEFW    TYPE
        DEFW    SPACE
        DEFW    SEMIS
;
DEFB     86H      ;CREATE
DM      'CREATE'
DEFW     IDDOT-6
CREAT:  DEFW     DOCOL
        DEFW     DFIND
        DEFW     ZBRAN     ;IF
        DEFW     CREA1-$
        DEFW     DROP
        DEFW     NFA
        DEFW     IDDOT
        DEFW     LIT
        DEFW     4
        DEFW     MESS
        DEFW     SPACE     ;ENDIF
CREA1:  DEFW     HERE
        DEFW     DUP
        DEFW     CAT
        DEFW     WIDTH
        DEFW     AT
        DEFW     MIN
        DEFW     ONEP
        DEFW     ALLOT
        DEFW     DUP
        DEFW     LIT
        DEFW     0A0H
        DEFW     TOGGL
        DEFW     HERE

```

```

DEFW  ONEMIN
DEFW  LIT
DEFW  80H
DEFW  TOGGL
DEFW  LATES
DEFW  COMMA
DEFW  CURR
DEFW  AT
DEFW  STORE
DEFW  HERE
DEFW  TWOP
DEFW  COMMA
DEFW  SEMIS

;
DEFB  0C9H          ;[COMPILE]
DM    '[COMPILE]'
DEFW  CREAT-9
BCOMP: DEFW  DOCOL
DEFW  DFIND
DEFW  ZEQU
DEFW  ZERO
DEFW  QERR
DEFW  DROP
DEFW  CFA
DEFW  COMMA
DEFW  SEMIS

;
DEFB  0C7H          ;LITERAL
DM    'LITERAL'
DEFW  BCOMP-0CH
LITER: DEFW  DOCOL
DEFW  STATE
DEFW  AT
DEFW  ZBRAN        ;IF
DEFW  LITE1-$
DEFW  COMP
DEFW  LIT
DEFW  COMMA        ;ENDIF
LITE1: DEFW  SEMIS
;
DEFB  0C8H          ;DLITERAL
DM    'DLITERAL'
DEFW  LITER-0AH
DLITE: DEFW  DOCOL
DEFW  STATE
DEFW  AT
DEFW  ZBRAN        ;IF
DEFW  DLIT1-$
DEFW  SWAP
DEFW  LITER
DEFW  LITER        ;ENDIF
DLIT1: DEFW  SEMIS
;
DEFB  86H          ;?STACK
DM    '?STACK'
DEFW  DLITE-0BH
QSTAC: DEFW  DOCOL
DEFW  SPAT
DEFW  SZERO
DEFW  AT
DEFW  SWAP
DEFW  ULESS
DEFW  ONE
DEFW  QERR

```

```

DEFW SPAT
DEFW HERE
DEFW LIT
DEFW 80H
DEFW PLUS
DEFW ULESS
DEFW LIT
DEFW 7
DEFW QERR
DEFW SEMIS
;
DEFB 89H ;INTERPRET
DM 'INTERPRET'
DEFW QSTAC-9
INTER: DEFW DOCOL
INTE1: DEFW DFIND ;BEGIN
DEFW ZBRAN ;IF
DEFW INTE2-$
DEFW STATE
DEFW AT
DEFW LESS
DEFW ZBRAN ;IF
DEFW INTE3-$
DEFW CFA
DEFW COMMA
DEFW BRAN ;ELSE
DEFW INTE4-$
INTE3: DEFW CFA
DEFW EXEC ;ENDIF
INTE4: DEFW QSTAC
DEFW BRAN ;ELSE
DEFW INTE5-$
INTE2: DEFW HERE
DEFW NUMB
DEFW DPL
DEFW AT
DEFW ONEP
DEFW ZBRAN ;IF
DEFW INTE6-$
DEFW DLITE
DEFW BRAN ;ELSE
DEFW INTE7-$
INTE6: DEFW DROP
DEFW LITER ;ENDIF
INTE7: DEFW QSTAC ;ENDIF
INTE5: DEFW BRAN ;AGAIN
DEFW INTE1-$
;
DEFB 89H ;IMMEDIATE
DM 'IMMEDIATE'
DEFW INTER-0CH
IMMED: DEFW DOCOL
DEFW LATES
DEFW LIT
DEFW 40H
DEFW TOGGL
DEFW SEMIS
;
DEFB 8AH ;VOCABULARY
DM 'VOCABULARY'
DEFW IMMED-0CH
VOCAB: DEFW DOCOL
DEFW BUILD
DEFW LIT

```

