

.MAIN. -

NEW
IMPROVED
DEC 4 1980
3.5

```

      .PABS
      .PHEX
; *****
; * BALLY BASIC INTERPRETER *
; *
; * (C) JULY 1978 BALLY MFG *
; *
; * WRITTEN BY: JAY FENTON *
; *
; * BALLY BASIC IS BASED ON *
; * PALO ALTO TINY BASIC BY *
; * LICHEN WANG *
; *
; *****
; TINY BASIC INTERPRETER
; MACROS:
      .DEFINE TOKEN[TINDX,TGOTO]=
[
      .BYTE TINDX
      DEFF TGOTO
]
      .DEFINE DEFF[WORDY]=
[
      .BYTE (WORDY>8)!80H
      .BYTE WORDY&OFFH
]
      .DEFINE TSTCC[CAT,DOG]=
[
      RST 1
      .BYTE 'CAT'
      .BYTE DOG--1
]
      .DEFINE TSTCC[CAT1,DOG1]=
[
      RST 1
      .BYTE CAT1
      .BYTE DOG1--1
]
      .DEFINE ITEM[STRANG,JUMPTO]=
[
      .ASCII 'STRANG'
      DEFF JUMPTO
]
4E20 BOTSCR == 04E20H
4FEF TOPSCR == 04FEFH
;BOTRAM == 03000H
;DFTLMT == 03FFFFH
A000 BOTRAM == 0A000H
A70C DFTLMT == 0A70CH
2000 BOTROM == 02000H
;
0012 TAPEIO == 12H ; TAPE INTERFACE I-O POR
T
00FC STPBCT == 0FCH ; -# OF STOP BIT WINDOWS
FOR VALID DATA
;
000D CR == 0DH
001F RUBOUT == 1FH
002C COMMA == 44
0066 EDKEY == 66H

```

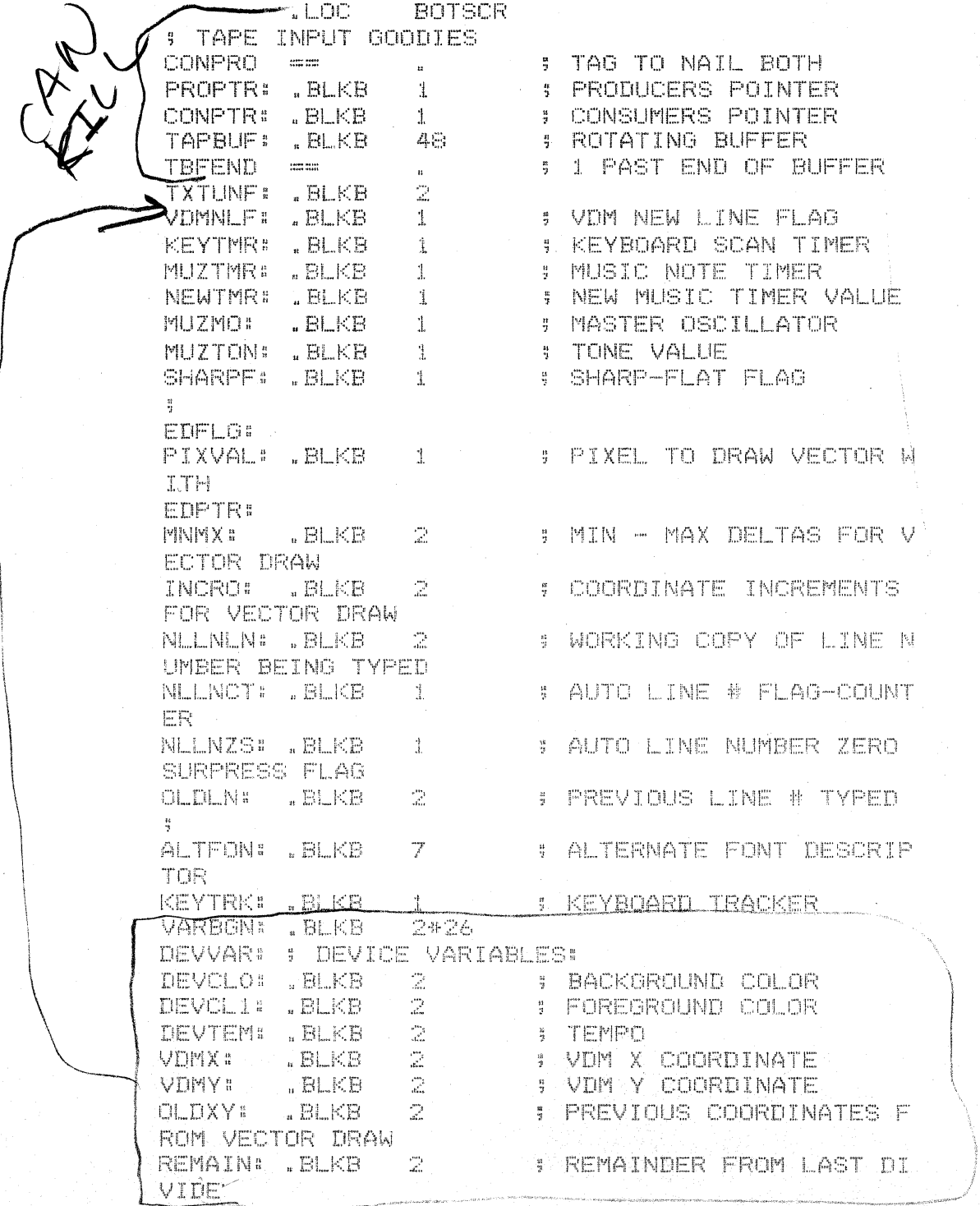
LEN
2FFF

.MAIN. --

```

0067      NLLN      ==      67H
          ; EQUATES FOR RESTART INSTRUCTIONS
0002      RSTEXP   ==      2      ; EXPR
0003      RSTOCH   ==      3      ; OUTCH
0004      RSTIGN   ==      4      ; IGNBLK
0005      RSTPAR   ==      5      ; PARN
0006      RSTFIN   ==      6      ; FINISH
          ;
4E20      .LOC      BOTSCR
          ; TAPE INPUT GOODIES
4E20      CONPRO   ==      .      ; TAG TO NAIL BOTH
4E20      PROPTR:  .BLKB   1      ; PRODUCERS POINTER
4E21      CONPTR:  .BLKB   1      ; CONSUMERS POINTER
4E22      TAPBUF:  .BLKB  48      ; ROTATING BUFFER
4E52      TBFEND   ==      .      ; 1 PAST END OF BUFFER
4E52      TXTUNF:  .BLKB   2      ;
4E54      VDMNLF:  .BLKB   1      ; VDM NEW LINE FLAG
4E55      KEYTMR:  .BLKB   1      ; KEYBOARD SCAN TIMER
4E56      MUZTMR:  .BLKB   1      ; MUSIC NOTE TIMER
4E57      NEWTMR:  .BLKB   1      ; NEW MUSIC TIMER VALUE
4E58      MUZMO:   .BLKB   1      ; MASTER OSCILLATOR
4E59      MUZTON:  .BLKB   1      ; TONE VALUE
4E5A      SHARPF:  .BLKB   1      ; SHARP-FLAT FLAG
          ;
4E5B      EDFLG:   .BLKB   1      ;
4E5B      PIXVAL:  .BLKB   1      ; PIXEL TO DRAW VECTOR W
          ITH
4E5C      EDPTR:   .BLKB   1      ;
4E5C      MNMX:   .BLKB   2      ; MIN - MAX DELTAS FOR V
          ECTOR DRAW
4E5E      INCRO:   .BLKB   2      ; COORDINATE INCREMENTS
          FOR VECTOR DRAW
4E60      NLLNLN:  .BLKB   2      ; WORKING COPY OF LINE N
          UMBER BEING TYPED
4E62      NLLNCT:  .BLKB   1      ; AUTO LINE # FLAG-COUNT
          ER
4E63      NLLNZS:  .BLKB   1      ; AUTO LINE NUMBER ZERO
          SURPRESS FLAG
4E64      OLDLN:   .BLKB   2      ; PREVIOUS LINE # TYPED
          ;
4E66      ALTFON:  .BLKB   7      ; ALTERNATE FONT DESCRIP
          TOR
4E6D      KEYTRK:  .BLKB   1      ; KEYBOARD TRACKER
4E6E      VARBGN:  .BLKB  2*26    ;
4EA2      DEVVAR:  ; DEVICE VARIABLES:
4EA2      DEVCL0:  .BLKB   2      ; BACKGROUND COLOR
4EA4      DEVCL1:  .BLKB   2      ; FOREGROUND COLOR
4EA6      DEVTEM:  .BLKB   2      ; TEMPO
4EA8      VDMX:    .BLKB   2      ; VDM X COORDINATE
4EAA      VDMY:    .BLKB   2      ; VDM Y COORDINATE
4EAC      OLDXY:   .BLKB   2      ; PREVIOUS COORDINATES F
          ROM VECTOR DRAW
4EAE      REMAIN:  .BLKB   2      ; REMAINDER FROM LAST DI
          VIDE
4EB0      BCDA1:   .BLKB   9      ; FIRST ARG AREA
    
```

*CAN
FIX*



.MAIN. -

```

4EB9          BCDA2:  .BLKB  9          ; SECOND ARG AREA
4EC2          TAPEST: .BLKB  1      ; TAPE STATUS
4EC3          CURRNT: .BLKB  2
4EC5          STKGOS: .BLKB  2
4EC7          VARNXT  ==
4EC7          STKINP: .BLKB  2
4EC9          LOPVAR: .BLKB  2
4ECB          LOPINC: .BLKB  2
4ECD          LOPLMT: .BLKB  2
4ECF          LOPLN:  .BLKB  2
4ED1          LOPPT:  .BLKB  2
4ED3          .BLKB  1
4ED4          BUFFER: .BLKB 104
4F3C          BUFEND  ==
4F3C          .BLKB  32
4F5C          STKLMT  ==
4FEF          .LOC   TOPSCR
4FEF          STACK  ==
A000          .LOC   BOTRAM
A000          TEXT:  .BLKB  2
2000          .LOC   BOTROM
2000          C3 2433 JMP     BEGIN    ; ** AUTOSTART CASSETTE

**
2003          80     PIXTBL: .BYTE  080H
2004          20     .BYTE  020H
2005          08     .BYTE  08H
2006          02     .BYTE  2H

; TRANSFER VECTORS TO RESTART ROUTINES
2007          C3 2A10 JMP     TSTCH    ; * RST 8
200A          C3 2761 JMP     EXPR     ; * RST 16
200D          C3 2C7C JMP     OUTCH    ; * RST 24
2010          C3 293C JMP     IGNBLK   ; * RST 32
2013          C3 286E JMP     PARN     ; * RST 40
2016          F1     POP     PSW     ; * RST 48
2017          C3 2929 JMP     FINISH
201A          C3 2D9C JMP     CHKID    ; ** LINK TO TAPE READER

OUTLINE **
; INITIAL VALUES FOR PARAMETER VECTOR
201D          0007   INIDEV: 0.WORD 7 2 ; BACKGROUND COLOR
201F          0000   2.WORD 0 4 ; FOREGROUND COLOR
2021          0003   4.WORD 3 6 ; MUSIC TEMPO
2023          FF83   6.WORD -77 8 ; VDM X COORDINATE
2025          0028   8.WORD 40 10 ; VDM Y COORDINATE
2027          0000   .WORD 0 12 ; RESET OLD X,Y FOR VECT

OR DRAW
; ASCII MESSAGES:
; MSG:  .ASCII  'BALLY BASIC'
;       .BYTE  CR
2029          7C     CKHLDE: MOV    A,H
202A          AA     XRA     D
202B          F2 202F JP     COMP
202E          EB     XCHG

; ...
202F          7C     COMP:  MOV    A,H
2030          BA     CMP     D

```

~~TAPEST: .BLKB 1~~

2
2
2
2

redefine
or
place
reader
in
other
place



push h
xor A
dsbc D
pop H
5

.MAIN. -

```

2031      C0                RNZ
2032      7D                MOV      A,L
2033      BB                CMP      E
2034      C9                RET
2035      574841543F        WHAT:   .ASCII  'WHAT?'
203A      0D                .BYTE   CR
203B      484F573F        HOW:    .ASCII  'HOW?'
203F      0D                .BYTE   CR
2040      534F525259        SORRY: .ASCII  'SORRY'
2045      0D                .BYTE   CR
;
; TABLE GIVING JUMP TO ADDRESS FOR COMMANDS
2046      257D        TOKJT:   .WORD   LIST
2048      2535        .WORD   CLRSCR
204A      254D        .WORD   RUN
204C      267F        .WORD   NEXT
204E      22BF        .WORD   LINEDR
2050      26EC        .WORD   IFF
2052      2572        .WORD   GOTO
2054      25F6        .WORD   GOSUB
2056      2616        .WORD   RETURN
2058      220D        .WORD   BOXDRW
205A      262C        .WORD   FOR
205C      2705        .WORD   INPUT
205E      25B5        .WORD   PRINT
; INTERRUPT VECTORS:
2060      20F6        JTAB:   .WORD   SIDINT  ;** TAPE INTERRUPT VECTO
R **
2062      20B0        ITAB:   .WORD   TBIINT  ;** MUSIC INTERRUPT VECT
OR **
; TABLE GIVING ASCII CHARS FOR TOKENS
2064      TOKTXT:
2064      4C4953        .ASCII  'LIS'
2067      D4           .BYTE   'T'+80H
2068      434C4541      .ASCII  'CLEA'
206C      D2           .BYTE   'R'+80H
206D      5255        .ASCII  'RU'
206F      CE           .BYTE   'N'+80H
2070      4E4558      .ASCII  'NEX'
2073      D4           .BYTE   'T'+80H
2074      4C494E      .ASCII  'LIN'
2077      C5           .BYTE   'E'+80H
2078      49           .BYTE   'I'
2079      C6           .BYTE   'F'+80H
207A      474F54      .ASCII  'GDT'
207D      CF           .BYTE   'O'+80H
207E      474F5355      .ASCII  'GOSU'
2082      C2           .BYTE   'B'+80H
2083      5245545552      .ASCII  'RETUR'
2088      CE           .BYTE   'N'+80H
2089      424F        .ASCII  'BO'
208B      D8           .BYTE   'X'+80H
208C      464F        .ASCII  'FO'
208E      D2           .BYTE   'R'+80H
208F      494E5055      .ASCII  'INPU'

```

.MAIN. -

```

2093      D4          .BYTE      'T'+80H
2094      5052494E   .ASCII     'PRIN'
2098      D4          .BYTE      'T'+80H
2099      5354445   .ASCII     'STE'
209C      D0          .BYTE      'P'+80H
209D      524E      .ASCII     'RN'
209F      C4          .BYTE      'D'+80H
20A0      54          .BYTE      'T'
20A1      CF          .BYTE      'O'+80H

;
; DEVICE VARIABLE TABLE
; THIS TABLE IS IN INVERSE ORDER OF APPEARENCE I
N MEMORY
0007      PARNUM == 7          ; 7 GUYS
20A2      DEVLST:
20A2      12          .BYTE      'R'-'@'
20A3      4D          .BYTE      'M'
20A4      18          .BYTE      'X'-'@'
20A5      59          .BYTE      'Y'
20A6      03          .BYTE      'C'-'@'
20A7      59          .BYTE      'Y'
20A8      03          .BYTE      'C'-'@'
20A9      58          .BYTE      'X'
20AA      0E          .BYTE      'N'-'@'
20AB      54          .BYTE      'T'
20AC      06          .BYTE      'F'-'@'
20AD      43          .BYTE      'C'
20AE      02          .BYTE      'B'-'@'
20AF      43          .BYTE      'C'

; TINY BASIC INTERRUPT ROUTINE
20B0      F5          TBIINT: PUSH  PSW          ; SAVE REGISTERS
20B1      C5          PUSH  B
20B2      D5          PUSH  D
20B3      E5          PUSH  H

; DEAL WITH KEYBOARD SCAN TIMER
20B4      21 4E55     LXI      H,KEYTMR
20B7      7E          MOV     A,M
20B8      A7          ANA     A
20B9      2801       JRZ     TBINO
20BB      35          DCR     M
20BC      23          TBINO: INX  H

; HAS MUSIC TIMER COUNTED DOWN?
20BD      7E          MOV     A,M
20BE      A7          ANA     A
20BF      2808       JRZ     TBIN1          ; YEP - PLAY NEXT NOTE
20C1      35          DCR     M          ; ELSE DECREMENT IT
20C2      201F       JRNZ    TBIN3       ; JUMP IF NOT NOW ZERO
20C4      AF          XRA     A
20C5      D311       OUT     TONEA      ; ELSE SILENCE
20C7      1818       JMPR    TBIN2

; MUSIC TIMER IS AT ZERO - ARE NEW PARAMETERS RE
ADY?
20C9      23          TBIN1: INX  H          ; STEP TO NEW TIMER VALU
E
20CA      B6          ORA     M          ; IS IT NON ZERO?