```

DEFW 0A081H
DEFW COMMA
DEFW CURR
DEFW AT
DEFW CFA
DEFW COMMA
DEFW HERE
DEFW VOCL
DEFW AT
DEFW COMMA
DEFW VOCL
DEFW STORE
DEFW DOES
DOVOC: DEFW TWOP
DEFW CONT
DEFW STORE
DEFW SEMIS
;
DEFB 0C5H ;FORTH
DM 'FORTH'
DEFW VOCAB-0DH
FORTH: DEFW DODOE
DEFW DOVOC
DEFW 0A081H
DEFW TASK-7 ;COLD START VALUE ONLY.
; CHANGED EACH TIME A DEF IS APPENDED
; TO THE FORTH VOCABULARY
DEFW 0 ;END OF VOCABULARY LIST
;
DEFB 8BH ;DEFINITIONS
DM 'DEFINITIONS'
DEFW FORTH-8
DEFIN: DEFW DOCOL
DEFW CONT
DEFW AT
DEFW CURR
DEFW STORE
DEFW SEMIS
;
DEFB 0C1H ;(
DM '('
DEFW DEFIN-0EH
PAREN: DEFW DOCOL
DEFW LIT
DEFW 29H
DEFW WORD
DEFW SEMIS
;
DEFB 84H ;QUIT
DM 'QUIT'
DEFW PAREN-4
QUIT: DEFW DOCOL
DEFW ZERO
DEFW BLK
DEFW STORE
DEFW LBRAC
QUIT1: DEFW RPSTO ;BEGIN
DEFW CR
DEFW QUERY
DEFW INTER
DEFW STATE
DEFW AT
DEFW ZEQU
DEFW ZBRAN ;IF

```

```

      DEFW    QUIT2-$
      DEFW    PDOTQ
      DEFB    2
      DB      'ok'           ;ENDIF
QUIT2: DEFW    BRAN           ;AGAIN
      DEFW    QUIT1-$
;
      DEFB    85H           ;ABORT
      DM      'ABORT'
      DEFW    QUIT-7
ABORT: DEFW    DOCOL
      DEFW    SPSTO
      DEFW    DEC
      DEFW    QSTAC
      DEFW    CR
      DEFW    DOTCPU
      DEFW    PDOTQ
      DEFB    0EH           ;count of CHR's to follow
      DB      'fig-FORTH '
      DEFB    FIGREL+30H,ADOT,FIGREV+30H,USRVER
      DEFW    FORTH
      DEFW    DEFIN
      DEFW    QUIT
;
WRM:   LD      BC,WRM1
      JNEXT
WRM1:  DEFW    WARM
;
      DEFB    84H           ;WARM
      DM      'WARM'
      DEFW    ABORT-8
WARM:  DEFW    DOCOL
      DEFW    MTBUF
      DEFW    ABORT
;
CLD:   LD      HL,(BDOS+1)   ;/
      LD      L,0           ;/ (HL)<--FBASE
      LD      (LIMIT+2),HL  ;/ set LIMIT
      LD      DE,BUFSIZ     ;/ (DE)<--total disc buffer size
      OR      A             ;/ clr carry
      SBC     HL,DE         ;/ (HL)<--addr. of 1st disc buffer
      LD      (FIRST+2),HL  ;/ set FIRST
      LD      (USE+2),HL    ;/ set USE
      LD      (PREV+2),HL   ;/ set PREV
      LD      (BUF1),HL     ;/
      LD      DE,US         ;/ (DE)<--user variable space
      SBC     HL,DE         ;/ (HL)<--INITR0
      LD      (UPINIT),HL   ;/
      LD      (R0INIT),HL   ;/
      LD      (UP),HL       ;/
      LD      (RPP),HL      ;/
      LD      DE,RTS        ;/ (DE)<--return stack & terminal buffer space
      SBC     HL,DE         ;/ (HL)<--INIT$0
      LD      (S0INIT),HL   ;/
      LD      (TIBINI),HL   ;/
      LD      SP,HL         ;/
      LD      BC,CLD1
      LD      IX,NEXT       ; POINTER TO NEXT
      LD      IY,HPUSH      ; POINTER TO HPUSH
      JNEXT
;
CLD1:  DEFW    COLD
;
      DEFB    84H           ;COLD

```

```

DM      'COLD'
DEFW    WARM-7
COLD:   DEFW    DOCOL
        DEFW    MTBUF
        DEFW    ONE,RECADR      ;AvdH
        DEFW    STORE
        DEFW    LIT,BUF1
        DEFW    AT              ;/
        DEFW    USE,STORE
        DEFW    LIT,BUF1
        DEFW    AT              ;/
        DEFW    PREV,STORE
        DEFW    DRZER
        DEFW    ZERO            ;/
        DEFW    LIT,EPRINT
        DEFW    CSTOR          ;/
;
        DEFW    LIT
        DEFW    ORIG+12H
        DEFW    LIT
        DEFW    UP
        DEFW    AT
        DEFW    LIT
        DEFW    6
        DEFW    PLUS
        DEFW    LIT
        DEFW    10H
        DEFW    CMOVE
        DEFW    LIT
        DEFW    ORIG+0CH
        DEFW    AT
        DEFW    LIT
        DEFW    FORTH+6
        DEFW    STORE
        DEFW    FCB            ;/A
        DEFW    LIT,OPNFIL      ;/A open mass storage
        DEFW    BDOS            ;/A
        DEFW    LIT,0FFH        ;/A
        DEFW    EQUAL           ;/A file present?
        DEFW    ZBRAN,CLD2-$    ;/A
        DEFW    ZERO            ;/A
        DEFW    WARN,STORE      ;/A
        DEFW    CR,PDOTQ        ;/A
        DEFB     7              ;/A
        DB       'No file'      ;/A
CLD2:   DEFW    ABORT
;
        DEFB     84H            ;S->D
        DM       'S->D'
        DEFW    COLD-7
STOD:   DEFW    $+2
        POP      DE
        LD       HL,0
        BIT      7,D            ;/ # NEGATIVE?
        JR       Z,STOD1        ; NO
        DEC      HL             ; YES, EXTEND SIGN
STOD1:  JP       DPUSH           ; ( n1--d1L d1H)
;
        DEFB     82H            ;+-
        DM       '+-'
        DEFW    STOD-7
PM:     DEFW    DOCOL
        DEFW    ZLESS
        DEFW    ZBRAN          ;IF

```



```

      DEFW    PM1-$
      DEFW    MINUS      ;ENDIF
PM1:  DEFW    SEMIS
;
      DEFB    83H      ;D+-
      DM      'D+- '
      DEFW    PM-5
DPM:  DEFW    DOCOL
      DEFW    ZLESS
      DEFW    ZBRAN      ;IF
      DEFW    DPM1-$
      DEFW    DMINU      ;ENDIF
DPM1: DEFW    SEMIS
;
      DEFB    83H      ;ABS
      DM      'ABS '
      DEFW    DPM-6
ABS:  DEFW    DOCOL
      DEFW    DUP
      DEFW    PM
      DEFW    SEMIS
;
      DEFB    84H      ;DABS
      DM      'DABS '
      DEFW    ABS-6
DABS: DEFW    DOCOL
      DEFW    DUP
      DEFW    DPM
      DEFW    SEMIS
;
      DEFB    83H      ;MIN
      DM      'MIN'
      DEFW    DABS-7
MIN:  DEFW    DOCOL,TDUP
      DEFW    GREAT
      DEFW    ZBRAN      ;IF
      DEFW    MIN1-$
      DEFW    SWAP      ;ENDIF
MIN1: DEFW    DROP
      DEFW    SEMIS
;
      DEFB    83H      ;MAX
      DM      'MAX'
      DEFW    MIN-6
MAX:  DEFW    DOCOL
      DEFW    TDUP
      DEFW    LESS
      DEFW    ZBRAN      ;IF
      DEFW    MAX1-$
      DEFW    SWAP      ;ENDIF
MAX1: DEFW    DROP
      DEFW    SEMIS
;
      DEFB    82H      ;M*
      DM      'M*'
      DEFW    MAX-6
MSTAR: DEFW    DOCOL,TDUP
      DEFW    XORR
      DEFW    TOR
      DEFW    ABS
      DEFW    SWAP
      DEFW    ABS
      DEFW    USTAR
      DEFW    FROMR

```

```

DEFW DPM
DEFW SEMIS

;
DEFB 82H ;M/
DM 'M/'
DEFW MSTAR-5
MSLAS: DEFW DOCOL
DEFW OVER
DEFW TOR
DEFW TOR
DEFW DABS
DEFW RR
DEFW ABS
DEFW USLAS
DEFW FROMR
DEFW RR
DEFW XORR
DEFW PM
DEFW SWAP
DEFW FROMR
DEFW PM
DEFW SWAP
DEFW SEMIS

;
DEFB 81H ; *
DM '*'
DEFW MSLAS-5
STAR: DEFW $+2
EXX ;/ SAVE IP
POP HL
POP DE
LD A,L ;/
LD C,H ;/
LD B,10H ;/
LD HL,0 ;/
STAR1: SRL C ;/
RRA ;/ SRL CA (MPCATOR)
JR NC,STAR2 ;/ LSB (CA)=0?
ADD HL,DE ;/ NO, ADD MPCANT TO HL
STAR2: SLA E ;/
RL D ;/ SLA DE (MPCANT 2 *)
DJNZ STAR1 ;/ DO ALL 16 BITS
PUSH HL ;/ (S1)<--PRODUCT
EXX ;/ RESTORE IP
JNEXT