```

.MAIN. -

```

20CB 2816 JRZ TBIN3 ; JUMP IF NOT
20CD 2B DCX H ; ELSE SET OFFICIAL TIME
;
; R
20CE 77 MOV M,A
20CF 23 INX H
20D0 3600 MVI M,0 ; ZERO NEW TIMER VALUE A
;
; S FLAG
20D2 23 INX H
20D3 7E MOV A,M ; GET NEW M.O.
20D4 D310 OUT TONMO
20D6 3647 MVI M,0A2 ; RESET DEFAULT MASTER 0
;
; SC
20D8 23 INX H
20D9 7E MOV A,M ; AND NEW TONE
20DA D311 OUT TONEA
20DC A7 ANA A ; REST WANTED?
20DD 2804 JRZ TBIN3 ; YES - JUMP AROUND VOLU
;
; ME UPDATE
20DF 3E0F MVI A,15
20E1 D316 TBIN2: OUT VOLAB
; SET COLOR REGISTERS TO VALUES IN PARAMETER VAR
; S %0 AND %1
20E3 3A 4EA2 TBIN3: LDA DEVCL0
20E6 D304 OUT COL0L
20E8 D305 OUT COL1L
20EA 3A 4EA4 LDA DEVCL1
20ED D306 OUT COL2L
20EF D307 OUT COL3L
; DONE - RESTORE REGISTERS AND GO BACK
20F1 E1 INTDON: POP H
20F2 D1 POP D
20F3 C1 POP B
20F4 1833 JMPR ZRONK
;
; SERIAL INPUT DRIVER INTERRUPT ROUTINE
; THIS ROUTINE SAMPLES THE SERIAL INPUT BIT
; AND FORMS CHARACTERS WHICH ARE PLACED IN THE
; CIRCULAR INPUT BUFFER FOR CONSUMPTION BY BACKG
; ROUND LEVEL
;
; THIS ROUTINE USES THE ALTERNATE REGISTER SET
; WHERE B=STATE VARIABLE, C=CHARACTER ACCUMULATO
; R
; IF STATE VAR < 0 => WE ARE AWAITING -N STOP BI
; TS BEFORE
; WE BELIEVE ANYTHING
; IF STATE VAR = 0 => WE ARE LOOKING FOR A START
; BIT
; IF STATE VAR > 0 => WE ARE GETTING DATA BITS
20F6 F5 SIDINT: PUSH PSW
20F7 D9 EXX
; SAMPLE THE INPUT BIT
20F8 DB12 GETBIT: IN TAPEIO
20FA 1F RAR
20FB 79 MOV A,C ; SHIFT INTO ACCUMULATOR

```

out PAGE 7

```

20FC 1F          RAR
20FD 4F          MOV     C,A
20FE 78          MOV     A,B      ; WHAT STATE ARE WE IN?
20FF A7          ANA     A
2100 FA 210D     JM      SIDINO    ; 1 COUNTING STATE?
2103 2011        JRNZ    SIDIN1   ; JUMP IF IN MIDDLE OF C
                HAR
                ; WE ARE WAITING FOR A START BIT
2105 CB79        BIT     7,C      ; DID WE GET ONE?
2107 201F        JRNZ    SIDIN3   ; JUMP IF RIGHT
2109 0608        MVI     B,8      ; SET WAITING FOR 8 BITS

210B 181B        JMPR    SIDIN3   ; AND GO GONZO
                ; TAPE INPUT WAS RECENTLY TURNED ON, AND WE ARE
                ; WAITING FOR
                ; SEVERAL SEARCH WINDOWS IN A ROW TO EACH SHOW A
                ; 1 BIT
                ; IMPLYING THAT THIS BEASTIE IS REALLY WORKING
210D 04          SIDINO: INR     B      ; ASSUME WE GOT IT
210E CB79        BIT     7,C
2110 2016        JRNZ    SIDIN3   ; JUMP IF GOOD ASSUMPTIO

                N
                ; NO - RESET COUNTER AND WAIT SOME MORE
2112 06FC        MVI     B,STPBCT
2114 1812        JMPR    SIDIN3
                ; IN THE MIDDLE OF A CHARACTER...
                ; WAS THIS THE LAST BIT?
2116 1010        SIDIN1: DJNZ   SIDIN3  ; JUMP IF NOT
                ; WELCOME TO LAST BITSVILLE
2118 2A 4E20     LHL     CONPRO  ; GET POINTERS
211B 7D          MOV     A,L
211C CD 212C     CALL    BUMPTR  ; A=P+1
211F BC          CMP     H      ; IS C = P+1?
2120 2806        JRZ     SIDIN3   ; YEP - FULL - IGNORE
2122 32 4E20     STA     PROPTR  ; STUFF P+1
2125 264E        MVI     H,TAPBUF>8 ; POINT INTO BUF

                FER
2127 71          MOV     M,C
2128 D9          SIDIN3: EXX
2129 F1          ZRONK: POP    PSW
212A FB          EI
212B C9          RET
                ; SUBROUTINE TO INCREMENT 1 BYTE POINTER TO CIRC
                ; ULAR BUFFER
212C 3C          BUMPTR: INR     A      ; BUMP IT
212D FE52        CPI     TBFEND&OFFH
212F C0          RNZ
                ; QUIT IF NOT AT END
2130 3E22        MVI     A,TAPBUF&OFFH ; ELSE WRAP AROU

                ND
2132 C9          RET
                ; ROUTINE TO ESTABLISH TAPE UNIT AS INPUT DEVICE
2133 F3          TINPES: DI
2134 21 227      LXI     H,((TAPBUF&OFFH)<8)+(TAPBUF&OFFH
    )
    
```

Handwritten signature/initials

.MAIN. -

out

```

2137      22 4E20          SHLD   CONPRO
213A      D9              EXX
213B      AF              XRA     A
213C      32 4EA6        STA    DEVTEM
213F      3C              INR    A
2140      32 4EC2        STA    TAPEST
2143      06FC           MVI    B,STPBCT
2145      D9              EXX
2146      3E18           MVI    A,18H
2148      D30E           OUT    INMOD
214A      FB              EI
; SUBROUTINE TO RETURN ZERO STATUS IF CHARACTER
; IN A IS NL OR
; ' '
214B      FE3B          ATNL:   CPI    ' ' ; CHECK FOR CONTINUATION
;
214D      C8              RZ
214E      FE0D          CPI    CR ; AND FOR CR
2150      C9              RET
; COMMAND TO ESTABLISH TAPE UNIT AS INPUT DEVICE
2151      CD 2133        TINPUT: CALL  TINPES
2154      F7              RST    RSTFIN ; GO HOME
; COMMAND TO LIST STUFF ON CRT
2155      CD 2133        TLIST:  CALL  TINPES
2158      CD 2D9C        TLIST1: CALL  CHKIO ; GET CHARACTER
215B      DF              RST    RSTOCH ; PRINT IT
215C      18FA           JMPR   TLIST1 ; FOREVER 1
; COMMAND TO ESTABLISH PRINTING THRU THE TAPE IN
; TERFACE
215E      3E06           TPRINT: MVI   A,6 ; SET PRINT FLAG
2160      01              .BYTE  01H ; LXI   B,#### OPCODE TO
; EFFECT SKIP
; COMMAND TO ESTABLISH OUTPUT TO TAPE
2161      3E02           TOUTPU: MVI   A,2 ; SET OUTPUT FLAG
2163      32 4EC2        STA    TAPEST
2166      F7              RST    RSTFIN
; COMMAND TO LOAD 128 BYTE BOOTSTRAP FROM TAPE
2167      CD 2133        TLOAD:  CALL  TINPES ; START TAPE READING
216A      21 4000        LXI    H,NORMEM
216D      E5              PUSH   H
216E      E5              TLOAD1: PUSH  H
216F      CD 2D9C        CALL   CHKIO
2172      E1              POP    H
2173      77              MOV    M,A
2174      23              INX    H
2175      CB7D           BIT    7,L
2177      28F5           JRZ    TLOAD1
2179      C9              RET ; ENTER VIA RETURNING
;
; BCD MATH ROUTINE INTERFACE
; IMPLEMENTS $+, $-, $*, */
217A      E7              BCDMAT: RST   RSTIGN
217B      13              INX    D
217C      F5              PUSH   PSW ; SAVE CODE SCANNED

```

out

.MAIN. -

```

217D    CD 21E1          CALL    TSTVFF  ; GET FIRST   VAR
2180    D5              PUSH    D
2181    11 4EB0         LXI    D,BCDA1 ; CONVERT TO BCD
2184    CD 21FE         CALL    ASCBCD
2187    D1              POP     D
2188    CF              TSTCC  COMMA,PIXDUD  ; INSIST ON COMM
                        ASI
                        RST    1
2189    2C              +     .BYTE  COMMA
218A    5A              +     .BYTE  PIXDUD  --1
                        +]

218B    CD 21E1          CALL    TSTVFF  ; GET 2ND VAR
218E    F1              POP     PSW
218F    D5              PUSH    D
2190    11 4EB9         LXI    D,BCDA2 ; SAME STORY
2193    F5              PUSH    PSW
2194    CD 21FE         CALL    ASCBCD
2197    11 4EB0         LXI    D,BCDA1
219A    F1              POP     PSW
219B    0609           MVI    B,9
219D    CD 21E8         CALL    DOBCDD
21A0    D1              POP     D                ; SCAN FOR PLACE
                        TO STORE
21A1    CF              TSTCC  COMMA,PIXDUDI  RST    1
21A2    2C              +     .BYTE  COMMA
21A3    41              +     .BYTE  PIXDUD--1
                        +]

21A4    CD 21E1          CALL    TSTVFF
21A7    D5              PUSH    D
21A8    11 4EB0         LXI    D,BCDA1
                        ; ARG1 CONTAINS THE RESULT - STICK IT LAST VAR
21AB    01 1200        LXI    B,1200H
21AE    EB              BCDASC: XCHG
21AF    FF              INDEXN(INTP%(.IFE .INTP.,LRST 7))
21B0    56              +.BYTE  B6+0]
21B1    C630           ADI    '0'
21B3    EB              XCHG
21B4    CD 2FE8         CALL    STHL
21B7    23              INX    H
21B8    23              INX    H
21B9    0C              INR    C
21BA    10F2           DJNZ  BCDASC
21BC    D1              POP     D
21BD    F7              RST    RSTFIN
                        ; FUNCTION TO RETURN STATE OF ADDRESSED PIXEL
                        ; IE... PIX(X,Y)= 1 IF PIXEL IS 1, 0 IF 0
21BE    CF              PIXFUN: TSTC  ((,PIXDUDI  RST    1
21BF    28              +     .BYTE  ((
21C0    24              +     .BYTE  PIXDUD--1
                        +]

21C1    C5              PUSH    B
21C2    D7              RST    RSTEXP
21C3    E5              PUSH    H
21C4    CF              TSTCC  COMMA,PIXDUDI  RST    1
21C5    2C              +     .BYTE  COMMA
21C6    1E              +     .BYTE  PIXDUD--1

```

.MAIN. -

```

+J
21C7 D7 RST RSTEXP
21C8 CF TSTC ^)^,PIXDUDI RST 1
21C9 29 + .BYTE ^)^
21CA 1A + .BYTE PIXDUD-.-1
+J

21CB C1 POP B
21CC D5 PUSH D ; SAVE PTR
21CD 55 MOV D,L
21CE 59 MOV E,C
21CF CD 23BE CALL R2A
21D2 EB XCHG
21D3 FF INDEXB 1[INTP%L.IFE .INTP.,[RST 7]]
21D4 5D +.BYTE 92+1]
21D5 2003 .WORD PIXTBL
21D7 1A LDAX D ; GET BYTE FROM SCREEN
21D8 A6 ANA M ; MASK OFF NONSENSE
21D9 2600 MVI H,0
21DB 6C MOV L,H
21DC D1 POP D
21DD C1 POP B
21DE C8 RZ
21DF 23 INX H
21E0 C9 RET
; SUBROUTINE TO GET VARIABLE MAKING SURE IT IS 0
; NE
21E1 CD 29AB TSTVFF: CALL TSTV
21E4 D0 RNC ; GO BACK IF GOOD
; ELSE FALL INTO ...
21E5 C3 2945 PIXDUD: JMP QWHAT
; SUBROUTINE TO IT 1
21E8 FE62 DOBCDD: CPI 62H ; < MULT?
21EA 300A JRNC BCD3 ; NO
21EC FE2D CPI ^-^ ; YES IS IT MINUS?
21EE 2803 JRZ BCD2 ; JUMP IF SO
21F0 FF BCDADD ; NO - ITS ADD THEN[INTP%L.IFE .
; INTP.,[RST 7]]
21F1 62 +.BYTE 98+0]
21F2 C9 RET
21F3 FF BCD2: BCDSUB ; SUBTR 1[INTP%L.IFE .INTP.,[RST
; 7]]
21F4 64 +.BYTE 100+0]
21F5 C9 RET
21F6 2003 BCD3: JRNZ BCD4 ; JUMP IF NOT = TIMES
21F8 FF BCDMUL ; ELSE MULT[INTP%L.IFE .INTP.,[R
; ST 7]]
21F9 66 +.BYTE 102+0]
21FA C9 RET
21FB FF BCD4: BCDDIV[INTP%L.IFE .INTP.,[RST 7]]
21FC 68 +.BYTE 104+0]
21FD C9 RET
; SUBROUTINE TO CONVERT ASCII STRING TO BCD
; HL = ASCII IN, DE=BCD OUT
; OUT: DE=ASCII IN BUMPED, HL=BCD OUT NOT BUMPE
D

```

.MAIN. -

```

21FE    EB          ASCBCD: XCHG
21FF    01 1200    LXI      B,1200H
2202    CD 2FD0    ASCBC1: CALL    LDE
2205    FF          STOREN(INTP%,IFE,INTP, [RST 7])
2206    58          +.BYTE  88+0]
2207    0C          INR      C
2208    13          INX      D
2209    13          INX      D
220A    10F6       DJNZ     ASCBC1
220C    C9          RET

; BOX DRAW ROUTINE
220D    D7          BOXDRW: RST     RSTEXP ; GET X
220E    E5          PUSH     H
220F    CF          TSTCC   COMMA,BOXDUD ; FIND COMMA
                RST     1
2210    2C          +      .BYTE  COMMA
2211    55          +      .BYTE  BOXDUD -,-1
                +]
2212    D7          RST     RSTEXP ; GET Y
2213    E5          PUSH     H
2214    CF          TSTCC   COMMA,BOXDUDE RST     1
2215    2C          +      .BYTE  COMMA
2216    50          +      .BYTE  BOXDUD-,-1
                +]
2217    CD 22B6    CALL    EXPRCP ; XS
221A    F5          PUSH     PSW
221B    CF          TSTCC   COMMA,BOXDUDE RST     1
221C    2C          +      .BYTE  COMMA
221D    49          +      .BYTE  BOXDUD-,-1
                +]
221E    CD 22B6    CALL    EXPRCP ; YS
2221    F5          PUSH     PSW
2222    CF          TSTCC   COMMA,BOXDUDE RST     1
2223    2C          +      .BYTE  COMMA
2224    42          +      .BYTE  BOXDUD-,-1
                +]
2225    D7          RST     RSTEXP
2226    D5          PUSH     D
2227    DDE1       POP      X
2229    F1          POP      PSW ; RESTORE YS
222A    47          MOV      B,A
222B    F1          POP      PSW ; AND XS
222C    4F          MOV      C,A
222D    7D          MOV      A,L ; PRESERVE FLAG
222E    E1          POP      H
222F    55          MOV      D,L
2230    E1          POP      H
2231    5D          MOV      E,L
2232    6F          MOV      L,A
                ; NOW WE HAVE: B=YS, C=XS, D=Y, E=X, L=FLAG
                ; LIMIT CHECK Y
2233    60          MOV      H,B
2234    25          DCR      H
2235    CB3C       SRLR     H
2237    7A          MOV      A,D

```

.MAIN. -

```

2238      CD 226A          CALL    SABS
223B      84              ADD     H
223C      FE2C           CPI     44
223E      3023          JRNC   BOXNDR
2240      7A            MOV     A,D
2241      84            ADD     H
2242      57            MOV     D,A
                ; AND X
2243      61            MOV     H,C
2244      CB3C          SRLR   H
2246      7B            MOV     A,E
2247      CD 226A          CALL    SABS
224A      84            ADD     H
224B      FE51          CPI     81
224D      3014          JRNC   BOXNDR
224F      7B            MOV     A,E
2250      94            SUB     H
2251      5F            MOV     E,A
                ; DIDDLE WITH FLAG BYTE
2252      7D            MOV     A,L
2253      E603          ANI     3      ; MODULO 4
2255      280C          JRZ    BOXNDR ; SKIP DRAW IF ZERO
2257      D602          SUI     2      ; ELSE SUBTRACT 2 FOR MA
                SK
2259      F5            BOXDR1: PUSH  PSW
225A      CD 23BE          CALL  R2A
                ; HL = ABS ADDR, A = SA, B=YS, C=XS
225D      D30C          OUT    MAGIC
225F      F1            POP    PSW
2260      CD 226F          CALL  BOXPUT
2263      DDE5          BOXNDR: PUSH  X
2265      D1            POP    D
2266      F7            RST    RSTFIN
2267      C3 2945        BOXDUD: JMP   GWHAT
226A      A7            SABS:  ANA   A
226B      F0            RP
226C      ED44          NEG
226E      C9            RET
                ; SUBROUTINE TO DRAW A BOX ON SCREEN
226F      5F            BOXPUT: MOV   E,A
2270      79            MOV   A,C      ; D = X / 4
2271      0F            RRC
2272      0F            RRC
2273      E63F          ANI   3FH
2275      3C            INR   A
2276      57            MOV   D,A
                ; PAINT FULL BYTE STRIPES
2277      15            MPT1:  DCR   D
2278      2807          JRZ    MPT2
227A      3EAA          MVI   A,10101010B
227C      CD 2294          CALL  STRIPE
227F      18F6          JMPR  MPT1
2281      79            MPT2:  MOV   A,C
2282      E603          ANI   3
2284      3C            INR   A

```

.MAIN. -

```

2285 4F          MOV      C,A
2286 AF          XRA      A
2287 0D          MPT3:   DCR      C
2288 2806         JRZ      MPT4
228A 0F          RRC
228B 0F          RRC
228C F680         ORI      10000000B
228E 18F7         JMPR     MPT3
2290 CD 2294     MPT4:   CALL     STRIPE
2293 AF          XRA      A
          ; FALL INTO ...
          ; SUBROUTINE TO PAINT A STRIPE
2294 E5          STRIPE:  PUSH     H
2295 C5          PUSH     B
2296 32 0FFF      STA      URINAL
2297 3A 4FFF      LDA      URINAL+4000H
229C 4F          MOV      C,A
229D 7B          STRP1:  MOV      A,E
229E FE01        CPI      1
22A0 2002        JRNZ    STRP2
22A2 7E          MOV      A,M
22A3 A9          XRA      C
22A4 AE          STRP2:  XRA      M
22A5 A1          ANA      C
22A6 AE          XRA      M
22A7 77          MOV      M,A
22A8 7D          MOV      A,L
22A9 C628        ADI      BYTEPL
22AB 6F          MOV      L,A
22AC 7C          MOV      A,H
22AD CE00        ACI      0
22AF 67          MOV      H,A
22B0 10EB        DJNZ    STRP1
22B2 C1          POP      B
22B3 E1          POP      H
22B4 23          INX      H
22B5 C9          RET
          ; ROUTINE TO GET EXPRESSION, MAKING SURE IT IS POSITIVE
          ; AND NONZERO
22B6 D7          EXPRCP:  RST      RSTEXP
22B7 7C          MOV      A,H
22B8 B7          ORA      A
22B9 2032        JRNZ    LINED4
22BB B5          ORA      L
22BC 282F        JRZ      LINED4
22BE C9          RET
          ; LINE DRAWER
22BF D7          LINEDR:  RST      RSTEXP
22C0 7D          MOV      A,L
22C1 F5          PUSH     PSW
22C2 CF          TSTCC   COMMA,LINED4[ RST      1
22C3 2C          +      .BYTE  COMMA
22C4 28          +      .BYTE  LINED4--1
          +]