;
DEFB 84H ; /MOD
DM '/MOD'
DEFW STAR-4
SLMOD: DEFW $+2
POP DE ;/ DIVISOR
POP HL ;/ DIVIDEND
PUSH BC ;/ SAVE IP
XOR A ;/ RESET NEGATE FLAG
EX AF,AF' ;/ ALT SET
LD A,D ;/
OR E ;/ DIV BY 0?
JR NZ,SLMOD1 ;/
LD HL,-1 ;/ YES
LD D,H ;/
LD E,L ;/ QUOT & REM <-- -1
JR SLMOD7 ;/ EXIT
SLMOD1: BIT 7,D ;/ DIVISOR NEGATIVE?
JR Z,SLMOD2 ;/

```

```

LD      A,E          ;/ YES
CPL                      ;/
LD      E,A          ;/
LD      A,D          ;/
CPL                      ;/
LD      D,A          ;/
INC      DE          ;/ (DE)<--(DE)'s 2's COMPLEMENT
EX      AF,AF'        ;/ STD SET
SCF                      ;/ SET NEGATE FLAG
EX      AF,AF'        ;/ ALT SET
SLMOD2: BIT      7,H    ;/ DIVIDEND NEGATIVE?
JR      Z,SLMOD3      ;/
LD      A,L          ;/ YES
CPL                      ;/
LD      L,A          ;/
LD      A,H          ;/
CPL                      ;/
LD      H,A          ;/
INC      HL          ;/ (HL)<--(HL)'s 2's COMPLEMENT
EX      AF,AF'        ;/ STD SET
CCF                      ;/ NEGATE FLAG
INC      A          ;/ DIVIDEND SIGN FLAG
EX      AF,AF'        ;/ ALT SET
SLMOD3: LD      A,L    ;/
LD      C,H          ;/ (CA)<--DIVIDEND
LD      HL,0         ;/ PRIME REMAINDER
LD      B,10H        ;/ LOOP COUNTER
SLMOD4: RLA          ;/
RL      C            ;/ RL CA
ADC      HL,HL        ;/ (HL)<--(HL) 2 * CARRY +
SBC      HL,DE        ;/ UNDERFLOW?
JR      NC,SLMOD5     ;/ NO
ADD      HL,DE        ;/ YES, RESTORE REMAINDER
SLMOD5: CCF          ;/
DJNZ     SLMOD4       ;/ DO ALL 16 BITS
RLA          ;/
RL      C            ;/ RL CA
LD      E,A          ;/
LD      D,C          ;/ (DE)<--QUOTIENT
EX      AF,AF'        ;/ STD SET
JR      Z,SLMOD6     ;/ DIVIDEND POSITIVE
LD      A,L          ;/
CPL                      ;/
LD      L,A          ;/
LD      A,H          ;/
CPL                      ;/
LD      H,A          ;/
INC      HL          ;/ REM GETS DIVIDEND'S SIGN
SLMOD6: JR      NC,SLMOD7 ;/ QUOTIENT POSITIVE
LD      A,E          ;/
CPL                      ;/
LD      E,A          ;/
LD      A,D          ;/
CPL                      ;/
LD      D,A          ;/
INC      DE          ;/ NEGATIVE QUOTIENT
SLMOD7: POP      BC    ;/ RESTORE IP
EX      DE,HL        ;/ (S2)<--REMAINDER
JP      DPUSH        ;/ (S1)<--QUOTIENT
;
DEFB     81H          ; /
DM      '/'
DEFW     SLMOD-7
SLASH:   DEFW     $+2

```

```

POP      DE          ;/ DIVISOR
POP      HL          ;/ DIVIDEND
PUSH     BC          ;/ SAVE IP
XOR      A           ;/ RESET NEG. FLAG
EX       AF,AF'      ;/ ALT SET
LD       A,D         ;/
OR       E           ;/ DIV BY 0?
JR       NZ,SLASH1   ;/
LD       HL,-1       ;/ YES, SET QUOTIENT TO -1
JR       SLASH6      ;/ EXIT
SLASH1:  BIT        7,D ;/ DIVISOR NEGATIVE?
JR       Z,SLASH2    ;/
LD       A,E         ;/ YES
CPL      ;/
LD       E,A         ;/
LD       A,D         ;/
CPL      ;/
LD       D,A         ;/
INC      DE          ;/ (DE)<--(DE)'s 2's COMPLEMENT
EX       AF,AF'      ;/ STD SET
SCF      ;/ SET NEG. FLAG
EX       AF,AF'      ;/ ALT SET
SLASH2:  BIT        7,H ;/ DIVIDEND NEGATIVE?
JR       Z,SLASH3    ;/
LD       A,L         ;/ YES
CPL      ;/
LD       L,A         ;/
LD       A,H         ;/
CPL      ;/
LD       H,A         ;/
INC      HL          ;/ (HL)<--(HL)'s 2's COMPLEMENT
EX       AF,AF'      ;/ STD SET
CCF      ;/ NEG. FLAG
EX       AF,AF'      ;/ ALT SET
SLASH3:  LD         A,L ;/
LD       C,H         ;/ (CA)<--DIVIDEND
LD       HL,0        ;/
LD       B,10H       ;/ LOOP COUNTER
SLASH4:  RLA         ;/
RL       C           ;/ RL CA
ADC      HL,HL        ;/ (HL)<--(HL) 2 * CARRY +
SBC      HL,DE        ;/ UNDERFLOW?
JR       NC,SLASH5   ;/ NO
ADD      HL,DE        ;/ YES, RESTORE REMAINDER
SLASH5:  CCF         ;/
DJNZ     SLASH4      ;/ DO ALL 16 BITS
RLA      ;/
RL       C           ;/ RL CA
LD       L,A         ;/
LD       H,C         ;/ (HL)<--QUOTIENT
EX       AF,AF'      ;/ STD SET
JR       NC,SLASH6   ;/ POSITIVE QUOTIENT
LD       A,L         ;/
CPL      ;/
LD       L,A         ;/
LD       A,H         ;/
CPL      ;/
LD       H,A         ;/
INC      HL          ;/ NEGATIVE QUOTIENT
SLASH6:  POP        BC ;/ RESTORE IP
JHPUSH
;
DEFB     83H         ; MOD
DM       'MOD'

```

```

MODD:  DEFW  SLASH-4
        DEFW  DOCOL
        DEFW  SLMOD
        DEFW  DROP
        DEFW  SEMIS

;
        DEFB  85H          ; */MOD
        DM    '*/MOD'
        DEFW  MODD-6
SSMOD:  DEFW  DOCOL
        DEFW  TOR
        DEFW  MSTAR
        DEFW  FROMR
        DEFW  MSLAS
        DEFW  SEMIS

;
        DEFB  82H          ; */
        DM    '*/'
        DEFW  SSMOD-8
SSLA:   DEFW  DOCOL
        DEFW  SSMOD
        DEFW  SWAP
        DEFW  DROP
        DEFW  SEMIS

;
        DEFB  85H          ; M/MOD
        DM    'M/MOD'
        DEFW  SSLA-5
MSMOD:  DEFW  DOCOL
        DEFW  TOR
        DEFW  ZERO
        DEFW  RR
        DEFW  USLAS
        DEFW  FROMR
        DEFW  SWAP
        DEFW  TOR
        DEFW  USLAS
        DEFW  FROMR
        DEFW  SEMIS

;
;      Block moved down 2 pages
;

        DEFB  86H          ; (LINE)
        DM    '(LINE)'
        DEFW  MSMOD-8
PLINE:  DEFW  DOCOL
        DEFW  TOR
        DEFW  LIT
        DEFW  40H
        DEFW  BBUF
        DEFW  SSMOD
        DEFW  FROMR
        DEFW  BSCR
        DEFW  STAR
        DEFW  PLUS
        DEFW  BLOCK
        DEFW  PLUS
        DEFW  LIT
        DEFW  40H
        DEFW  SEMIS

;
        DEFB  85H          ; .LINE
        DM    '.LINE'
        DEFW  PLINE-9

```

```

DLIN:  DEFW  DOCOL
      DEFW  PLINE
      DEFW  DTRAI
      DEFW  TYPE
      DEFW  SEMIS
;
      DEFB  87H          ;MESSAGE
      DM    'MESSAGE '
      DEFW  DLINE-8
MESS:  DEFW  DOCOL
      DEFW  WARN
      DEFW  AT
      DEFW  ZBRAN        ;IF
      DEFW  MESS1-$
      DEFW  DDUP
      DEFW  ZBRAN        ;IF
      DEFW  MESS2-$
      DEFW  LIT
      DEFW  4            ;1st MESSAGE SCREEN
      DEFW  OFSET
      DEFW  AT
      DEFW  BSCR
      DEFW  SLASH
      DEFW  SUBB
      DEFW  DLINE
      DEFW  SPACE        ;ENDIF
MESS2: DEFW  BRAN        ;ELSE
      DEFW  MESS3-$
MESS1: DEFW  PDOTQ
      DEFB  6
      DB    'MSG # '
      DEFW  DOT          ;ENDIF
MESS3: DEFW  SEMIS
;
      DEFB  82H          ;P@
      DM    'P@'
      DEFW  MESS-0AH
PTAT:  DEFW  $+2
      EXX
      POP   BC           ;d SAVE REGISTERS
      IN    L,(C)        ;d (BC)--PORT#
      LD    H,0          ;d (L)--DATA BYTE
      PUSH  HL
      EXX
      JNEXT              ;d RESTORE REGISTERS
;
      DEFB  82H          ;P!
      DM    'P!'
      DEFW  PTAT-5
PTSTO: DEFW  $+2
      EXX
      POP   BC           ;d SAVE REGISTERS
      POP   HL           ;d (C)--PORT#
      OUT   (C),L        ;d (L)--DATA BYTE
      EXX
      JNEXT              ;d RESTORE REGISTERS
;
      FORM
*INCLUDE DISCIO.FTH
      FORM
*INCLUDE CONPRTIO.FTH
      FORM
;
      DEFB  0C1H          ; ' (tick)

```