```

.MAIN. -

```

22C5 D7 RST RSTEXP
22C6 7D MOV A,L
22C7 F5 PUSH PSW
22C8 CF LINED1: TSTCC COMMA,LINED4[ RST 1
22C9 2C + .BYTE COMMA
22CA 22 + .BYTE LINED4--.-1
+J

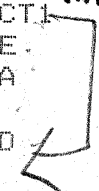
22CB D7 RST RSTEXP
22CC 44 MOV B,H
22CD 4D MOV C,L
22CE D5 PUSH D
22CF DDE1 POP X
22D1 ED5B 4EAC LDED OLDXY
22D5 F1 POP PSW
22D6 67 MOV H,A
22D7 F1 POP PSW
22D8 6F MOV L,A
22D9 22 4EAC SHLD OLDXY ; SET NEW LAST PLACE
; DIDDLE WITH FLAG BYTE
22DC 79 MOV A,C
22DD E603 ANI 3
22DF 2808 JRZ LINED3
22E1 D602 SUI 2
22E3 32 4E5B LINED2: STA PIXVAL ; SET PIXVAL
22E6 CD 22F0 CALL DVECT
22E9 DDE5 LINED3: PUSH X
22EB D1 POP D
22EC F7 RST RSTFIN
22ED C3 2A4C LINED4: JMP QHOW
; LARRY LIVERMORE'S VECTOR DRAWING ALGORITHM
; H=Y1, L=X1, D=Y2, E=X2
DVECT: PUSH D
22F0 D5 MOV B,L
22F1 45 MOV C,E
22F2 4B CALL CDELTA
22F3 CD 238C MOV E,B
22F6 58 MOV L,C
22F7 69 MOV B,H
22F8 44 MOV C,D
22F9 4A CALL CDELTA
22FA CD 238C MOV H,C
22FD 61 MOV D,B
22FE 50
; WE NOW HAVE: H=SGN(DY), L=SGN(DX)
; D=ABS(DY), E=ABS(DX)
22FF 22 4E5E SHLD INCRO
; DECIDE WHICH DELTA IS LARGER
; CALL BIGGER MX, SMALLER MN
2302 0E00 MVI C,0
2304 7A MOV A,B
2305 BB CMP E
2306 3803 JRC VECT1
2308 53 MOV D,E
2309 5F MOV E,A
230A 0C INR C
230B 7A VECT1: MOV A,D ; MX TO A

```

SWITCHED OR NOT

*C,0
MIN
MAX*

*D=MIN
E=MAX*



.MAIN. -

```

230C    CB3F          SRLR    A
230E    47           MOV     B,A
230F    EB          XCHG
2310    22 4E5C      SHLD   MNMX
2313    D1          POP     D
2314    7D          MOV     A,L
2315    3C          INR     A           ; MAKE SURE LAST PIXEL W
    
```

```

RITTEN
; THE INFAMOUS PIXEL PAINTING LOOP
    
```

```

2316    F5          VECT2:  PUSH   PSW
2317    CD 23AE      CALL   R2ACLP
231A    3819        JRC    VECT2A
231C    C5          PUSH   B
231D    E5          PUSH   H
231E    4F          MOV     C,A
231F    0600        MVI    B,0
2321    21 2003     LXI    H,PIXTBL
2324    09          DAD     B
2325    46          MOV     B,M
2326    E1          POP     H
2327    3A 4E5B     LDA    PIXVAL
232A    FE01        CPI    1
232C    2002        JRNZ   VECT9
232E    7E          MOV     A,M
232F    A8          XRA    B
2330    AE          VECT9:  XRA    M
2331    A0          ANA    B
2332    AE          XRA    M
2333    77          MOV     M,A
2334    C1          POP     B
    
```

```

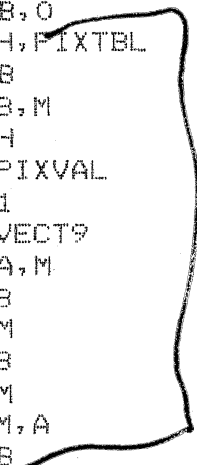
; INCREMENT COORDINATES
    
```

```

2335    2A 4E5C      VECT2A: LHLD  MNMX
2338    78          MOV     A,B
2339    84          ADD    H
233A    BD          CMP    L
233B    380D        JRC    VECT4
233D    95          SUB    L
233E    47          MOV     B,A
233F    2A 4E5E     LHLD  INCRO
2342    7A          MOV     A,D
2343    84          ADD    H
2344    57          MOV     D,A
2345    7B          VECT3:  MOV     A,E
2346    85          ADD    L
2347    5F          MOV     E,A
2348    180B        JMPR  VECT5
234A    47          VECT4:  MOV     B,A
234B    2A 4E5E     LHLD  INCRO
234E    79          MOV     A,C
234F    0F          RRC
2350    30F3        JRNC  VECT3
2352    7A          MOV     A,D
2353    84          ADD    H
2354    57          MOV     D,A
    
```

```

; END OF LOOP
    
```



WRITE ROUTINE
to SNARF

.MAIN. -

```

2355      F1          VECT5:  POP      PSW
2356      3D          DCR      A
2357      20BD       JRNZ     VECT2
2359      C9          RET

; SUBROUTINE TO LOAD HL WITH VDM COORDINATES
; FROM DEVICE VARIABLES
235A      F5          LDVDMC:  PUSH     PSW
235B      3A 4EAA    LDA      VDMY
235E      2F          CMA
235F      C629      ADI      41
2361      FE51      CPI      81      ; OUT OF RANGE?
2363      3801      JRC      LDVDM1 ; NO
2365      AF          XRA      A
2366      67          LDVDM1:  MOV     H,A
2367      3A 4EAB    LDA      VDMX   ; DIDDLE WITH X
236A      C64D      ADI      77
236C      FE9D      CPI      157
236E      3801      JRC      LDVDM2
2370      AF          XRA      A
2371      6F          LDVDM2:  MOV     L,A
2372      F1          POP      PSW
2373      C9          RET

; SUBROUTINE TO STORE HL INTO VDM COORDINATE DEV
ICE VARIABLES
2374      E5          STVDMC:  PUSH     H
2375      7C          MOV     A,H
2376      D629      SUI      41
2378      2F          CMA
2379      6F          MOV     L,A
237A      CD 28AF    CALL     SGNEXT
237D      22 4EAA    SHLD    VDMY
2380      E1          POP      H
2381      7D          MOV     A,L
2382      D64D      SUI      77
2384      6F          MOV     L,A
2385      CD 28AF    CALL     SGNEXT
2388      22 4EAB    SHLD    VDMX
238B      C9          RET

; SUBROUTINE TO COMPUTE DELTA AND INCREMENT FOR
TWO COORDINATES
238C      E5          CDELTA:  PUSH     H
238D      D5          PUSH     D
238E      69          MOV     L,C
238F      CD 28AF    CALL     SGNEXT
2392      EB          XCHG
2393      68          MOV     L,B
2394      CD 28AF    CALL     SGNEXT
2397      AF          XRA      A
2398      ED52       DSBC     D

; COMPUTE SGN(DELTA) AND ABS(DELTA)
239A      B4          ORA      H
239B      2807      JRZ      CDELT1
239D      4F          MOV     C,A
239E      7D          MOV     A,L
239F      ED44       NEG

```


.MAIN. -

```

23A1 47          MOV      B,A
23A2 1807       JMPR    CDELT3
23A4 B5         CDELT1: ORA    L      ; POS CASE 0?
23A5 2802       JRZ    CDELT2
23A7 3E01       MVI    A,1
23A9 45         CDELT2: MOV    B,L
23AA 4F         MOV    C,A
23AB D1         CDELT3: POP    D
23AC E1         POP    H
23AD C9         RET

; RELATIVE TO ABSOLUTE CONVERSION WITH CLIPPING
23AE 7B         R2ACLP: MOV   A,E
23AF FE50       CPI    80
23B1 3803       JRC    ..OKY
23B3 FE60       CPI    0B0H ; ** -80 **
23B5 D8         RC
23B6 7A         ..OKY: MOV   A,D
23B7 FE2C       CPI    44
23B9 3803       JRC    R2A
23BB FE44       CPI    0D4H
23BD D8         RC

; ...
; RELATIVE TO ABSOLUTE CONVERSION
23BE D5         R2A:  PUSH   D
23BF 7A         MOV    A,D
23C0 2F         CMA
23C1 C62C       ADI    44
23C3 57         MOV    D,A
23C4 7B         MOV    A,E
23C5 C650       ADI    80
23C7 5F         MOV    E,A
23C8 AF         XRA    A
23C9 FF         RELAB1[INTP%.IFE .INTP.,[RST 7]]
23CA 3A         +.BYTE 58+0]
23CB EB         XCHG
23CC D1         POP    D
23CD C9         RET

; KB - FUNCTION TO RETURN NEXT CHARACTER FROM KE
YBOARD
23CE C5         GETKB:  PUSH   B
23CF D5         PUSH   D
23D0 CD 2D9C    CALL   CHKIO
23D3 D1         POP    D
23D4 C1         POP    B
23D5 6F         MOV    L,A
23D6 2600       MVI    H,0
23D8 C9         RET

; DEVICE VARIABLE TO OUTPUT TO REFERENCED IO POR
T
23D9 EF         PUTIO:  RST    RSTPAR ; GET PORT #
23DA CF         TSTC   '=' ,PUTCD2 ; GET EQUALSI

RST 1
23DB 3D         + .BYTE '='
23DC 26         + .BYTE PUTCD2 -,-1
+1
    
```

Handwritten notes:
~~MOV A,E~~
~~SUB~~
~~CPI~~

Handwritten notes:
 + 80
 > 160

.MAIN. -

```

23DD    E5          PUSH    H          ; SAVE PORT #
23DE    D7          RST     RSTEXP ; EVALUATE EXPRESSION FOLLOWING
23DF    7D          MOV     A,L          ; A=VALUE TO OUTPUT
23E0    E1          POP     H          ; RESTORE PORT #
23E1    C5          PUSH    B
23E2    44          MOV     B,H
23E3    4D          MOV     C,L
23E4    ED79       OUTP   A          ; IT      1
23E6    C1          POP     B
23E7    F7          RST     RSTFIN ; GO HOME
; FUNCTION TO RETURN VALUE OF A GIVEN IO PORT
23E8    EF          IOFUN: RST     RSTPAR ; GET PORT NUMBA
23E9    C5          PUSH    B
23EA    44          MOV     B,H
23EB    4D          MOV     C,L
23EC    ED78       INP     A
23EE    6F          MOV     L,A
23EF    2600       MVI    H,0
23F1    C1          POP     B
23F2    C9          RET
; DEVICE VARIABLE TO PLAY NOTE WITHOUT PRINTING
23F3    CF          PUTMU: TSTC  '=' ,PUTCD2I  RST     1
23F4    3D          +      .BYTE  '='
23F5    0D          +      .BYTE  PUTCD2-,-1
+J
23F6    D7          RST     RSTEXP
23F7    7D          MOV     A,L
23F8    CD 2E86    CALL   PNOTE
23FB    F7          RST     RSTFIN
; DEVICE VARIABLE TO OUTPUT CHARACTER ON VDM
23FC    CF          PUTCD: TSTC  '=' ,PUTCD2I  RST     1
23FD    3D          +      .BYTE  '='
23FE    04          +      .BYTE  PUTCD2-,-1
+J
23FF    D7          RST     RSTEXP
2400    7D          MOV     A,L
2401    DF          RST     RSTOCH
2402    F7          RST     RSTFIN
2403    C3 2945    PUTCD2: JMP   QWHAT
; ROUTINE TO TRANSFER CONTROL TO ASSEMBLY LANGUAGE SUBROUTINE
2406    21 240F    DOCALL: LXI   H,BBRET ; PUSH RETURN ADDR ON STACK
2409    E5          PUSH    H
240A    D7          RST     RSTEXP ; GET ADDRESS
240B    E9          PCHL   ; AND JUMP TO IT
; COMMAND TO RESTORE TINY BASIC INTERRUPT ROUTINE
240C    CD 2410    TRETUR: CALL  TRIEST
240F    F7          BBRET: RST     RSTFIN ; GO HOME
2410    DB12       TBIEST: IN    TAPEIO ; INITIALIZE TAPE INTERFACE
2412    E602       ANI    2
2414    20FA       JRNZ   TBIEST ; WAIT FOR THANG TO GO T

```

act

.MAIN. -

?

desubroutine

```

0 ZERO
2416 32 4EC2 STA TAPEST
2419 3E03 MVI A,5
241B 32 4EA6 STA DEVTEM
; ACTIVATE TINY BASIC INTERRUPT ROUTINE
241E F3 DI
241F ED5E IM2
2421 3E20 MVI A,ITAB>8
2423 ED47 STAI
2425 3E08 MVI A,8
2427 D30E OUT INMOD
2429 3EC8 MVI A,200
242B D30F OUT INLIN
242D 3E62 MVI A,ITAB&OFFH
242F D30D OUT INFBK
2431 FB EI
2432 C9 RET
; ** TINY BASIC EXECUTION STARTS HERE **
; CLEAR WHOLE KIT AND KABOODLE
2433 AF BEGIN: XRA A
2434 D30C OUT MAGIC
2436 67 MOV H,A
2437 6F MOV L,A
2438 47 MOV B,A
2439 70 BEGIN1: MOV M,B
243A 23 INX H
243B 7C MOV A,H
243C FE50 CPI 50H
243E 20F9 JRNZ BEGIN1
2440 31 4FCE LXI SP,SYSRAM
2443 FF ENTER[CRST 7
2444 00 +.BYTE 0
0001 +.INTP.=1]
SETOUT 1[INTP%.IFE .INTP.,[RST 7]]
2445 17 +.BYTE 22+1]
2446 B0 .BYTE 176
2447 2C .BYTE 44
2448 08 .BYTE 8
EMUSIC 1 ; SHUT DOWN MUSIC[INTP%I
.IFE .INTP.,[RST 7]]
2449 15 +.BYTE 20+1 ]
SETB 1[INTP%.IFE .INTP.,[RST 7]]
244A 7B +.BYTE 122+1]
244B 47 .BYTE 0A2 ; SET INITIAL MASTER OSC
ILATOR
244C 4E58 .WORD MUZMO
; INITIALIZE DEVICE VARIABLES
MOVE 1[INTP%.IFE .INTP.,[RST 7]]
244E 5F +.BYTE 94+1]
244F 4EA2 .WORD DEVVAR
2451 000A .WORD 10
2453 201D .WORD INIDEV
MOVE 1[INTP%.IFE .INTP.,[RST 7]]
2455 5F +.BYTE 94+1]
2456 4E66 .WORD ALTFON
    
```

.MAIN. -

```

2458      0007          .WORD      7
245A      0206          .WORD      FNTSYS
                SETW      1[INTP%[.IFE .INTP.,[RST 7]]
245C      7D           +.BYTE     124+1]
245D      06A0          .WORD      6A0H
245F      4E66          .WORD      ALTFON
                SETW      1[INTP%[.IFE .INTP.,[RST 7]]
2461      7D           +.BYTE     124+1]
2462      A004          .WORD      TEXT+4
2464      4E52          .WORD      TXTUNF
2466      02           EXIT[.BYTE  2
0000          +.INTP.=0]
2467      21 A001      LXI        H,TEXT+1
246A      3EFF          MVI        A,OFFH
246C      CD 2FE8      CALL       STHL
246F      CD 2410      INIT0:   CALL       TBIEST
2472      CD 2C7A      INIT:    CALL       CRLF
                ; DIRECT COMMAND - TEXT COLLECTOR
2475      TELL:
                ;          LXI        D,MSG
                ;          CALL       PRTSTG
2475      STOP:
2475      31 4FEF      RSTART:  LXI        SP,STACK
2478      21 247F      LXI        H,XXST1+1
247B      22 4EC3      SHLD       CURRNT
247E      21 0000      XXST1:   LXI        H,0
2481      22 4EC9      SHLD       LOPVAR
2484      22 4EC5      SHLD       STKGOS
2487      3E3E          XXST2:   MVI        A,'>'
2489      CD 2BD5      CALL       GETLN
248C      D5           PUSH       D
248D      11 4ED4      LXI        D,BUFFER
                ; IGNORE ANY LEADING '>'
2490      1A           LDAX       D
2491      FE3E          CPI        '>'
2493      2001          JRNZ       XXST4
2495      13           INX        D
2496      CD 2A21      XXST4:   CALL       TSTNUM
2499      E7           RST        RSTIGN
249A      7C           MOV        A,H
249B      B5           ORA        L
249C      C1           POP        B
249D      284C          JRZ        EXECO
249F      22 4E64      SHLD       OLDLN
24A2      1B           DCX        D
24A3      7C           MOV        A,H
24A4      12           STAX       D
24A5      1B           DCX        D
24A6      7D           MOV        A,L
24A7      12           STAX       D
24A8      C5           PUSH       B
24A9      D5           PUSH       D
24AA      79           MOV        A,C
24AB      93           SUB        E
24AC      F5           PUSH       PSW

```

OUT

@ preservation

```

24AD    CD 2982          CALL    FNDLN
24B0    D5              PUSH    D
24B1    2010           JRNZ   XXST3
24B3    D5              PUSH    D
24B4    CD 299F          CALL    FNDNXT
24B7    C1              POP     B
24B8    2A 4E52         LHLD   TXTUNF
24BB    CD 2A53          CALL    MVUP
24BE    60              MOV    H,B
24BF    69              MOV    L,C
24C0    22 4E52         SHLD  TXTUNF
24C3    C1              XXST3: POP    B
24C4    2A 4E52         LHLD  TXTUNF
24C7    F1              POP    PSW-
24C8    E5              PUSH   H
24C9    FE03            CPI    3
24CB    28A8            JRZ   RSTART
24CD    85              ADD    L
24CE    5F              MOV    E,A
24CF    3E00            MVI   A,0
24D1    8C              ADC    H
24D2    57              MOV    D,A
24D3    21 A70C         LXI   H,DFTLMT
24D6    EB              XCHG
24D7    CD 202F          CALL   COMP
24DA    D2 297C         JNC   QSORRY
24DD    22 4E52         SHLD  TXTUNF
24E0    D1              POP    D
24E1    CD 2A65          CALL   MVDOWN
24E4    D1              POP    D
24E5    E1              POP    H
24E6    CD 2A53          CALL   MVUP
24E9    189C            JMPR  XXST2

                ; DIRECT AND EXEC
24EB    E7              EXEC0: RST   RSTIGN ; GET FIRST   NONBLANK

24EC    D5              PUSH    D
24ED    FE68            CPI    68H ; IS SHE A TOKEN?
24EF    3813            JRC   EXEC0A ; NO
24F1    FE75            CPI    75H
24F3    300F            JRNC  EXEC0A ; NO

                ; WE FOUND A TOKEN - LOOKUP IN TABLE AND JUMP TO
                IT
24F5    07              RLC
24F6    5F              MOV    E,A
24F7    1600            MVI   D,0
24F9    21 1F76         LXI   H,TOKJT-208
24FC    19              DAD   D
24FD    5E              MOV    E,M
24FE    23              INX   H
24FF    56              MOV    D,M
2500    EB              XCHG
2501    D1              POP    D
2502    13              INX   D
2503    E9              PCHL
    
```