```

    DEFB    0A7H
    DEFW    ARROW-6
TICK:  DEFW    DOCOL
    DEFW    DFIND
    DEFW    ZEQU
    DEFW    ZERO
    DEFW    QERR
    DEFW    DROP
    DEFW    LITER
    DEFW    SEMIS
;
    DEFB    86H                ;FORGET
    DM      'FORGET'
    DEFW    TICK-4
FORG:  DEFW    DOCOL
    DEFW    CURR
    DEFW    AT
    DEFW    CONT
    DEFW    AT
    DEFW    SUBB
    DEFW    LIT
    DEFW    18H
    DEFW    QERR
    DEFW    TICK
    DEFW    DUP
    DEFW    FENCE
    DEFW    AT
    DEFW    unless            ;/ FORGET >8000h nw o.k.
    DEFW    LIT
    DEFW    15H
    DEFW    QERR
    DEFW    DUP
    DEFW    NFA
    DEFW    DP
    DEFW    STORE
    DEFW    LFA
    DEFW    AT
    DEFW    CONT
    DEFW    AT
    DEFW    STORE
    DEFW    SEMIS
;
    DEFB    84H                ;BACK
    DM      'BACK'
    DEFW    FORG-9
BACK:  DEFW    DOCOL
    DEFW    HERE
    DEFW    SUBB
    DEFW    COMMA
    DEFW    SEMIS
;
    DEFB    0C5H                ;BEGIN
    DM      'BEGIN'
    DEFW    BACK-7
BEGIN: DEFW    DOCOL
    DEFW    QCOMP
    DEFW    HERE
    DEFW    ONE
    DEFW    SEMIS
;
    DEFB    0C5H                ;ENDIF
    DM      'ENDIF'
    DEFW    BEGIN-8
ENDIFF: DEFW    DOCOL

```

```

      DEFW  QCOMP
      DEFW  TWO
      DEFW  QPAIR
      DEFW  HERE
      DEFW  OVER
      DEFW  SUBB
      DEFW  SWAP
      DEFW  STORE
      DEFW  SEMIS
;
      DEFB  0C4H          ;THEN
      DM    'THEN'
      DEFW  ENDIFF-8
THEN:  DEFW  DOCOL
      DEFW  ENDIFF
      DEFW  SEMIS
;
      DEFB  0C2H          ;DO
      DM    'DO'
      DEFW  THEN-7
DO:    DEFW  DOCOL
      DEFW  COMP
      DEFW  XDO
      DEFW  HERE
      DEFW  THREE
      DEFW  SEMIS
;
      DEFB  0C4H          ;LOOP
      DM    'LOOP'
      DEFW  DO-5
LOOP:  DEFW  DOCOL
      DEFW  THREE
      DEFW  QPAIR
      DEFW  COMP
      DEFW  XLOOP
      DEFW  BACK
      DEFW  SEMIS
;
      DEFB  0C5H          ;+LOOP
      DM    '+LOOP'
      DEFW  LOOP-7
PLOOP: DEFW  DOCOL
      DEFW  THREE
      DEFW  QPAIR
      DEFW  COMP
      DEFW  XPLOO
      DEFW  BACK
      DEFW  SEMIS
;
      DEFB  0C5H          ;UNTIL
      DM    'UNTIL'
      DEFW  PLOOP-8
UNTIL: DEFW  DOCOL
      DEFW  ONE
      DEFW  QPAIR
      DEFW  COMP
      DEFW  ZBRAN
      DEFW  BACK
      DEFW  SEMIS
;
      DEFB  0C3H          ;END
      DM    'END'
      DEFW  UNTIL-8
ENDD:  DEFW  DOCOL

```