.MAIN. -

```

; NOT A TOKEN - A VARIABLE PERHAPS?
2504 CD 29AB EXECOA: CALL TSTV
2507 3808 JRC EXECOB ; NO - SEARCH 1
2509 CF TSTC '<=' ,EXECOB ; MAYBE1
RST 1
250A 3D + .BYTE '<='
250B 05 + .BYTE EXECOB -,-1
+1
250C C1 POP B ; THROW OUT OLD PTR
250D CD 2918 CALL SETV1 ; ASSIGNMENT 1
2510 F7 RST RSTFIN
2511 D1 EXECOB: POP D
2512 21 2B37 LXI H,TAB2-1
2515 E7 EXEC: RST RSTIGN
2516 D5 PUSH D
2517 CD 2FD0 EX1: CALL LDE
251A 13 INX D
251B 23 INX H
251C BE CMP M
251D 28F8 JRZ EX1
251F 3E7F MVI A,07FH
2521 1B DCX D
2522 BE CMP M
2523 3808 JRC EX5
2525 23 EX2: INX H
2526 BE CMP M
2527 30FC JRNC EX2
2529 23 INX H
252A D1 POP D
252B 18E8 JMPR EXEC
252D 7E EX5: MOV A,M
252E 23 INX H
252F 6E MOV L,M
2530 E67F ANI 07FH
2532 67 MOV H,A
2533 F1 POP PSW
2534 E9 PCHL
; COMMAND TO CLEAR SCREEN
2535 CLRSCR:
2535 21 4000 LXI H,4000H
2538 7E CLRLP: MOV A,M
2539 E655 ANI 01010101B
253B 77 MOV M,A
253C 23 INX H
253D 7C MOV A,H
253E FE4E CPI 4EH
2540 20F6 JRNZ CLRLP
; RESET VDM GOODIES
2542 D5 PUSH D
2543 FF MOVE 1[INTP%L,IFE .INTP.,[RST 7]]
2544 5F +.BYTE 94+1]
2545 4EA8 .WORD VDMX
2547 0006 .WORD 6
2549 2023 .WORD INIDEV+6
254B D1 POP D

```

.MAIN. -

```

254C   F7           RST      RSTFIN
254D   11 A000     RUN:    LXI      D,TEXT
2550   21 0000     RUNNXL: LXI      H,0
2553   CD 298A     CALL    FNDLP
2556   DA 2475     JC      RSTART
2559   ED53 4EC3   RUNTSL: SDED    CURRNT
255D   CD 2E3F     CALL    WHATSU ; CHECK FOR HALT
2560   2008        JRNZ    ..OK   ; AND FOR TRACE KEY
2562   D5          PUSH   D      ; TRACE IS WANTED
2563   CD 2B25     CALL    PRTLN
2566   CD 2AB7     CALL    PRTSTG
2569   D1          POP    D
256A   ..OK:
256A   13          INX    D
256B   13          INX    D
256C   CD 2E3F     RUNSML: CALL    WHATSU ; CHECK FOR INTERRUPT KE
Y
256F   C3 24EB     JMP    EXECC
2572   D7          GOTO:   RST    RSTEXP
2573   D5          PUSH   D
2574   CD 2982     CALL    FNDLN
2577   C2 2A4D     JNZ    AHOW
257A   F1          POP    PSW
257B   18DC        JMPR   RUNTSL

; LIST AND PRINT
; NEW - IMPROVED LIST COMMAND
; LETS YOU PUT IT IN A PROGRAM
257D   21 0000     LIST:   LXI      H,0      ; ASSUME AT EOL
2580   E7          RST    RSTIGN ; IGNORE
2581   CD 214B     CALL    ATNL
2584   2805        JRZ    LS3
2586   FE2C        CPI    ','    ; LEADING COMMA?
2588   2801        JRZ    LS3    ; YEP - SKIP FIRST

      EXPR GET
      ; NOT AT END - GET FIRST      EXPR
258A   D7          LS2:   RST    RSTEXP
258B   E5          LS3:   PUSH   H
258C   21 FFFF     LXI    H,0FFFFH
258F   CF          TSTCC  COMMA,LS4E   RST      1
2590   2C          +     .BYTE  COMMA
2591   01          +     .BYTE  LS4-.-1
      +]
2592   D7          RST    RSTEXP
2593   D5          LS4:   PUSH   D
2594   FDE1        POP    Y
2596   E3          XTHL
2597   CD 2982     CALL    FNDLN
259A   3815        LS5:   JRC    LSQUIT
259C   E3          XTHL
259D   7C          MOV    A,H
259E   B5          ORA    L
259F   2810        JRZ    LSQUIT
25A1   2B          DCX    H
25A2   E3          XTHL
25A3   CD 2B25     CALL    PRTLN

```

.MAIN. -

```

25A6      CD 2AB7          CALL    PRTSTG
25A9      CD 2E3F          CALL    WHATSU
25AC      CD 298A          CALL    FNDLP
25AF      18E9            JMPR    L55
25B1      FDE5            LSQUIT: PUSH    Y
25B3      D1              POP     D
25B4      F7              RST     RSTFIN
25B5      0E08            PRINT: MVI    C,8
25B7      CF              TSTCC   59,PR1[RST    1
25B8      3B              +      .BYTE  59
25B9      05              +      .BYTE  PR1-,-1
                +]

25BA      CD 2C7A          CALL    CRLF
25BD      18AD            JMPR    RUNSML
25BF      CF              PR1:   TSTCC   CR,PR6[RST    1
25C0      0D              +      .BYTE  CR
25C1      1B              +      .BYTE  PR6-,-1
                +]

25C2      CD 2C7A          CALL    CRLF
25C5      1889            JMPR    RUNNXL
25C7      CF              PR2:   TSTCC   '#',PR4[RST    1
25C8      23              +      .BYTE  '#'
25C9      0B              +      .BYTE  PR4-,-1
                +]

25CA      D7              PR3:   RST     RSTEXP
25CB      3EC0            MVI    A,0COH
25CD      A5              ANA    L
25CE      B4              ORA    H
25CF      C2 2A4C          JNZ    GHOW
25D2      4D              MOV    C,L
25D3      1805            JMPR    PR5
25D5      CD 2AC5          PR4:   CALL    QTSTG
25D8      1814            JMPR    PR9
25DA      CF              PR5:   TSTCC   COMMA,PR8[RST    1
25DB      2C              +      .BYTE  COMMA
25DC      0D              +      .BYTE  PR8-,-1
                +]

25DD      CF              PR6:   TSTCC   COMMA,PR7[RST    1
25DE      2C              +      .BYTE  COMMA
25DF      05              +      .BYTE  PR7-,-1
                +]

25E0      3E20            MVI    A,' '
25E2      DF              RST     RSTOCH
25E3      18F8            JMPR    PR6
25E5      CD 292E          PR7:   CALL    FIN
25E8      18DD            JMPR    PR2
25EA      CD 2C7A          PR8:   CALL    CRLF
25ED      F7              RST     RSTFIN
25EE      D7              PR9:   RST     RSTEXP
25EF      C5              PUSH   B
25F0      CD 2AE8          CALL    PRTNUM
25F3      C1              POP    B
25F4      18E4            JMPR    PR5
                ; GOSUB AND RETURN
25F6      CD 2A92          GOSUB: CALL   PUSHA

```


.MAIN. -

25F9	D7		RST	RSTEXP
25FA	D5		PUSH	D
25FB	CD 2982		CALL	FNDLN
25FE	C2 2A4D		JNZ	AHOW
2601	2A 4EC3		LHLD	CURRNT
2604	E5		PUSH	H
2605	2A 4EC5		LHLD	STKGOS
2608	E5		PUSH	H
2609	21 0000		LXI	H,0
260C	22 4EC9		SHLD	LOPVAR
260F	39		DAD	SP
2610	22 4EC5		SHLD	STKGOS
2613	C3 2559		JMP	RUNTSL
2616	2A 4EC5	RETURN:	LHLD	STKGOS
2619	7C		MOV	A,H
261A	B5		ORA	L
261B	CA 2945		JZ	QWHAT
261E	F9		SFHL	
261F	E1	RESTO:	POP	H
2620	22 4EC5		SHLD	STKGOS
2623	E1		POP	H
2624	22 4EC3		SHLD	CURRNT
2627	(D1)		POP	D
2628	CD 2A77		CALL	POPA
262B	(F7)		RST	RSTFIN
			; FOR AND NEXT	
262C	CD 2A92	FOR:	CALL	PUSHA
262F	CD 2912		CALL	SETVAL
2632	2B		DCX	H
2633	22 4EC9		SHLD	LOPVAR
2636	CF		TSTCC	77H,FR1A ; TO?L RST
			1	
2637	77	+	.BYTE	77H
2638	01	+	.BYTE	FR1A --1
			+J	
2639	D7	FR1:	RST	RSTEXP
263A	22 4ECD	FR1A:	SHLD	LOPLMT
263D	21 0001		LXI	H,1
2640	CF		TSTCC	75H,FR4 ; STEP?L RST
			1	
2641	75	+	.BYTE	75H
2642	01	+	.BYTE	FR4 --1
			+J	
2643	D7		RST	RSTEXP
2644	22 4ECB	FR4:	SHLD	LOPINC
2647	2A 4EC3		LHLD	CURRNT
264A	22 4ECF		SHLD	LOPLN
264D	EB		XCHG	
264E	22 4ED1		SHLD	LOPPT
2651	01 000A		LXI	B,10
2654	2A 4EC9		LHLD	LOPVAR
2657	EB		XCHG	
2658	60		MOV	H,B
2659	68		MOV	L,B
265A	39		DAD	SP

.MAIN. -

265B	1801		JMPR	FR6
265D	09	FR5:	DAD	B
265E	7E	FR6:	MOV	A,M
265F	23		INX	H
2660	B6		ORA	M
2661	2817		JRZ	FR7
2663	7E		MOV	A,M
2664	2B		DCX	H
2665	BA		CMP	D
2666	20F5		JRNZ	FR5
2668	7E		MOV	A,M
2669	AB		XRA	E
266A	20F1		JRNZ	FR5
266C	EB		XCHG	
266D	67		MOV	H,A
266E	6F		MOV	L,A
266F	39		DAD	SP
2670	44		MOV	B,H
2671	4D		MOV	C,L
2672	21 000A		LXI	H,10
2675	19		DAD	D
2676	CD 2A65		CALL	MVDOWN
2679	F9		SPHL	
267A	2A 4ED1	FR7:	LHLD	LOPPT
267D	EB		XCHG	
267E	F7		RST	RSTFIN
267F	CD 29AB	NEXT:	CALL	TSTV
2682	DA 2945		JC	QWHAT
2685	22 4EC7		SHLD	VARNXT
2688	D5	NX1:	PUSH	D
2689	EB		XCHG	
268A	2A 4EC9		LHLD	LOPVAR
268D	7C		MOV	A,H
268E	B5		ORA	L
268F	CA 2946		JZ	AWHAT
2692	CD 202F		CALL	COMP
2695	2809		JRZ	NX2
2697	D1		POP	D
2698	CD 2A77		CALL	POPA
269B	2A 4EC7		LHLD	VARNXT
269E	18E8		JMPR	NX1
26A0	EB	NX2:	XCHG	
26A1	CD 2FD0		CALL	LDE
26A4	6F		MOV	L,A
26A5	13		INX	D
26A6	CD 2FD0		CALL	LDE
26A9	67		MOV	H,A
26AA	EB		XCHG	
26AB	2A 4ECB		LHLD	LOPINC
26AE	E5		PUSH	H
26AF	7C		MOV	A,H
26B0	AA		XRA	D
26B1	7A		MOV	A,D
26B2	19		DAD	D
26B3	FA 26BA		JM	NX3

.MAIN. -

26B6	AC		XRA	H
26B7	FA 26E1		JM	NX5
26BA	EB	NX3:	XCHG	
26BB	2A 4EC9		LHLD	LOPVAR
26BE	7B		MOV	A,E
26BF	CD 2FE8		CALL	STHL
26C2	23		INX	H
26C3	7A		MOV	A,D
26C4	CD 2FE8		CALL	STHL
26C7	2A 4ECD		LHLD	LOPLMT
26CA	F1		POP	PSW
26CB	B7		ORA	A
26CC	F2 26D0		JP	NX4
26CF	EB		XCHG	
26D0	CD 2029	NX4:	CALL	CKHLDE
26D3	D1		POP	D
26D4	380D		JRC	NX6
26D6	2A 4ECF		LHLD	LOPLN
26D9	22 4EC3		SHLD	CURRNT
26DC	2A 4ED1		LHLD	LOPPT
26DF	EB		XCHG	
26E0	F7		RST	RSTFIN
26E1	E1	NX5:	POP	H
26E2	D1	NXXX:	POP	D
26E3	CD 2A77	NX6:	CALL	NOFA
26E6	F7		RST	RSTFIN
			: REM, IF, INPUT, LET	
26E7	21 0000	REM:	LXI	H,0
26EA	1801		JMPR	IF1
26EC	D7	IFF:	RST	RSTEXP
26ED	7C	IF1:	MOV	A,H
26EE	B5		ORA	L
26EF	C2 256C		JNZ	RUNSML
26F2	CD 29A1	REM:	CALL	FNDSKP
26F5	D2 2559		JNC	RUNTSL
26F8	C3 2475		JMP	RSTART
26FB	2A 4EC7	INPERR:	LHLD	STKINP
26FE	F9		SPHL	
26FF	E1		POP	H
2700	22 4EC3		SHLD	CURRNT
2703	D1		POP	D
2704	D1		POP	D
2705		INPUT	==	.
2705	D5	IP1:	PUSH	D
2706	CD 2AC5		CALL	QTSTG
2709	1823		JMPR	IFS
270B	CD 29AB	IP2:	CALL	TSTV
270E	3817		JRC	IP5
2710	CD 273E	IP3:	CALL	IP12
2713	11 4ED4		LXI	D, BUFFER
2716	D7		RST	RSTEXP
2717	D1		POP	D
2718	EB		XCHG	
2719	7B		MOV	A,E
271A	CD 2FE8		CALL	STHL

*RLC
JRNC
XCHG NX4*

*crunch
REM*

.MAIN. -

271D	23		INX	H		
271E	7A		MOV	A, D		
271F	CD 2FE8		CALL	STHL		
2722	E1	IP4:	POP	H		
2723	22 4EC3		SHLD	CURRNT		
2726	D1		POP	D		
2727	F1	IP5:	POP	PSW		
2728	CF	IP6:	TSTCC	COMMA, IP7:	RST	1
2729	2C	+	.BYTE	COMMA		
272A	02	+	.BYTE	IP7--1		
		+]				
272B	18D8		JMPR	INPUT		
272D	F7	IP7:	RST	RSTFIN		
272E	D5	IP8:	PUSH	D		
272F	CD 29AB		CALL	TSTV		
2732	3003		JRNC	IP11		
2734	C3 2945	IP10:	JMP	QWHAT		
2737	43	IP11:	MOV	B, E		
2738	D1		POP	D		
2739	CD 2ADE		CALL	PRTCHS		
273C	18D2		JMPR	IP3		
273E	C1	IP12:	POP	B		
273F	D5		PUSH	D		
2740	EB		XCHG			
2741	2A 4EC3		LHLD	CURRNT		
2744	E5		PUSH	H		
2745	21 2705		LXI	H, IP1		
2748	22 4EC3		SHLD	CURRNT		
274B	21 0000		LXI	H, 0		
274E	39		DAD	SP		
274F	22 4EC7		SHLD	STKINP		
2752	D5		PUSH	D		
2753	C5		PUSH	B		
2754	3E20		MVI	A, / /		
2756	C3 2BD5		JMP	GETLN		
2759	1A	DEFLT:	LDAX	D		
275A	FE0D		CPI	CR		
275C	28CF		JRZ	IP7		
275E	C3 2945		JMP	QWHAT		
		;	** EXPR **			
2761	CD 27A9	EXPR:	CALL	EXPR1		
2764	E5		PUSH	H		
2765	21 2B97		LXI	H, TAB6-1		
2768	C3 2515		JMP	EXEC		
276B	CD 2794	XPR1:	CALL	XPR8		
276E	D8		RC			
276F	6F		MOV	L, A		
2770	C9		RET			
2771	CD 2794	XPR2:	CALL	XPR8		
2774	C8		RZ			
2775	6F		MOV	L, A		
2776	C9		RET			
2777	CD 2794	XPR3:	CALL	XPR8		
277A	C8		RZ			
277B	D8		RC			

.MAIN. -

277C	6F		MOV	L,A		
277D	C9		RET			
277E	CD 2794	XPR4:	CALL	XPR8		
2781	6F		MOV	L,A		
2782	C8		RZ			
2783	D8		RC			
2784	6C		MOV	L,H		
2785	C9		RET			
2786	CD 2794	XPR5:	CALL	XPR8		
2789	C0		RNZ			
278A	6F		MOV	L,A		
278B	C9		RET			
278C	CD 2794	XPR6:	CALL	XPR8		
278F	D0		RNC			
2790	6F		MOV	L,A		
2791	C9		RET			
2792	E1	XPR7:	POP	H		
2793	C9		RET			
2794	79	XPR8:	MOV	A,C		
2795	E1		POP	H		
2796	C1		POP	B		
2797	E5		PUSH	H		
2798	C5		PUSH	B		
2799	4F		MOV	C,A		
279A	CD 27A9		CALL	EXPR1		
279D	EB		XCHG			
279E	E3		XTHL			
279F	CD 2029		CALL	CKHLDE		
27A2	D1		POP	D		
27A3	21 0000		LXI	H,0		
27A6	3E01		MVI	A,1		
27A8	C9		RET			
27A9	CF	EXPR1:	TSTC	<--,XP11E	RST	1
27AA	2D	+	.BYTE	<--		
27AB	05	+	.BYTE	XP11--,-1		
		+J				
27AC	21 0000		LXI	H,0		
27AF	1821		JMPR	XP16		
27B1	CF	XP11:	TSTC	<+<,XP12E	RST	1
27B2	2B	+	.BYTE	<+<		
27B3	00	+	.BYTE	XP12--,-1		
		+J				
27B4	CD 27DB	XP12:	CALL	EXPR2		
27B7	CF	XP13:	TSTC	<+<,XP15E	RST	1
27B8	2B	+	.BYTE	<+<		
27B9	15	+	.BYTE	XP15--,-1		
		+J				
27BA	E5		PUSH	H		
27BB	CD 27DB		CALL	EXPR2		
27BE	EB	XP14:	XCHG			
27BF	E3		XTHL			
27C0	7C		MOV	A,H		
27C1	AA		XRA	D		
27C2	7A		MOV	A,D		
27C3	19		DAD	D		

.MAIN. -

27C4	D1		POP	D		
27C5	FA 27B7		JM	XP13		
27C8	AC		XRA	H		
27C9	F2 27B7		JP	XP13		
27CC	C3 2A4C		JMP	QHOW		
27CF	CF	XP15:	TSTC	^--^, XPR9[RST	1
27D0	2D	+	.BYTE	^--^		
27D1	A0	+	.BYTE	XPR9--.-1		
		+J				
27D2	E5	XP16:	PUSH	H		
27D3	CD 27DB		CALL	EXPR2		
27D6	CD 28FD		CALL	CHKSGN		
27D9	18E3		JMPR	XP14		
27DB	CD 2840	EXPR2:	CALL	EXPR3		
27DE	CF	XP21:	TSTCC	62H, XP24[RST	1
27DF	62	+	.BYTE	62H		
27E0	29	+	.BYTE	XP24--.-1		
		+J				
27E1	E5		PUSH	H		
27E2	CD 2840		CALL	EXPR3		
27E5	0600		MVI	B, 0		
27E7	CD 28FA		CALL	CHKSGN		
27EA	E3		XTHL			
27EB	CD 28FA		CALL	CHKSGN		
27EE	EB		XCHG			
27EF	E3		XTHL			
27F0	7C		MOV	A, H		
27F1	B7		ORA	A		
27F2	2806		JRZ	XP22		
27F4	7A		MOV	A, D		
27F5	B2		ORA	D		
27F6	EB		XCHG			
27F7	C2 2A4D		JNZ	AHOW		
27FA	7D	XP22:	MOV	A, L		
27FB	21 0000		LXI	H, 0		
27FE	B7		ORA	A		
27FF	2832		JRZ	XP25		
2801	19	XP23:	DAD	D		
2802	DA 2A4D		JC	AHOW		
2805	3D		DCR	A		
2806	20F9		JRNZ	XP23		
2808	1829		JMPR	XP25		
280A	CF	XP24:	TSTCC	63H, XPR9[RST	1
280B	63	+	.BYTE	63H		
280C	65	+	.BYTE	XPR9--.-1		
		+J				
280D	E5		PUSH	H		
280E	CD 2840		CALL	EXPR3		
2811	0600		MVI	B, 0		
2813	CD 28FA		CALL	CHKSGN		
2816	E3		XTHL			
2817	CD 28FA		CALL	CHKSGN		
281A	EB		XCHG			
281B	E3		XTHL			
281C	EB		XCHG			

.MAIN. -

```

281D 7A      MOV      A,D
281E B3      ORA      E
281F CA 2A4D JZ       AHOW
2822 C5      PUSH     B
2823 CD 28E5  CALL    DIVIDE
2826 D1      POP      D      ; SIGN STUFF TO DE
2827 C5      PUSH     B      ; SAVE DIVIDE RESULT
2828 CB7A    BIT      7,D    ; WAS SIGN SET?
282A C4 28FD CNZ     CHGSGN ; YEP - CHANGE
282D 22 4EAE SHLD   REMAIN  ; STUFF IT
2830 E1      POP      H      ; RESULT TO HL
2831 42     MOV      B,D    ; COPY OVER SIGN STUFF
2832 4B     MOV      C,E
2833 D1      POP      D
2834 7C     MOV      A,H
2835 B7     ORA      A
2836 FA 2A4C JM      QHOW
2839 78     MOV      A,B
283A B7     ORA      A
283B FC 28FD CM      CHGSGN
283E 189E   JMPR   XP21
2840 21 2B6E  EXPR3: LXI   H,TAB3-1
2843 C3 2515   JMP    EXEC
2844 CD 29AB  NOTF:  CALL   TSTV
2849 380E   JRC    XP32
284B EB     XCHG
284C CD 2FDD  CALL   LDE
284F F5     PUSH   PSW
2850 13     INX   D
2851 CD 2FDD  CALL   LDE
2854 EB     XCHG
2855 67     MOV    H,A
2856 F1     POP   PSW
2857 6F     MOV   L,A
2858 C9     RET
2859 CD 2A21  XP32:  CALL   TSTNUM
285C 78     MOV   A,B
285D B7     ORA   A
285E C0     RNZ
                ; SINGLE CHAR STRING CONSTANT?
285F CF     TSTC  '<>',PARN ; HAVE WE GOT QU
                OTES?
                RST  1
2860 22     + .BYTE '<>'
2861 09     + .BYTE PARN  --,-1
                +]
2862 CD 2FDD  CALL   LDE
2865 6F     MOV   L,A      ; FAILED TSTNUM SET H TO
                ZERO
2866 13     INX   D
2867 CF     TSTC  '<>',XPRO ; ERROR IF NO TR
                AILING
                RST  1
2868 22     + .BYTE '<>'
2869 09     + .BYTE XPRO  --,-1
                +]
286A C9     RET

```

.MAIN. -

```

286B  CF      PARN:  TSTC  ( ( , XPROI      RST      1
286C  28      +      .BYTE  ( (
286D  05      +      .BYTE  XPRO-. -1
      +]
286E  D7      PARNP:  RST    RSTEXP
286F  CF      TSTC  ( ) , XPROI      RST      1
2870  29      +      .BYTE  ( )
2871  01      +      .BYTE  XPRO-. -1
      +]
2872  C9      XPR9:  RET
2873  C3 2945 XPRO:  JMP    @WHAT
2876  EF      RND:   RST    RSTPAR
2877  7C      MOV    A, H
2878  B7      ORA    A
2879  FA 2A4C JM     @HOW
287C  B5      ORA    L
287D  CA 2A4C JZ     @HOW
2880  D5      PUSH   D
2881  EB      XCHG           ; DE = RANGE
2882  AF      XRA    A
2883  FF      RANGED[INTP%[.IFE .INTP.,[RST 7]]
2884  76      +.BYTE  118+0]
2885  6F      MOV    L, A
2886  AF      XRA    A
2887  FF      RANGED[INTP%[.IFE .INTP.,[RST 7]]
2888  76      +.BYTE  118+0]
2889  67      MOV    H, A
      ; HL = RANDOM #
288A  C5      PUSH   B
288B  CD 28E5 CALL   DIVIDE
288E  C1      POP    B
288F  D1      POP    D
2890  23      INX    H
2891  C9      RET
2892  EF      ABS:   RST    RSTPAR
2893  1B      DCX    D
2894  CD 28FA CALL   CHKSGN
2897  13      INX    D
2898  C9      RET
2899  2A 4E52 SIZE:  LHLD   TXTUNF
289C  D5      PUSH   D
289D  EB      XCHG
289E  21 A70C LXI    H, DFTLMT
28A1  A7      ANA    A
28A2  ED52   DSBC   D
28A4  D1      POP    D
28A5  C9      RET
      ; FUNCTION TO SENSE DIAL VALUE
28A6  3E1B   GETPOT: MVI    A, 1BH
28A8  CD 28D8 CALL   CHKRNG ; GET DATA
28AB  2F      CMA
28AC  D680   SUI    80H
28AE  6F      MOV    L, A
      ; FALL INTO ...
      ; SIGN EXTEND SUBROUTINE

```


.MAIN. -

```

28AF      2600      SGNEXT: MVI      H,0
28B1      7D              MOV      A,L
28B2      A7              ANA      A
28B3      F0              RP
28B4      25              DCR      H
28B5      C9              RET

; FUNCTION TO SENSE STATE OF TRIGGER
28B6      CD 28D6      GETTRG: CALL     CHKRN1
28B7      E610          ANI      10H
28BB      C8              RZ
28BC      2C              INR      L
28BD      C9              RET

; FUNCTIONS TO RETURN JOYSTICK VALUE
; THESE FUNCTIONS RETURN EITHER +1, ,0 OR -1, D
; EPENDING
; ON JOYSTICK STATE
28BE      CD 28D6      GETJX:  CALL     CHKRN1 ; PARM IN RANGE?
28C1      0F              RRC
28C2      0F              RRC
28C3      0F              RRC
28C4      380E          JRC      GETJY3
28C6      0F              RRC
28C7      3807          JRC      GETJY1
28C9      C9              RET

; ENTRY FOR Y JOYSTICK VALUE
28CA      CD 28D6      GETJY:  CALL     CHKRN1
28CD      0F              RRC
28CE      3002          JRNC     GETJY2
28D0      23              GETJY1: INX      H
28D1      C9              RET
28D2      0F              GETJY2: RRC
28D3      D0              RNC
28D4      2B              GETJY3: DCX     H
28D5      C9              RET

; SUBROUTINE TO GET PARAMETER BETWEEN 1 AND 4
28D6      3E0F          CHKRN1: MVI      A,0FH
28D8      C5              CHKRNG: PUSH     B
28D9      F5              PUSH     PSW
28DA      EF              RST     RSTPAR
28DB      F1              POP      PSW
28DC      85              ADD      L
28DD      4F              MOV      C,A
28DE      ED78          INP      A
28E0      C1              POP      B
28E1      21 0000       LXI      H,0
28E4      C9              RET

; DIVIDE, SUBDE, CHKSGN, CHGSGN, CKHLDE
28E5      E5              DIVIDE: PUSH     H
28E6      6C              MOV      L,H
28E7      2600          MVI      H,0
28E9      CD 28F0       CALL     DV1
28EC      41              MOV      B,C
28ED      7D              MOV      A,L
28EE      E1              POP      H
28EF      67              MOV      H,A

```

.MAIN. -

```

28F0 0EFF          DV1:  MVI      C,-1
28F2 0C           DV2:  INR      C
28F3 A7           ANA      A
28F4 ED52        DSBC   D
28F6 30FA        JRNC   DV2
28F8 19           DAD    D
28F9 C9           RET
28FA 7C          CHKSGN: MOV    A,H
28FB B7           ORA    A
28FC F0           RP
28FD 7C          CHGSGN: MOV    A,H
28FE B5           ORA    L
28FF C8           RZ
2900 7C          MOV    A,H
2901 F5          PUSH   PSW
2902 2F          CMA
2903 67          MOV    H,A
2904 7D          MOV    A,L
2905 2F          CMA
2906 6F          MOV    L,A
2907 23          INX    H
2908 F1          POP    PSW
2909 AC          XRA    H
290A F2 2A4C     JP     QHOW
290D 78          MOV    A,B
290E EE80       XRI   80H
2910 47          MOV    B,A
2911 C9          RET
;CKHLDE:
;          XRA    D
;          JP     CK1
;          XCHG
;CK1:
;          CALL   COMP
;          RET
;COMP:
;          MOV    A,H
;          CMP    D
;          RNZ
;          MOV    A,L
;          CMP    E
;          RET
; SETVAL, FIN, AND ERROR
2912 CD 21E1     SETVAL: CALL   TSTVFF
2915 CF          TSTC   '=',QWHA1E   RST    1
2916 3D          +     .BYTE   '= '
2917 2D          +     .BYTE   QWHAT-.-1
+ ]
2918 E5          SETV1: PUSH   H
2919 D7          RST   RSTEXP
291A 44          MOV    B,H
291B 4D          MOV    C,L
291C E1          POP    H
291D F5          PUSH   PSW
291E 79          MOV    A,C
291F CD 2FE8     CALL   STHL
2922 23          INX    H

```

.MAIN. -

```

2923 78          MOV      A,B
2924 CD 2FE8     CALL    STHL
2927 F1         POP     PSW
2928 C9         RET
2929 CD 292E     FINISH: CALL   FIN
292C 1817       JMPR   QWHAT
292E CF        FIN#:  TSTCC  59,FI1I RST    1
292F 3B        +      .BYTE  59
2930 04        +      .BYTE  FI1-.-1
                +J
2931 F1         POP     PSW
2932 C3 256C     JMP     RUNSML
2935 CF        FI1#:  TSTCC  CR,FI2I RST    1
2936 0D        +      .BYTE  CR
2937 66        +      .BYTE  FI2-.-1
                +J
2938 F1         POP     PSW
2939 C3 2550     JMP     RUNNXL
293C CD 2FD0     IGNBLK: CALL   LDE
293F FE20       CPI
2941 C0         RNZ
2942 13         INX     D
2943 18F7       JMPR   IGNBLK
2945 D5        QWHAT:  PUSH   D
2946 11 2035    AWHAT:  LXI    D,WHAT
2949 CD 2C7A    ERROR:  CALL   CRLF
294C CD 2AB7     CALL   PRTSTG
294F 2A 4EC3    LHLD  CURRNT
2952 E5        PUSH   H
2953 EB        XCHG
2954 CD 2FD0     CALL   LDE
2957 67        MOV     H,A
2958 13         INX     D
2959 CD 2FD0     CALL   LDE
295C B4        ORA     H
295D EB        XCHG
295E D1        POP     D
295F CA 2475    JZ     TELL
2962 EB        XCHG
2963 CD 2FD0     CALL   LDE
2966 EB        XCHG
2967 B7        ORA     A
2968 FA 26FB     JM     INPERR
296B CD 2B25     CALL   PRTLN
296E C1        POP     B
296F 41        MOV     B,C
2970 CD 2ADE     CALL   PRTCHS
2973 3E3F       MVI    A,'?'
2975 DF        RST    RSTOCH
2976 CD 2AB7     CALL   PRTSTG
2979 C3 2475     JMP    TELL
297C D5        QSORRY:  PUSH   D
297D 11 2040    ASORRY:  LXI    D,SORRY
2980 18C7       JMPR   ERROR

```

CPI '9'
TSTCH COMMA, QWHAT

: FNDLN

.MAIN. -

2982	7C	FNDLN:	MOV	A,H	
2983	B7		ORA	A	
2984	FA 2A4C		JM	QHOW	
2987	11 A000		LXI	D,TEXT	
298A	13	FNDLP:	INX	D	
298B	CD 2FD0		CALL	LDE	
298E	4F		MOV	C,A	
298F	1B		DCX	D	
2990	87		ADD	A	
2991	D8		RC		
2992	CD 2FD0		CALL	LDE	
2995	95		SUB	L	
2996	47		MOV	B,A	
2997	13		INX	D	
2998	79		MOV	A,C	
2999	9C		SBB	H	
299A	3804		JRC	FL1	
299C	1B		DCX	D	
299D	B0		ORA	B	
299E	C9	FI2:	RET		
299F	13	FNDNXT:	INX	D	
29A0	13	FL1:	INX	D	
29A1	CD 2FD0	FNDSKP:	CALL	LDE	
29A4	FE0D		CPI	CR	
29A6	20F8		JRNZ	FL1	
29A8	13		INX	D	
29A9	18DF		JMPR	FNDLP	
29AB	E7	TSTV:	RST	RSTIGN	
29AC	FE25		CPI	'%'	; PEEK-POKE?
29AE	281D		JRZ	TSTV0	
29B0	D640		SUI	'e'	
29B2	D8		RC		
29B3	201C		JRNZ	TV1	
29B5	13		INX	D	
29B6	EF		RST	RSTPAR	
29B7	29		DAD	H	
29B8	DA 2A4C		JC	QHOW	
29BB	D5	TSTB:	FINISH	D	
29BC	EB		XCHG		
29BD	CD 2899		CALL	SIZE	
29C0	CD 202F		CALL	COMP	
29C3	38B8		JRC	ASORRY	
29C5	2A 4E52		LHLD	TXTUNF	
29C8	2B		DCX	H	
29C9	2B		DCX	H	
29CA	19		DAD	D	
29CB	D1		POP	D	
29CC	C9		RET		
			; %(ADDR) PEEK-POKE CALL		
29CD	13	TSTV0:	INX	D	
29CE	EF		RST	RSTPAR	; GET ADDR
29CF	AF		XRA	A	; CLEAR CY
29D0	C9		RET		; AND GO BACK
29D1	FE1B	TV1:	CPI	27	
29D3	3F		CMC		

10 + 9

if in size (range)

~~FINISH~~

.MAIN. -

```

29D4    D8          RC
29D5    13          INX      D
          ; IS SECOND CHARACTER ALSO ALPHA?
29D6    6F          MOV     L,A      ; SAVE FIRST      ONE
29D7    CD 2FD0     CALL    LDE
29DA    FE41       CPI     'A'
29DC    3826       JRC     DEVV4   ; IF NOT IN RANGE A-Z
29DE    FE5B       CPI     'Z'+1
29E0    3022       JRNC   DEVV4   ; THEN SEARCH
29E2    C5         PUSH   B
29E3    D5         PUSH   D
29E4    67         MOV     H,A      ; SECOND CHAR TO H
29E5    0607       MVI     B,PARNUM   ; B - ITERATION
          CTR
29E7    11 20A2    LXI     D,DEVLST   ; DE - SEARCH TA
          BLE
29EA    1A         LDAX   D      ; GET FIRST      ENTRY
29EB    13         INX     D
29EC    BD         CMP    L
29ED    1A         LDAX   D
29EE    13         INX     D
29EF    200B       JRNZ   DEVV2
29F1    BC         CMP    H
29F2    200B       JRNZ   DEVV2
          ; MATCH FOUND - FIGURE OUT LOOKUP INDEX
29F4    78         MOV     A,B
29F5    C61A       ADI    26
29F7    6F         MOV     L,A
29F8    D1         POP    D
29F9    13         INX     D      ; BUMP CHAR PTR
29FA    1807       JMPR   DEVV3
          ; MISMATCH - LOOP BACK IF POSS
29FC    10EC       DEVV2: DJNZ   DEVV1
          ; NOT POSSIBLE - RETURN NOT A VAR
29FE    D1         POP    D
29FF    C1         POP    B
2A00    1B         DCX   D      ; BACKUP TO CHAR START
2A01    37         STC           ; SET CARRY
2A02    C9         RET
2A03    C1         DEVV3: POP    B
2A04    7D         DEVV4: MOV    A,L
2A05    21 4E6C    LXI     H,VARBGN-2
2A08    07         RLC
2A09    85         ADD    L
2A0A    6F         MOV    L,A
2A0B    3E00       MVI    A,0
2A0D    8C         ADC    H
2A0E    67         MOV    H,A
2A0F    C9         RET
          ; TSTCH AND TSTNUM
2A10    E3         TSTCH: XTHL
2A11    E7         RST     RSTIGN
2A12    BE         CMP    M
2A13    23         INX     H
2A14    2807       JRZ    TC1

```

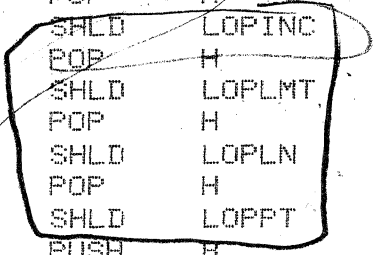
.MAIN. -

2A16	C5		PUSH	B
2A17	4E		MOV	C,M
2A18	0600		MVI	B,0
2A1A	09		DAD	B
2A1B	C1		POP	B
2A1C	1B		DCX	D
2A1D	13	TC1:	INX	D
2A1E	23		INX	H
2A1F	E3		XTHL	
2A20	C9		RET	
2A21	21 0000	TSTNUM:	LXI	H,0
2A24	44		MOV	B,H
2A25	E7		RST	RSTIGN
2A26	FE30	TN1:	CPI	'0'
2A28	D8		RC	
2A29	FE3A		CPI	3AH
2A2B	D0		RNC	
2A2C	3EF0		MVI	A,0F0H
2A2E	A4		ANA	H
2A2F	201B		JRNZ	QH0W
2A31	04		INR	B
2A32	C5		PUSH	B
2A33	44		MOV	B,H
2A34	4D		MOV	C,L
2A35	29		DAD	H
2A36	29		DAD	H
2A37	09		DAD	B
2A38	29		DAD	H
2A39	CD 2FD0		CALL	LDE
2A3C	13		INX	D
2A3D	E60F		ANI	00FH
2A3F	85		ADD	L
2A40	6F		MOV	L,A
2A41	3E00		MVI	A,0
2A43	8C		ADC	H
2A44	67		MOV	H,A
2A45	C1		POP	B
2A46	CD 2FD0		CALL	LDE
2A49	F2 2A26		JP	TN1
2A4C	D5	QH0W:	PUSH	D
2A4D	11 203B	AH0W:	LXI	D,0H0W
2A50	C3 2949		JMP	ERROR
		: MVUP, MVDOWN, POPA, AND PUSHA		
2A53	CD 202F	MVUP:	CALL	COMP
2A56	C8		RZ	
2A57	CD 2FD0		CALL	LDE
2A5A	E5		PUSH	H
2A5B	60		MOV	H,B
2A5C	69		MOV	L,C
2A5D	CD 2FEB		CALL	STHL
2A60	E1		POP	H
2A61	13		INX	D
2A62	03		INX	B
2A63	18EE		JMPR	MVUP
2A65	78	MVDOWN:	MOV	A,B