```

      DEFW  UNTIL
      DEFW  SEMIS
;
      DEFB  0C5H          ;AGAIN
      DM    'AGAIN'
      DEFW  ENDD-6
AGAIN:  DEFW  DOCOL
      DEFW  ONE
      DEFW  QPAIR
      DEFW  COMP
      DEFW  BRAN
      DEFW  BACK
      DEFW  SEMIS
;
      DEFB  0C6H          ;REPEAT
      DM    'REPEAT'
      DEFW  AGAIN-8
REPEA:  DEFW  DOCOL
      DEFW  TOR
      DEFW  TOR
      DEFW  AGAIN
      DEFW  FROMR
      DEFW  FROMR
      DEFW  TWOMIN        ;/
      DEFW  ENDIFF
      DEFW  SEMIS
;
      DEFB  0C2H          ;IF
      DM    'IF'
      DEFW  REPEA-9
IFF:    DEFW  DOCOL
      DEFW  COMP
      DEFW  ZBRAN
      DEFW  HERE
      DEFW  ZERO
      DEFW  COMMA
      DEFW  TWO
      DEFW  SEMIS
;
      DEFB  0C4H          ;ELSE
      DM    'ELSE'
      DEFW  IFF-5
ELSEE:  DEFW  DOCOL
      DEFW  TWO
      DEFW  QPAIR
      DEFW  COMP
      DEFW  BRAN
      DEFW  HERE
      DEFW  ZERO
      DEFW  COMMA
      DEFW  SWAP
      DEFW  TWO
      DEFW  ENDIFF
      DEFW  TWO
      DEFW  SEMIS
;
      DEFB  0C5H          ;WHILE
      DM    'WHILE'
      DEFW  ELSEE-7
WHILE:  DEFW  DOCOL
      DEFW  IFF
      DEFW  TWOP
      DEFW  SEMIS
;

```

```

      DEFB      86H          ;SPACES
      DM        ' SPACES '
      DEFW      WHILE-8
SPACS: DEFW      DOCOL
      DEFW      ZERO
      DEFW      MAX
      DEFW      DDUP
      DEFW      ZBRAN        ;IF
      DEFW      SPAX1-$
      DEFW      ZERO
      DEFW      XDO          ;DO
SPAX2: DEFW      SPACE
      DEFW      XLOOP        ;LOOP  ENDIF
      DEFW      SPAX2-$
SPAX1: DEFW      SEMIS
;
      DEFB      82H          ;<#
      DM        '<#'
      DEFW      SPACS-9
BDIGS: DEFW      DOCOL
      DEFW      PAD
      DEFW      HLD
      DEFW      STORE
      DEFW      SEMIS
;
      DEFB      82H          ;#>
      DM        '#>'
      DEFW      BDIGS-5
EDIGS: DEFW      DOCOL
      DEFW      DROP
      DEFW      DROP
      DEFW      HLD
      DEFW      AT
      DEFW      PAD
      DEFW      OVER
      DEFW      SUBB
      DEFW      SEMIS
;
      DEFB      84H          ;SIGN
      DM        'SIGN'
      DEFW      EDIGS-5
SIGN:  DEFW      DOCOL
      DEFW      ROT
      DEFW      ZLESS
      DEFW      ZBRAN        ;IF
      DEFW      SIGN1-$
      DEFW      LIT
      DEFW      2DH
      DEFW      HOLD        ;ENDIF
SIGN1: DEFW      SEMIS
;
      DEFB      81H          ;#
      DM        '# '
      DEFW      SIGN-7
DIG:   DEFW      DOCOL
      DEFW      BASE
      DEFW      AT
      DEFW      MSMOD
      DEFW      ROT
      DEFW      LIT
      DEFW      9
      DEFW      OVER
      DEFW      LESS
      DEFW      ZBRAN        ;IF

```

```

      DEFW    DIG1-$
      DEFW    LIT
      DEFW    7
      DEFW    PLUS          ;ENDIF
DIG1:  DEFW    LIT
      DEFW    30H
      DEFW    PLUS
      DEFW    HOLD
      DEFW    SEMIS
;
      DEFB    82H          ;#S
      DM      '#S'
      DEFW    DIG-4
DIGS:  DEFW    Docol
DIGS1: DEFW    DIG          ;BEGIN
      DEFW    TDUP          ;/
      DEFW    ORR
      DEFW    ZEQU
      DEFW    ZBRAN          ;UNTIL
      DEFW    DIGS1-$
      DEFW    SEMIS
;
      DEFB    83H          ;D.R
      DM      'D.R'
      DEFW    DIGS-5
DDOTR: DEFW    Docol
      DEFW    TOR
      DEFW    SWAP
      DEFW    OVER
      DEFW    DABS
      DEFW    BDIGS
      DEFW    DIGS
      DEFW    SIGN
      DEFW    EDIGS
      DEFW    FROMR
      DEFW    OVER
      DEFW    SUBB
      DEFW    SPACS
      DEFW    TYPE
      DEFW    SEMIS
;
      DEFB    82H          ;.R
      DM      '.R'
      DEFW    DDOTR-6
DOTR:  DEFW    Docol
      DEFW    TOR
      DEFW    STOD
      DEFW    FROMR
      DEFW    DDOTR
      DEFW    SEMIS
;
      DEFB    82H          ;D.
      DM      'D.'
      DEFW    DOTR-5
DDOT:  DEFW    Docol
      DEFW    ZERO
      DEFW    DDOTR
      DEFW    SPACE
      DEFW    SEMIS
;
      DEFB    81H          ;.
      DM      '.'
      DEFW    DDOT-5
DOT:   DEFW    Docol

```