@ preservation

.MAIN. -

2A66	92		SUB	D
2A67	C2 2A6D		JNZ	MD1
2A6A	79		MOV	A,C
2A6B	93		SUB	E
2A6C	C8		RZ	
2A6D	1B	MD1:	DCX	D
2A6E	2B		DCX	H
2A6F	CD 2FD0		CALL	LDE
2A72	CD 2FE8		CALL	STHL
2A75	18EE		JMPR	MVDOWN
2A77	C1	POFA:	POP	B
2A78	E1		POP	H
2A79	22 4EC9		SHLD	LOPVAR
2A7C	7C		MOV	A,H
2A7D	B5		ORA	L
2A7E	2810		JRZ	PP1
2A80	E1		POP	H
2A81	22 4ECB		SHLD	LOPINC
2A84	E1		POP	H
2A85	22 4ECD		SHLD	LOPLMT
2A88	E1		POP	H
2A89	22 4ECF		SHLD	LOPLN
2A8C	E1		POP	H
2A8D	22 4ED1		SHLD	LOPPT
2A90	C5	PP1:	PUSH	B
2A91	C9		RET	
2A92	21 BOA4	PUSHA:	LXI	H,-STKLMT
2A95	C1		POP	B
2A96	39		DAD	SP
2A97	D2 297C		JNC	OSORRY
2A9A	2A 4EC9		LHLD	LOPVAR
2A9D	7C		MOV	A,H
2A9E	B5		ORA	L
2A9F	2813		JRZ	PU1
2AA1	2A 4ED1		LHLD	LOPPT
2AA4	E5		PUSH	H
2AA5	2A 4ECF		LHLD	LOPLN
2AA8	E5		PUSH	H
2AA9	2A 4ECD		LHLD	LOPLMT
2AAC	E5		PUSH	H
2AAD	2A 4ECB		LHLD	LOPINC
2AB0	E5		PUSH	H
2AB1	2A 4EC9		LHLD	LOPVAR
2AB4	E5	PU1:	PUSH	H
2AB5	C5		PUSH	B
2AB6	C9		RET	
			; PRTSTG, QTSTG, PRTNUM, PRTLN	
2AB7	97	PRTSTG:	SUB	A
2AB8	47	PS1:	MOV	B,A
2AB9	CD 2FD0	PS2:	CALL	LDE
2ABC	13		INX	D
2ABD	B8		CMP	B
2ABE	C8		RZ	
2ABF	DF		RST	RSTOCH
2AC0	FE0D		CPI	CR



Handwritten notes:
 LDI D,LOPINC
 LDI B,LOPINC
 LDI ...

LDIR this

.MAIN. --

```

2AC2    20F5                JRNZ    PS2
2AC4    C9                 RET
2AC5    CF                QTSTG:  TSTC    <"/>,QT3E    RST    1
2AC6    22                +      .BYTE  <"/>
2AC7    0E                +      .BYTE  QT3--, -1
                +J

2AC8    3E22              MVI     A, <"/>
2ACA    CD 2AB8          QT1:    CALL   PS1
2ACD    FE0D            QT2:    CPI    CR
2ACF    E1              POP     H
2AD0    CA 2550          JZ     RUNNXL
2AD3    23              INX    H
2AD4    23              INX    H
2AD5    E9              PCHL
2AD6    CF            QT3:    TSTCC   027H, QT5E    RST    1
2AD7    27                +      .BYTE  027H
2AD8    04                +      .BYTE  QT5--, -1
                +J

2AD9    3E27              MVI     A, 027H
2ADB    18ED            JMPR   QT1
2ADD    C9                QT5:    RET
2ADE    7B            PRTCHS:  MOV    A, E
2ADF    B8              CMP    B
2AE0    C8              RZ
2AE1    CD 2FD0          CALL   LDE
2AE4    DF              RST   RSTOCH
2AE5    13              INX    D
2AE6    18F6            JMPR   PRTCHS
2AE8                                PRTNUM == .
2AEB    0600            PN3:    MVI     B, 0
2AEA    CD 28FA          CALL   CHKSGN
2AED    F2 2AF3          JP     PN4
2AF0    062D            MVI     B, <"/>
2AF2    0D              DCR    C
2AF3    D5            PN4:    PUSH   D
2AF4    11 000A          LXI    D, 10
2AF7    D5              PUSH   D
2AF8    0D              DCR    C
2AF9    C5              PUSH   B
2AFA    CD 28E5          PN5:    CALL   DIVIDE
2AFD    78              MOV    A, B
2AFE    B1              ORA    C
2AFF    CA 2B09          JZ     PN6
2B02    E3              XTHL
2B03    2D              DCR    L
2B04    E5              PUSH   H
2B05    60              MOV    H, B
2B06    69              MOV    L, C
2B07    18F1            JMPR   PN5
2B09    C1            PN6:    POP     B
2B0A    0D            PN7:    DCR    C
2B0B    79              MOV    A, C
2B0C    B7              ORA    A
2B0D    FA 2B15          JM     PN8
2B10    3E20            MVI     A, <"/>

```


.MAIN. -

2B12	DF		RST	RSTOCH		
2B13	18F5		JMPR	PN7		
2B15	78	PN8:	MOV	A,B		
2B16	B7		ORA	A		
2B17	C4 2C7C		CNZ	OUTCH		
2B1A	5D		MOV	E,L		
2B1B	7B	PN9:	MOV	A,E		
2B1C	FE0A		CPI	10		
2B1E	D1		POP	D		
2B1F	C8		RZ			
2B20	C630		ADI	'0'		
2B22	DF		RST	RSTOCH		
2B23	18F6		JMPR	PN9		
2B25	CD 2FD0	PRTLN:	CALL	LDE		
2B28	6F		MOV	L,A		
2B29	13		INX	D		
2B2A	CD 2FD0		CALL	LDE		
2B2D	67		MOV	H,A		
2B2E	13		INX	D		
2B2F	0E04		MVI	C,4		
2B31	CD 2AE8		CALL	PRTNUM		
2B34	3E20		MVI	A,' '		
2B36	DF		RST	RSTOCH		
2B37	C9		RET			
2B38	5456	TAB2:	ITEM	'TV',PUTCDE	.ASCII	'TV'
2B3A	A3	+	DEFF	PUTCDE	.BYTE	(PUTCD>8)!80H
2B3B	FC	+	.BYTE	PUTCD&OFFH		
		+]				
2B3C	4D55		ITEM	'MU',PUTMUE	.ASCII	'MU'
2B3E	A3	+	DEFF	PUTMUE	.BYTE	(PUTMU>8)!80H
2B3F	F3	+	.BYTE	PUTMU&OFFH		
		+]				
2B40	26		ITEM	'&',PUTIOE	.ASCII	'&'
2B41	A3	+	DEFF	PUTIOE	.BYTE	(PUTIO>8)!80H
2B42	D9	+	.BYTE	PUTIO&OFFH		
		+]				
2B43	43414C4C		ITEM	'CALL',DOCALL	E	.ASCII
		'CALL'				
2B47	A4	+	DEFF	DOCALL	E	.BYTE (DOCALL
		>8)!80H				
2B48	06	+	.BYTE	DOCALL	&OFFH	
		+]				
2B49	2E		ITEM	'.',REMC	.ASCII	'.'
2B4A	A6	+	DEFF	REMC	.BYTE	(REM>8)!80H
2B4B	E7	+	.BYTE	REM&OFFH		
		+]				
2B4C	24		ITEM	'\$',BCDMAT[E	.ASCII	'\$'
2B4D	A1	+	DEFF	BCDMAT[E	.BYTE	(BCDMAT>8)!80H
2B4E	7A	+	.BYTE	BCDMAT&OFFH		
		+]				
2B4F	3A		.BYTE	':'		
2B50	68		TOKEN	68H,TLIST[E	.BYTE	68H
2B51	A1	+	DEFF	TLIST[E	.BYTE	(TLIST>8)!80H
2B52	55	+	.BYTE	TLIST&OFFH		
		+]				

.MAIN. -

```

2B53  3A          .BYTE  ':/
2B54  74          TOKEN  74H,TOUTPUI  .BYTE  74H
2B55  A1          +      DEFF  TOUTPUI  .BYTE  (TOUTPU>8)!80H
2B56  61          +      .BYTE  TOUTPU&OFFH
                +]

2B57  3A          .BYTE  ':/
2B58  73          TOKEN  73H,TINPUTI  .BYTE  73H
2B59  A1          +      DEFF  TINPUTI  .BYTE  (TINPUT>8)!80H
2B5A  51          +      .BYTE  TINPUT&OFFH
                +]

2B5B  3A          .BYTE  ':/
2B5C  70          TOKEN  70H,TRETURI  .BYTE  70H
2B5D  A4          +      DEFF  TRETURI  .BYTE  (TRETUR>8)!80H
2B5E  0C          +      .BYTE  TRETUR&OFFH
                +]

2B5F  3A          .BYTE  ':/
2B60  6A          TOKEN  6AH,TLOADI  .BYTE  6AH
2B61  A1          +      DEFF  TLOADI  .BYTE  (TLOAD>8)!80H
2B62  67          +      .BYTE  TLOAD&OFFH
                +]

2B63  2A          .BYTE  '*
2B64  74          TOKEN  74H,TPRINTI  .BYTE  74H
2B65  A1          +      DEFF  TPRINTI  .BYTE  (TPRINT>8)!80H
2B66  5E          +      .BYTE  TPRINT&OFFH
                +]

2B67  53544F50   ITEM   'STOP',STOPI  .ASCII  'STOP
                /

2B68  A4          +      DEFF  STOPI  .BYTE  (STOP>8)!80H
2B6C  75          +      .BYTE  STOP&OFFH
                +]

2B6D  A7          DEFF  DEFLTI  .BYTE  (DEFLT>8)!80H
2B6E  59          +      .BYTE  DEFLT&OFFH
                +]

2B6F  76          TAB3:  TOKEN  76H,RNDI  .BYTE  76H
2B70  A8          +      DEFF  RNDI  .BYTE  (RND>8)!80H
2B71  76          +      .BYTE  RND&OFFH
                +]

2B72  4B4E       ITEM   'KN',GETPOTI  .ASCII  'KN'
2B74  A8          +      DEFF  GETPOTI  .BYTE  (GETPOT>8)!80H
2B75  A6          +      .BYTE  GETPOT&OFFH
                +]

2B76  5452       ITEM   'TR',GETTRGI  .ASCII  'TR'
2B78  A8          +      DEFF  GETTRGI  .BYTE  (GETTRG>8)!80H
2B79  B6          +      .BYTE  GETTRG&OFFH
                +]

2B7A  4A58       ITEM   'JX',GETJXI  .ASCII  'JX'
2B7C  A8          +      DEFF  GETJXI  .BYTE  (GETJX>8)!80H
2B7D  BE          +      .BYTE  GETJX&OFFH
                +]

2B7E  4A59       ITEM   'JY',GETJYI  .ASCII  'JY'
2B80  A8          +      DEFF  GETJYI  .BYTE  (GETJY>8)!80H
2B81  CA          +      .BYTE  GETJY&OFFH
                +]

2B82  4B50       ITEM   'KP',GETKBI  .ASCII  'KP'
2B84  A3          +      DEFF  GETKBI  .BYTE  (GETKB>8)!80H

```

.MAIN. -

```

2B85    CE          +          .BYTE    GETKB&OFFH
          +]

2B86    5058        +          ITEM      'PX',PIXFUNI  .ASCII   'PX'
2B88    A1          +          DEFF PIXFUNI  .BYTE    (PIXFUN>8)!80H
2B89    BE          +          .BYTE    PIXFUN&OFFH
          +]

2B8A    26          +          ITEM      '&',IOFUNI   .ASCII   '&'
2B8B    A3          +          DEFF IOFUNI   .BYTE    (IOFUN>8)!80H
2B8C    E8          +          .BYTE    IOFUN&OFFH
          +]

2B8D    414253     +          ITEM      'ABS',ABSI   .ASCII   'ABS'

2B90    A8          +          DEFF ABSI     .BYTE    (ABS>8)!80H
2B91    92          +          .BYTE    ABS&OFFH
          +]

2B92    535A       +          ITEM      'SZ',SIZEI   .ASCII   'SZ'
2B94    A8          +          DEFF SIZEI   .BYTE    (SIZE>8)!80H
2B95    99          +          .BYTE    SIZE&OFFH
          +]

2B96    A8          +          DEFF NOTFI   .BYTE    (NOTF>8)!80H
2B97    46          +          .BYTE    NOTF&OFFH
          +]

2B98    3E3D       +          TAB6:  ITEM      '>=',XPR1I  .ASCII   '>='
2B9A    A7          +          DEFF XPR1I  .BYTE    (XPR1>8)!80H
2B9B    6B          +          .BYTE    XPR1&OFFH
          +]

2B9C    23         +          ITEM      '#',XPR2I  .ASCII   '#'
2B9D    A7          +          DEFF XPR2I  .BYTE    (XPR2>8)!80H
2B9E    71          +          .BYTE    XPR2&OFFH
          +]

2B9F    3E         +          ITEM      '>',XPR3I  .ASCII   '>'
2BA0    A7          +          DEFF XPR3I  .BYTE    (XPR3>8)!80H
2BA1    77          +          .BYTE    XPR3&OFFH
          +]

2BA2    3D         +          ITEM      '= ',XPR5I  .ASCII   '= '
2BA3    A7          +          DEFF XPR5I  .BYTE    (XPR5>8)!80H
2BA4    86          +          .BYTE    XPR5&OFFH
          +]

2BA5    3C3D       +          ITEM      '<=',XPR4I  .ASCII   '<='
2BA7    A7          +          DEFF XPR4I  .BYTE    (XPR4>8)!80H
2BA8    7E          +          .BYTE    XPR4&OFFH
          +]

2BA9    3C         +          ITEM      '<',XPR6I  .ASCII   '<'
2BAA    A7          +          DEFF XPR6I  .BYTE    (XPR6>8)!80H
2BAB    8C          +          .BYTE    XPR6&OFFH
          +]

2BAC    A7         +          DEFF XPR7I  .BYTE    (XPR7>8)!80H
2BAD    92          +          .BYTE    XPR7&OFFH
          +]

2BAE                                RANEND  ==          .
2BAE    3A 4E5B     +          GLED:  LDA      EDFLG
2BB1    A7          +          ANA      A
2BB2    2814       +          JRZ      GLEDA
2BB4    11 4ED4    +          LXI      D, BUFFER
2BB7    CD 2A21    +          CALL     TSTNUM

```

.MAIN. -

```

2BBA    CD 2982          CALL    FNDLN
2BBB    3E3F            MVI    A, '?'
2BBF    C0             RNZ
2BC0    13             INX    D
2BC1    CD 2BCF        CALL    GLEDB
2BC4    AF            XRA    A
2BC5    32 4E5B        STA    EDFLG
2BC8    ED5B 4E5C      GLEDA:  LDED    EDPTR
2BCC    CD 2FD0        CALL    LDE
2BCF    13             INX    D
2BD0    ED53 4E5C      GLEDB:  SDED    EDPTR
2BD4    C9             RET
2BD5    11 4ED4        GETLN:  LXI    D, BUFFER
2BD8    32 4E5B        STA    EDFLG
2BDB    DF            GL1:   RST    RSTOCH ; PROMPT OR ECHO
2BDC    C5            GL2:   PUSH   B
2BDD    E5            PUSH   H
2BDE    D5            PUSH   D
                ; PLACE UP CURSOR BLOCK
2BDF    0EAA          MVI    C, 0AAH
2BE1    CD 2D88        CALL    CURSE
                ; RETURN CHAR FROM NEXT LINE #
2BE4    21 4E62        GL2A:  LXI    H, NLLNCT
2BE7    7E            MOV    A, M ; SENSE FLAG
2BE8    A7            ANA    A
2BE9    2838          JRZ    GL2C
2BEB    35            DCR    M
                ; FIRST TIME THRU?
2BEC    FE05          CPI    5
2BEE    200C          JRNZ   GL2B ; JUMP IF NOT
                ; GET PREVIOUS LINE # AND BUMP IT
2BF0    2A 4E64        LHLD   OLDLN
2BF3    11 000A        LXI    D, 10
2BF6    19            DAD    D
2BF7    CBBC          RES    7, H ; ALLOW NEG
2BF9    22 4E60        GL2J:  SHLD  NLLNLN ; MOVE TO WORKING RAM CE
                LL
                ; COMPUTE DIVISION SUBTRACTOR
2BFC    FF            GL2B:  INDEXW 1[INTP%.IFE .INTP., [RST 7]]
2BFD    5B            +.BYTE 90+1]
2BFE    2F10          .WORD  TBLDIV-2
2C00    2A 4E60        LHLD   NLLNLN
2C03    0600          MVI    B, 0
2C05    A7            GL2E:  ANA    A
2C06    ED52          DSBC   D
2C08    FA 2C0E        JM     GL2F
2C0B    04            INR    B
2C0C    18F7          JMPR   GL2E
2C0E    19            GL2F:  DAD    D
2C0F    22 4E60        SHLD  NLLNLN
2C12    21 4E63        LXI    H, NLLNZS
2C15    78            MOV    A, B
2C16    A7            ANA    A
2C17    2005          JRNZ   GL2G
2C19    7E            MOV    A, M

```

.MAIN. -

2C1A	A7		ANA	A	
2C1B	28C7		JRZ	GL2A	; YES - JUMP BACK
2C1D	AF		XRA	A	
2C1E	C630	GL2G:	ADI	'0'	; MAKE ASCII
2C20	77		MOV	M,A	; SET NONZERO FLAG
2C21	180B		JMPR	GL2D	
					; NOTHIN FANCY
2C23	CD 2D9C	GL2C:	CALL	CHKIO	; GET NORMAL CHARACTER
2C26	D1		POP	D	
2C27	12		STAX	D	; STUFF CHAR AS DELIMITE
		R			
2C28	D5		PUSH	D	
2C29	FE66		CPI	EDKEY	
2C2B	CC 2BAE		CZ	GLED	
2C2E	D1	GL2D:	POP	D	
2C2F	E1		POP	H	
2C30	C1		POP	B	
2C31	12	GL3:	STAX	D	
2C32	FE1F		CPI	RUBOUT	
2C34	202F		JRNZ	GL4	
2C36	7B		MOV	A,E	
2C37	FED4		CPI	BUFFER&OFFH	
2C39	28A1		JRZ	GL2	
2C3B	1B		DCX	D	
2C3C	1A		LDAX	D	
2C3D	FE68		CPI	68H	; TOKEN TO RUB OUT?
2C3F	3007		JRNC	TOKIN	
2C41	CD 2E86		CALL	PNOTE	
2C44	3E1F		MVI	A,RUBOUT	
2C46	1893		JMPR	GL1	
2C48	D5	TOKIN:	PUSH	D	
2C49	CD 2EEF		CALL	TOKEPT	
2C4C	7E	TOKER:	MOV	A,M	
2C4D	E5		PUSH	H	
2C4E	E67F		ANI	7FH	
2C50	CD 2E86		CALL	PNOTE	
2C53	3E1F		MVI	A,RUBOUT	
2C55	CD 2CCE		CALL	VDM	
2C58	E1		POP	H	
2C59	7E		MOV	A,M	
2C5A	23		INX	H	
2C5B	07		RLC		
2C5C	30EE		JRNC	TOKER	
2C5E	3E1F	TOKEQ:	MVI	A,RUBOUT	
2C60	DF		RST	RSTOCH	; ECHO ONE RUBOUT CHAR
2C61	D1		POP	D	
2C62	C3 2BDC	GL9:	JMP	GL2	
2C65	FE0D	GL4:	CPI	CR	
2C67	CA 2C74		JZ	GL5	
2C6A	7B		MOV	A,E	
2C6B	FE3C		CPI	BUFEND&OFFH	
2C6D	28F3		JRZ	GL9	
2C6F	1A		LDAX	D	
2C70	13		INX	D	
2C71	C3 2BDB		JMP	GL1	