```

      DEFW      STOD
      DEFW      DDOT
      DEFW      SEMIS
;
      DEFB      81H          ;?
      DM        '? '
      DEFW      DOT-4
      QUES:     DEFW      DOCOL
      DEFW      AT
      DEFW      DOT
      DEFW      SEMIS
;
      DEFB      82H          ;U.
      DM        'U. '
      DEFW      QUES-4
      UDOT:     DEFW      DOCOL
      DEFW      ZERO
      DEFW      DDOT
      DEFW      SEMIS
;
      DEFB      85H          ;VLIST
      DM        'VLIST'
      DEFW      UDOT-5
      VLIST:    DEFW      DOCOL
      DEFW      LIT
      DEFW      80H
      DEFW      OUTT
      DEFW      STORE
      DEFW      CONT
      DEFW      AT
      DEFW      AT
      VLIS1:    DEFW      OUTT      ;BEGIN
      DEFW      AT
      DEFW      CSLL
      DEFW      GREAT
      DEFW      ZBRAN          ;IF
      DEFW      VLIS2-$
      DEFW      CR
      DEFW      ZERO
      DEFW      OUTT
      DEFW      STORE          ;ENDIF
      VLIS2:    DEFW      DUP
      DEFW      IDDOT
      DEFW      SPACE
      DEFW      SPACE
      DEFW      PFA
      DEFW      LFA
      DEFW      AT
      DEFW      DUP
      DEFW      ZEQU
      DEFW      QTERM
      DEFW      ORR
      DEFW      ZBRAN          ;UNTIL
      DEFW      VLIS1-$
      DEFW      DROP
      DEFW      SEMIS
;
      DEFB      83H          ;BYE
      DM        'BYE'
      DEFW      VLIST-8
      BYE:      DEFW      DOCOL      ;/A
      DEFW      FLUSH          ;/A
      DEFW      FCB,LIT        ;/E
      DEFW      10H,BDOS        ;/E close file

```

```

DEFW DROP ;/E discard directory code
DEFW ZERO,ZERO ;/A
DEFW BDOS ;/A return to CP/M
DEFW SEMIS ;/A won't get this far, just for pretty
;
DEFB 84H ;LIST
DM 'LIST'
DEFW BYE-6
LIST: DEFW DOCOL,DEC
DEFW CR,DUP
DEFW SCR,STORE
DEFW PDOTQ
DEFB 6
DB 'SCR # '
DEFW DOT
DEFW LIT,10H
DEFW ZERO,XDO
LIST1: DEFW CR,IDO
DEFW THREE ;/ WAS LIT,3
DEFW DOTR,SPACE
DEFW IDO,SCR
DEFW AT,DLINE
DEFW QTERM
DEFW ZBRAN,LIST2-$ ;IF
DEFW LEAVE
LIST2: DEFW XLOOP,LIST1-$ ;ENDIF
DEFW CR,SEMIS
;
DEFB 85H ;INDEX
DM 'INDEX'
DEFW LIST-7
INDEX: DEFW DOCOL
DEFW LIT,FF
DEFW EMIT,CR
DEFW ONEP,SWAP
DEFW XDO
INDE1: DEFW CR,IDO
DEFW THREE ;/ WAS LIT,3
DEFW DOTR,SPACE
DEFW ZERO,IDO
DEFW DLINE,QTERM
DEFW ZBRAN,INDE2-$ ;IF
DEFW LEAVE ;ENDIF
INDE2: DEFW XLOOP,INDE1-$
DEFW SEMIS
;
DEFB 85H ;TRIAD
DM 'TRIAD'
DEFW INDEX-8
TRIAD: DEFW DOCOL
DEFW LIT,FF
DEFW EMIT
DEFW THREE ;/ WAS LIT,3
DEFW SLASH
DEFW THREE ;/ WAS LIT,3
DEFW STAR
DEFW THREE ;/ WAS LIT,3
DEFW OVER,PLUS
DEFW SWAP,XDO
TRIA1: DEFW CR,IDO
DEFW LIST
DEFW QTERM
DEFW ZBRAN,TRIA2-$ ;IF
DEFW LEAVE

```

[illegible]

54/55

58.84.148.197:[LOG IN](#)

TOP NEW MORE HELP FIND:

Search