.MAIN. --

```

2C74 13          GL5:  INX      D
2C75 13          INX      D
2C76 3EFF       MVI      A,OFFH
2C78 12          STAX     D
2C79 1B          DCX      D
2C7A 3E0D       CRLF:  MVI      A,CR
                ; SUBROUTINE TO SIMULATE A CHARACTER DISPLAY IN
                ; THE ARCADE FRAME BUFFER.  THE SIMULATED VDM HAS
                ; DIMENSIONS 26 CHARS BY 11 LINES.  THE CHARACTER
                ; GRAPHICS ARE 5 X 7
                ; IN A 6 X 8 FRAME.  ALTERNATE FONT IS USED TO GET THIS.
                ; THE 64 UPPER CASE ASCII CHARACTERS ARE DISPLAYED BY THIS
                ; HANDLER.  THE ASCII CONTROL CHARACTERS CARRIAGE RETURN AND
                ; RUBOUT ARE ALSO PROCESSED BY THIS HANDLER.  CR CAUSES
                ; THE DISPLAY TO GO TO THE NEXT LINE OF THE DISPLAY, WITH
                ; SCROLL UP IF NECESSARY.  RUBOUT CAUSES THE CURSOR TO MOVE
                ; BACKWARDS ONE CHARACTER POSITION.
                ; CHARACTER TO DISPLAY IS IN A.  THE ALTERNATE REGISTER SET
                ; IS USED.
2C7C E5          OUTCH:  PUSH     H
2C7D D5          PUSH     D
2C7E C5          PUSH     B
2C7F F5          PUSH     PSW
2C80 57          MOV      D,A
2C81 3A 4EC2     LDA      TAPEST
2C84 FE06       CPI      6
2C86 282F       JRZ      VDMFRG
2C88 E602       TURDC1: ANI      2          ; WRITE TO TAPE WANTED?
2C8A 2830   JRZ      VDMCAL
2C8C 4A          TURDC2: MOV      C,D
2C8D CB01       RLCR      C
                ; YEP
                ; WRITE START BIT
2C8F DB12       TAPCH0: IN      TAPEID  ; INPUT FF STATE
2C91 E602       ANI      2          ; WAIT FOR IT TO GO NONZ
                ERO
2C93 28FA       JRZ      TAPCH0  ; MEANING START BIT WRIT
                TEN
2C95 060A       MVI      B,10      ; B = # OF BITS TO WRITE
                ; WAIT ABOUT 1.8 MILLISECONDS TO MOVE BEYOND CHANGE WINDOW
2C97 3E00       TAPCH1: MVI      A,192
2C99 3D          TAPCH2: DCR      A
2C9A 20FD       JRNZ     TAPCH2
                ; ARE WE DONE WRITING?
2C9C 05          DCR      B
    
```

OUT

.MAIN. -

```

2C9D    281D'          JRZ      VDMCAL    ; JUMP IF SO
          ; (POSITION IS BIT 1 )
2C9F    DB12          IN       TAPEIO    ; KEEP READING INPUT
2CA1    5F            MOV      E,A
2CA2    DB12          TAPCH3: IN      TAPEIO    ; UNTIL IT TOGGLES
2CA4    AB            XRA      E
2CA5    E602          ANI      2
2CA7    28F9          JRZ      TAPCH3
2CA9    7B            MOV      A,E      ; E = WHAT WE GOT NOW
2CAA    A9            XRA      C      ; C = WHAT WE WANT
2CAB    E602          ANI      2      ; DID WE 'GET IT?'
2CAD    2802          JRZ      TAPCH4    ; JUMP IF SO
2CAF    DB12          IN       TAPEIO    ; NO - READ AGAIN TO TOG

          GLE
2CB1    C8C9          TAPCH4: SET    1,C      ; SET EACH BIT WRITTEN T
          0 1
2CB3    C809          RRCR      C      ; SHIFT INTO NEXT POSITI
          ON
2CB5    18E0          JMPR     TAPCH1    ; SO STOP BIT WILL GO OU
          T
          ; MODE 6 - TOKEN COMING THRU?
2CB7    7A            VDMFRG: MOV    A,D
2CB8    FE68          CPI      68H
2CBA    38D0          JRC      TURDC2    ; NO - TWEEDLE IT TO TAP
          E
          ; YES - FALL INTO ...
2CBC    7A            VDMCAL: MOV    A,D
2CBD    CD 2CCE          CALL     VDM
2CC0    F1            POP      PSW
2CC1    C1            POP      B
2CC2    D1            POP      D
2CC3    E1            POP      H
2CC4    C9            RET

          ; EVEN STRANGER GUYS ENTER HERE ...
2CC5    6F            VDMTOK: MOV    L,A
2CC6    3A 4EC2          LDA      TAPEST
2CC9    FE06          CPI      6
2CCB    7D            MOV      A,L
2CCC    28AE          JRZ      OUTCH

          ; SOME FUNNY GUYS ENTER HERE
2CCE    FE0D          VDM:      CPI      CR
2CD0    282E          JRZ      VDMOCR
2CD2    FE1F          CPI      RUBOUT    ; TRANSLATE TRASH TO ?
2CD4    280F          JRZ      VDM1
2CD6    3804          JRC      FILT1
2CD8    FE78          CPI      78H
2CDA    3802          JRC      FILT2
2CDC    3E3F          FILT1:  MVI      A,'?'
2CDE    FE68          FILT2:  CPI      68H      ; TOKEN TO PRINT?
2CE0    304F          JRNC     TOKEP      ; JUMP IF SO

          ; PLAY NOTE FOR THIS CHAR
2CE2    CD 2E86          CALL     PNOTE

          ; NON NEW LINE CHAR - UNWRITE OLD CURSOR
2CE5    CD 2D46          VDM1:   CALL     UCURSE
2CE8    CD 235A          CALL     LDVDMC

```

.MAIN. -

```

2CEB FE1F CPI RUBOUT ; WAS THAT RUBOUT?
2CED 201F JRNZ VDM3 ; JUMP IF NOT
; RUBOUT ENTERED - SO RUB OUT
2CEF 7D MOV A,L ; GET X
2CF0 A7 ANA A ; IS X = 0?
2CF1 2805 JRZ VDM2 ; YES - JUMP
2CF3 D606 SUI 6 ; NO - BACKUP X
2CF5 6F MOV L,A
2CF6 1831 JMPR VMDN1 ; AND JOIN STORE BACK
2CF8 2E96 VDM2: MVI L,150
2CFA 7C MOV A,H
2CFB D608 SUI 8
2CFD 67 MOV H,A
2CFE 1829 JMPR VMDN1
; NEW LINE CHAR - DID WE JUST WRAP AROUND
2D00 3A 4E54 VDMOCR: LDA VDMNLF ; CHECK OLD GLORY
2D03 A7 ANA A
2D04 2026 JRNZ VMDON ; YES - SKIP DIDDLING
2D06 CD 2D46 CALL UCURSE ; NO - UNWRITE CURSOR
2D09 CD 2D4A CALL NEWLIN ; GO TO NEXT LINE
2D0C 181E JMPR VMDON ; AND QUIT
; NORMAL CHARACTER ENTERED - DISPLAY IT
2D0E 54 VDM3: MOV D,H ; COORDINATES TO DE
2D0F 5D MOV E,L
2D10 F680 ORI 80H ; ALT FONT THE CHAR
2D12 0E18 MVI C,011000B ; OR WRITE THE C
HAR
2D14 DD21 4E66 LXI X,ALTFON ; USING ALTERNAT
E CHAR FONT
2D18 FF CHRDIS ; IT 1[INTP%.IFE ,INTP.,IRST
7]]
+.BYTE 50+0]
2D19 32 MOV A,L ; ADVANCE X POINTER
2D1A 7D ADI 6
2D1B C606 MOV L,A
2D1D 6F CPI 156 ; END OF LINE?
2D1E FE9C JRNZ VMDN1 ; NO - JUMP
2D20 2007 CALL NEWLIN ; YES - NEW 1 LINE
2D22 CD 2D4A MVI A,1 ; AND SET NEW LINE FORCE
2D25 3E01 D FLAG
2D27 1804 JMPR VMDN2
2D29 CD 2374 VMDN1: CALL STVDMC
2D2C AF VMDON: XRA A ; CLEAR NEW LINE FORCED
FLAG
2D2D 32 4E54 VMDN2: STA VDMNLF
2D30 C9 RET
; ROUTINE TO DISPLAY A TOKEN IN FULL FORM
2D31 CD 2EEF TOKEP: CALL TOKEPT
2D34 7E TOKEP1: MOV A,M
2D35 E67F ANI 7FH
2D37 E5 PUSH H
2D38 CD 2CC5 CALL VDMTOK
2D3B E1 POP H
2D3C 7E MOV A,M
2D3D 23 INX H

```


.MAIN. -

```

2D3E 07 RLC
2D3F 30F3 JRNC TOKEP1
2D41 3E20 TOKEP2: MVI A, ' ' ; PUT SPACE AFTER TOKEN
2D43 C3 2CC5 JMP VDMTOK ; AND RETURN
; SUBROUTINE TO UNWRITE THE CURSOR
2D46 0E00 UCURSE: MVI C, 0
2D48 183E JMPR CURSE
; SUBROUTINE TO DISPLAY NEW LINE
2D4A CD 235A NEWLIN: CALL LDVDMC
; IS SCROLL UP NEEDED?
2D4D 2E00 MVI L, 0
2D4F 7C MOV A, H
2D50 FE50 CPI 80
2D52 202D JRNZ NEWL1 ; JUMP IF NOT NEEDED
; SCROLL UP IS NEEDED
2D54 CD 2374 CALL STVDMC
2D57 21 4DC0 LXI H, 4DC0H
2D5A 7E SCRL9: MOV A, M
2D5B E655 ANI 01010101B
2D5D 77 MOV M, A
2D5E 23 INX H
2D5F 7D MOV A, L
2D60 FE20 CPI 20H
2D62 20F6 JRNZ SCRL9
2D64 0604 MVI B, 4
2D66 C5 SCRLP: PUSH B
2D67 21 4000 LXI H, NORMEM
2D6A 11 4050 LXI D, NORMEM+80
2D6D 01 980E LXI B, 0980EH
2D70 1A SCRUP: LDAX D
2D71 AE XRA M
2D72 E6AA ANI 10101010B
2D74 AE XRA M
2D75 77 MOV M, A
2D76 23 INX H
2D77 13 INX D
2D78 10F6 DJNZ SCRUP
2D7A 0D DCR C
2D7B 20F3 JRNZ SCRUP
2D7D C1 POP B
2D7E 10E6 DJNZ SCRLP
2D80 C9 RET
;
2D81 C608 NEWL1: ADI 8
2D83 67 MOV A, H
2D84 CD 2374 CALL STVDMC
2D87 C9 RET
; SUBROUTINE TO PAINT CURSOR
; C = DATA TO PAINT 00 OR AA
2D88 F5 CURSE: PUSH PSW
2D89 CD 235A CALL LDVDMC
2D8C EB PCURS1: XCHG
2D8D AF XRA A
2D8E FF RELAB1[INTP%[L,IFE ,INTP.,[RST 7]]
2D8F 3A +.BYTE 58+0]

```



35

Handwritten notes and scribbles on the right side of the page, including some illegible text and a small diagram.

.MAIN. -

```

2D90      D30C          OUT      MAGIC
2D92      EB           XCHG
2D93      79           MOV      A,C
2D94      01 0806     LXI      B,0806H
2D97      CD 226F     CALL    BOXPUT
2D9A      F1          POP      PSW
2D9B      C9          RET

; NEW KEYBOARD HANDLER
; WITH SHIFT KEY ROLLOVER

2D9C      3A 4EC2     CHKIO:  LDA      TAPEST ; TAPE INPUT WANTED?
2D9F      3D          DCR      A
2DA0      2019       JRNZ    CHKIO0 ; NO - SKIP IT THEN
; YES - CHECK FOR ABORT KEYSTROKE
2DA2      CD 2E5C     CALL    KEYSN
2DA5      C2 246F     JNZ     INITO
; GET CHARACTER FROM THE BUFFER
2DAB      2A 4E20     LHLD   CONPRO
2DAB      7C          MOV     A,H
2DAC      BD          CMP     L ; ARE POINTERS EQUAL?
2DAD      28ED       JRZ     CHKIO ; YEP - LOOP WAITING
2DAF      6F          MOV     L,A ; MAKE OFFICIAL POINTER
2DB0      264E       MVI    H,TAPBUF>8
2DB2      4E          MOV     C,M
2DB3      CD 212C     CALL    BUMPTR
2DB6      32 4E21     STA    CONPTR ; UPDATE MY POINTER
2DB9      79          MOV     A,C ; PASS BACK THE CHARACTE
R
2DBA      C9          RET
2DBB      CD 2E5C     CHKIO0: CALL   KEYSN ; MAKE SURE PREVIOUS KEY
; RELEASED
2DBE      20FB       JRNZ    CHKIO0
; AWAIT DEBOUNCE TIMER COUNTDOWN
2DC0      21 4E55     CHKIO0: LXI    H,KEYTMR
2DC3      3606       MVI    M,6 ; SET IT
2DC5      7E          LOOPER: MOV   A,M
2DC6      A7          ANA    A
2DC7      20FC       JRNZ    LOOPER
; SAVE BACKGROUND COLOR
2DC9      3A 4EA2     LDA    DEVCL0
2DCC      F5          PUSH   PSW
; ASSUME FIRST LEVEL KEYCODE
2DCD      21 2F68     LXI    H,FIRSTL
2DD0      E5          GETK1: PUSH  H ; SAVE TABLE PTR
; SCAN ONLY FOR SHIFT KEYS
2DD1      21 2FBB     LXI    H,KTBL4
2DD4      11 FFEB     LXI    D,-21 ; ** SIZE OF LOOKUP TABL
E
2DD7      01 0414     LXI    B,0414H
2DDA      ED78       GETK2: INP   A ; INPUT FROM PORT
2DDC      E620       ANI    20H ; SHIFT KEY DOWN?
2DDE      2007       JRNZ    GETK3 ; JUMP IF YEP
2DE0      19          DAD    D ; ELSE TO NEXT TABLE
2DE1      0C          INR   C ; AND PORT
2DE2      10F6       DJNZ   GETK2
; NO SHIFT KEY IS DOWN - USE WHATEVER WE HAD BEF

```

out

.MAIN. -

```

ORE
2DE4 E1 POP H
2DE5 180F JMPR GETK5
; A SHIFT KEY IS DOWN - SAME OLD STORY?
2DE7 D1 GETK3: POP D ; DISCARD OLD BELIEFS
2DE8 3A 4EA2 LDA DEVCLO ; IS CURRENT SCREEN COLO
R
2DEB BE CMP M ; THE SAME AS WHAT WE WO
ULD SET?
2DEC 2807 JRZ GETK4 ; YEP - DON'T BOTHER WIT
H FEEDBACK
2DEE 7E MOV A,M ; ELSE SET NEW COLOR
2DEF 32 4EA2 STA DEVCLO
2DF2 CD 2E29 CALL WCLICK ; AND GO CLICK
2DF5 23 GETK4: INX H ; SKIP COLOR BYTE
; NOW SCAN FOR ANY 'NORMAL' KEY DEPRESSION
2DF6 CD 2E5C GETK5: CALL KEYSCLN
2DF9 28D5 JRZ GETK1 ; JUMP IF NO KEY DOWN
; WE GOT ONE - CONVERT TO ASCII
2DFB 3D DCR A ; BY TABLE LOOKUP
2DFC 4F MOV C,A
2DFD 0600 MVI B,0
2DFF 09 DAD B
2E00 F1 POP PSW ; RESTORE COLOR
2E01 32 4EA2 STA DEVCLO
2E04 7E MOV A,M ; GET CODE
2E05 A7 ANA A ; A HLT PERCHANCE?
2E06 284E JRZ INIJMP ; YEP - RESET
2E08 FE01 CPI 1 ; AN ERROR?
2E0A CA 2D9C JZ CHKIO ; YEP - GO DOIT AGAIN
; GOOD KEY...
2E0D F5 CHKIO2: PUSH PSW ; NEED WE GO 'CLICK'?
2E0E FE67 CPI NLLN ; REJECT TOKENS
2E10 300A JRNC NOCLK
2E12 FF INDEXB 1[INTP%.IFE .INTP.,[RST 7]]
2E13 5D +.BYTE 92+1]
2E14 2F04 .WORD NOTES-CR
2E16 3C INR A
2E17 2003 JRNZ NOCLK
2E19 CD 2E29 CALL WCLICK
2E1C F1 NOCLK: POP PSW
2E1D FE67 CPI NLLN
2E1F C0 RNZ
2E20 21 0005 LXI H,5
2E23 22 4E62 SHLD NLLNCT ; SET FLAG AND ZERO SUPR
ESS
2E26 3E0D MVI A,CR ; PASS BACK CR AS FIRST
CHAR
2E28 C9 RET
2E29 3A 4E57 WCLICK: LDA NEWTMR
2E2C A7 ANA A
2E2D 20FA JRNZ WCLICK
2E2F 3EFD MVI A,GO
2E31 32 4E59 STA MUZTON
2E34 3A 4EA6 LDA DEVTEM ; DON'T CLICK IF NT=0
    
```

*IN
RRC
JC* *15*

.MAIN. -

```

2E37    A7          ANA      A
2E38    C8          RZ
2E39    3E01       MVI     A,1
2E3B    32 4E57    STA     NEWTMR
2E3E    C9          RET

;
; SUBROUTINE TO CHECK FOR HLT KEY WHILE PGM RUNNING
2E3F    C5          WHATSU:  PUSH   B
2E40    D5          PUSH   D
2E41    CD 2E5C    CALL   KEYSCN ; GET KEY CODE
2E44    D602       SUI     2 ; FREEZE?
2E46    2807       JRZ    FRZKEY
2E48    3D          DCR     A
2E49    280B       JRZ    INIUMP
2E4B    FE01       CPI     1 ; RETURN Z STATUS FOR LI
; ST KEY
2E4D    180A       JMPR   FRZGBK ; ELSE GO BACK TO CALLER
2E4F    CD 2E5C    FRZKEY:  CALL   KEYSCN ; SCAN FOR NONZERO KEY T
; O REL
2E52    28FB       JRZ    FRZKEY
2E54    FE03       CPI     3 ; HLT NAILED?
2E56    CA 2472    INIUMP: JZ     INIT
2E59    D1          FRZGBK:  POP   D
2E5A    C1          POP   B
2E5B    C9          RET
; SUBROUTINE TO SCAN TINY BASIC KEYBOARD
2E5C    01 0414    KEYSCN: LXI   B,0414H ; B = CNT, C = PORT #
2E5F    11 4E6D    LXI   D,KEYTRK ; DE = KEYBOARD
; MEMORY
2E62    AF          XRA     A
2E63    FF          RANGED[INTP%[.IFE .INTP.,[RST 7]]
2E64    76          +.BYTE  118+0]
2E65    ED78       KYSCN1:  INF   A ; LOOK AT COLUMN
2E67    E61F       ANI    1FH ; ISOLATE THE RELEVANT
2E69    2006       JRNZ   KYSCN2 ; JUMP IF BITS HIGH
2E6B    0C          INR    C ; BUMP PORT #
2E6C    10F7       DJNZ   KYSCN1
2E6E    AF          XRA     A ; SET ZERO STATUS
2E6F    12          STAX   D ; NOTHIN - SAY ZIP
2E70    C9          RET
; DEPRESSION FOUND - JUMP UP AND DOWN
2E71    05          KYSCN2:  DCR   B
2E72    0E00       MVI    C,0 ; COME UP WITH BIT #
2E74    0F          KYSCN4:  RRC   ; SHIFT BIT OVER
2E75    3803       JRC    KYSCN3 ; JUMP IF THE ONE
2E77    0C          INR    C ; ELSE COUNT UP
2E78    18FA       JMPR   KYSCN4 ; AND TRY AGAIN
; FOUND BIT - ASSEMBLE KEYCODE
2E7A    79          KYSCN3:  MOV   A,C ; BIT # TO A
2E7B    07          RLC   ; * 4
2E7C    07          RLC
2E7D    B0          ORA   B ; COMBINE WITH COL #
2E7E    3C          INR   A

```

.MAIN. -

```

2E7F 47          MOV     B,A
2E80 1A          LDAX   D
2E81 A8          XRA    B
2E82 78          MOV    A,B
2E83 C8          RZ           ; QUIT IF THE SAME
2E84 12          STAX   D           ; ELSE UPDATE TRACKER
2E85 C9          RET

; SUBROUTINE TO PLAY A NOTE
2E86 E5          PNOTE:  PUSH   H
2E87 D5          PUSH   D
2E88 F5          PUSH   PSW
2E89 67          MOV    H,A

; WAIT FOR PREVIOUS PARAMETERS TO BE EATEN
2E8A 3A 4E57     PRWAIT: LDA    NEWTMR
2E8D A7          ANA    A
2E8E 20FA       JRNZ   PRWAIT ; LOOP
2E90 7C          MOV    A,H
2E91 FE63       CPI    63H ; DIVIDE?
2E93 282E       JRZ    PNOTDV
2E95 FE62       CPI    62H ; MULTIPLY?
2E97 2831       JRZ    PNOTML
2E99 FE2B       CPI    '+'
2E9B 2831       JRZ    PNOTPL
2E9D FE2D       CPI    '-'
2E9F 2830       JRZ    PNOTMN
2EA1 FE30       CPI    '0'
2EA3 2830       JRZ    PNOTZ
2EA5 21 2F04    LXI    H,NOTES-CR
2EA8 FF          HOOKIN: INDEXBIINTP%.IFE .INTP.,[RST 7]]
2EA9 5C          +.BYTE 92+0]
2EAA 3C          INR    A ; CLICK?
2EAB 280E       JRZ    PNOTCL
2EAD 3C          INR    A ; AINSWORTH NUMBA?
2EAE 2831       JRZ    PNOTNO
2EB0 3D          DCR    A
2EB1 3D          DCR    A
2EB2 32 4E59    STA    MUZTON
2EB5 3A 4EA6    LDA    DEVTEM
2EB8 32 4E57    STA    NEWTMR
2EBB AF          PNOTCL: XRA    A
2EBC 32 4E5A    PSHARP: STA    SHARPF
2EBF F1          PNOTC1: POP   PSW
2EC0 D1          POP   D
2EC1 E1          POP   H
2EC2 C9          RET
2EC3 3E8F       PNOTDV: MVI    A,0A1
2EC5 32 4E58    PNOTD1: STA    MUZMO
2EC8 18F5       JMPR   PNOTC1
2ECA 3E23       PNOTML: MVI    A,0A3
2ECC 18F7       JMPR   PNOTD1
2ECE 3E01       PNOTPL: MVI    A,1
2ED0 11          .BYTE 11H ; OPCODE TO MUNCH NEXT T
; WO BYTES INTO DE
2ED1 3E02       PNOTMN: MVI    A,2
2ED3 18E7       JMPR   PSHARP

```

.MAIN. -

```

2ED5      21 4E56      PNOTZ:  LXI      H,MUZTMR
2ED8      3A 4EA6      LDA      DEVTEM
2EDB      F3          DI
2EDC      86          ADD      M
2EDD      77          MOV      M,A
2EDE      FB          EI
2EDF      18DE       JMPR     PNOTC1
          ; PLAY AINSWORTH NOTE
2EE1      3A 4E5A      PNOTND: LDA      SHARPF ; RESTORE CHARACTER
2EE4      FF          INDEXW  1[INTP%.IFE .INTP.,[RST 7]]
2EE5      5B          +.BYTE  90+1]
2EE6      2EFD       .WORD   DICKY
2EE8      EB          XCHG
2EE9      F1          POP      PSW
2EEA      F5          PUSH     PSW
2EEB      D631       SUI      '1'
2EED      18B9       JMPR     HOOKIN
          ; SUBROUTINE TO POINT AT A TOKEN
2EEF      21 2064     TOKEPT: LXI      H,TOKTXT ; POINT AT TEXT
          LIST
2EF2      D668       SUI      68H
2EF4      C8          JOKEP1: RZ          ; QUIT IF POINTING AT EM
2EF5      CB7E       JOKEP2: BIT      7,M    ; MOVE PAST NEXT WORD
2EF7      23          INX      H
2EF8      28FB       JRZ     JOKEP2
2EFA      3D          DCR      A
2EFB      18F7       JMPR     JOKEP1 ; LOOP BACK AND CHECK
          ; MUSIC TABLES
2EFD      2F0A       DICKY:  .WORD   MUZNOR
2EFF      2F1C       .WORD   MUZMAX
2F01      2F03       .WORD   MUZMIN
2F03      MUZMIN:
2F03      64          .BYTE   B1
2F04      59          .BYTE   CS2
2F05      4F          .BYTE   DS2
2F06      4A          .BYTE   E2
2F07      42          .BYTE   FS2
2F08      3B          .BYTE   GS2
2F09      34          .BYTE   AS2
2F0A      MUZNOR:
2F0A      5E          .BYTE   C2
2F0B      54          .BYTE   D2
2F0C      4A          .BYTE   E2
2F0D      46          .BYTE   F2
2F0E      3E          .BYTE   G2
2F0F      37          .BYTE   A2
2F10      31          .BYTE   B2
          ; TONE GENERATION TABLE - ORDERED BY ASCII CHARA
          CTER
00FF      CLICK    ==      OFFH
00FE      NUMBA    ==      OFEH
2F11      FF          NOTES:  .BYTE   CLICK ; CR
          ; IRRELEVANT STUFF TO EAT TABLE SPACE
2F12      0001      TBLDIV: .WORD   1
    
```


.MAIN. -

2F4D	11	.BYTE	F4
2F4E	10	.BYTE	FS4
2F4F	0F	.BYTE	G4
2F50	0E	.BYTE	GS4
2F51	0D	.BYTE	A4
2F52	0B	.BYTE	C5
2F53	0A	.BYTE	CS5
2F54	09	.BYTE	DS5
2F55	08	.BYTE	F5
2F56	07	.BYTE	G5
2F57	06	.BYTE	A5
2F58	05	.BYTE	C6
2F59	04	.BYTE	DS6
2F5A	03	.BYTE	G6
2F5B	02	.BYTE	C7
2F5C	01	.BYTE	G7
2F5D	64	.BYTE	B1
2F5E	5E	.BYTE	C2
2F5F	59	.BYTE	CS2
2F60	54	.BYTE	D2
2F61	4F	.BYTE	DS2
2F62	4A	.BYTE	E2
2F63	46	.BYTE	F2
2F64	42	.BYTE	FS2
2F65	3E	.BYTE	G2
2F66	FF	.BYTE	CLICK
2F67	FF	.BYTE	CLICK

; TABLE OF FIRST LEVEL KEYCODES
FIRSTL1:

2F68		.BYTE	CR
2F68	0D	.BYTE	1
2F69	01	.BYTE	0
2F6A	00	.BYTE	63H
2F6B	63	.BYTE	'7'
2F6C	37	.BYTE	'8'
2F6D	38	.BYTE	'9'
2F6E	39	.BYTE	62H
2F6F	62	.BYTE	'4'
2F70	34	.BYTE	'5'
2F71	35	.BYTE	'6'
2F72	36	.BYTE	'-'
2F73	2D	.BYTE	'1'
2F74	31	.BYTE	'2'
2F75	32	.BYTE	'3'
2F76	33	.BYTE	'+'
2F77	2B	.BYTE	'/'
2F78	20	.BYTE	'0'
2F79	30	.BYTE	RUBOUT
2F7A	1F	.BYTE	'='
2F7B	3D	.BYTE	

; FIRST SHIFT KEY

2F7C		.BYTE	0A7H	; FIRST SHIFT KEY COLOR
2F7C	A7	.BYTE	CR	
2F7D	0D	.BYTE	1	
2F7E	01	.BYTE	0	
2F7F	00	.BYTE		

.MAIN. -

```

2F80      01      .BYTE      1
2F81      41      .BYTE      'A'
2F82      44      .BYTE      'D'
2F83      47      .BYTE      'G'
2F84      4A      .BYTE      'J'
2F85      4D      .BYTE      'M'
2F86      50      .BYTE      'P'
2F87      53      .BYTE      'S'
2F88      56      .BYTE      'V'
2F89      59      .BYTE      'Y'
2F8A      5F      .BYTE      5FH
2F8B      5E      .BYTE      5EH
2F8C      26      .BYTE      '&'
2F8D      24      .BYTE      '$'
2F8E      3C      .BYTE      '<'
2F8F      28      .BYTE      '('
2F90      23      .BYTE      '#'

; SECOND SHIFT KEY
KTBL2:
2F91      5F      .BYTE      05FH      ; SECOND SHIFT KEY COLOR

2F92      0D      .BYTE      CR
2F93      2F      .BYTE      2FH
2F94      00      .BYTE      0
2F95      5B      .BYTE      5BH
2F96      42      .BYTE      'B'
2F97      45      .BYTE      'E'
2F98      48      .BYTE      'H'
2F99      4B      .BYTE      'K'
2F9A      4E      .BYTE      'N'
2F9B      51      .BYTE      'Q'
2F9C      54      .BYTE      'T'
2F9D      57      .BYTE      'W'
2F9E      5A      .BYTE      'Z'
2F9F      27      .BYTE      27H
2FA0      2E      .BYTE      '.'
2FA1      40      .BYTE      '@'
2FA2      2C      .BYTE      ','
2FA3      22      .BYTE      22H
2FA4      3B      .BYTE      ';'
2FA5      25      .BYTE      '%'

; TABLE THE THIRD
KTBL3:
2FA6      0F      .BYTE      0FH      ; THIRD SHIFT KEY COLOR
2FA7      0D      .BYTE      CR
2FA8      5C      .BYTE      5CH
2FA9      00      .BYTE      0
2FAA      5D      .BYTE      5DH
2FAB      43      .BYTE      'C'
2FAC      46      .BYTE      'F'
2FAD      49      .BYTE      'I'
2FAE      4C      .BYTE      'L'
2FAF      4F      .BYTE      'O'
2FB0      52      .BYTE      'R'
2FB1      55      .BYTE      'U'

```

.MAIN. -

2FB2	58	.BYTE	'X'
2FB3	21	.BYTE	'!'
2FB4	61	.BYTE	61H
2FB5	60	.BYTE	60H
2FB6	2A	.BYTE	'*'
2FB7	3F	.BYTE	'?'
2FB8	3E	.BYTE	'>'
2FB9	29	.BYTE	'('
2FBA	3A	.BYTE	'!'

; TOKEN KEY

2FBB		KTBL4:		
2FBB	77	.BYTE	77H	; WORDS KEY COLOR
2FBC	67	.BYTE	NLLN	
2FBD	66	.BYTE	EDKEY	
2FBE	6A	.BYTE	6AH	
2FBF	68	.BYTE	68H	
2FC0	72	.BYTE	72H	
2FC1	77	.BYTE	77H	
2FC2	75	.BYTE	75H	
2FC3	6B	.BYTE	6BH	
2FC4	6F	.BYTE	6FH	
2FC5	70	.BYTE	70H	
2FC6	76	.BYTE	76H	
2FC7	6D	.BYTE	6DH	
2FC8	69	.BYTE	69H	
2FC9	6C	.BYTE	6CH	
2FCA	71	.BYTE	71H	
2FCB	6E	.BYTE	6EH	
2FCC	66	.BYTE	EDKEY	
2FCD	73	.BYTE	73H	
2FCE	01	.BYTE	1	
2FCF	74	.BYTE	74H	

; SUBROUTINE TO LDAX D FROM SCREEN TEXT MEMOR
Y IF NECESSARY

2FD0	08	LDE:	EXAF	
2FD1	CB7A		BIT	7,D
2FD3	2810		JRZ	LDE1
2FD5	D5		PUSH	D
2FD6	EB		XCHG	
2FD7	29		DAD	H
2FD8	7E		MOV	A,M
2FD9	07		RLC	
2FDA	23		INX	H
2FDB	AE		XRA	M
2FDC	E6AA		ANI	10101010B
2FDE	AE		XRA	M
2FDF	67		MOV	H,A
2FE0	08		EXAF	
2FE1	70		MOV	A,H
2FE2	EB		XCHG	
2FE3	D1		POP	D
2FE4	C9		RET	
2FE5	08	LDE1:	EXAF	
2FE6	1A		LDAX	D
2FE7	C9		RET	

.MAIN. -

2FE8 E5
 2FE9 F5
 2FEA CB7C
 2FEC 280E
 2FEE 29
 2FEF 0F
 2FF0 AE
 2FF1 E655
 2FF3 AE
 2FF4 77
 2FF5 23
 2FF6 F1
 2FF7 F5
 2FF8 AE
 2FF9 E655
 2FFB AE
 2FFC 77
 2FFD F1
 2FFE E1
 2FFF C9

```

: SUBROUTINE TO STORE MOV          M,A
STHL:  PUSH      H      11
        PUSH      PSW   11
        BIT       7,H   8
        JRZ      STHL1 13
        DAD      H      11
        RRC      4
        XRA      M      7
        ANI      01010101B 7
        XRA      M      7
        MOV      M,A   7
        INX     H      6
        POP      PSW   11
        PUSH     PSW   12
        XRA      M      7
        ANI      01010101B 7
        XRA      M      7
STHL1: MOV      M,A   7
        POP      PSW   12
        POP      H     12
        RET
        .END
    
```

92
 146
 153
 24
 177

.MAIN. -

+++++ SYMBOL TABLE +++++

A0	00E1		A1	0070		A2	0037	
A3	001B		A4	000D		A5	0006	
ABS	2892		AHOW	2A4D		ALKEYS	0214	I
ALTFON	4E66		AS0	00D4		AS1	006A	
AS2	0034		AS3	001A		ASCBC1	2202	
ASCBCD	21FE		ASORRY	297D		ATNL	214B	
AWHAT	2946		B0	00CB		B1	0064	
B2	0031		B3	0018		BBRET	240F	
BCD2	21F3		BCD3	21F6		BCD4	21FB	
BCDA1	4EB0		BCDA2	4EB9		BCDASC	21AE	
BCDMAT	217A		BEGIN	2433		BEGIN1	2439	
BEGRAM	4FCE	I	BITSPL	00A0	I	BOTRAM	A000	
BOTROM	2000		BOTSCR	4E20		BOXDR1	2259	
BOXDRW	220D		BOXDUD	2267		BOXNDR	2263	
BOXPUT	226F		BUFEND	4F3C		BUFFER	4ED4	
BUMPTR	212C		BYTEPL	0028	I	C1	00BD	
C2	005E		C3	002E		C4	0017	
C5	000B		C6	0005		C7	0002	
CBA	0009	I	CBB	0007	I	CBC	0006	I
CBD	0005	I	CBE	0004	I	CBFLAG	0008	I
CBH	000B	I	CBIXH	0003	I	CBIXL	0002	I
CBIYH	0001	I	CBIYL	0000	I	CBL	000A	I
CDELT1	23A4		CDELT2	23A9		CDELT3	23AB	
CDELTA	238C		CHDOWN	0001	I	CHGSGN	28FD	
CHKIO	2D9C		CHKIO0	2DC0		CHKIO2	2E0D	
CHKIO0	2DBB		CHKRN1	28D6		CHKRNG	28D8	
CHKSGN	28FA		CHLEFT	0002	I	CHRIGH	0003	I
CHTRIG	0004	I	CHUP	0000	I	CKHLDE	2029	
CLICK	00FF		CLRLP	2538		CLRSCR	2535	
COL0L	0004	I	COL0R	0000	I	COL1L	0005	I
COL1R	0001	I	COL2L	0006	I	COL2R	0002	I
COL3L	0007	I	COL3R	0003	I	COLBX	000B	I
COLLST	4FE8	I	COMMA	002C		COMP	202F	
CONCM	0008	I	CONPRO	4E20		COMPTR	4E21	
CR	000D		CRLF	2C7A		CS1	00B2	
CS2	0059		CS3	002C		CS4	0015	
CS5	000A		CT0	4FD5	I	CT1	4FD6	I
CT2	4FD7	I	CT3	4FD8	I	CT4	4FD9	I
CT5	4FDA	I	CT6	4FDB	I	CT7	4FDC	I
CTIMER	0203	I	CUNT	4FDD	I	CURRNT	4EC3	
CURSE	2D88		D1	00A8		D2	0054	
D3	0029		D4	0014		DEFLT	2759	
DEVCL0	4EA2		DEVCL1	4EA4		DEVLST	20A2	
DEVTEM	4EA6		DEVV1	29EA		DEVV2	29FC	
DEVV3	2A03		DEVV4	2A04		DEVVAR	4EA2	
DFTLMT	A70C		DICKY	2EFD		DIVIDE	28E5	
DOBCD0	21E8		DOCALL	2406		DS1	009F	
DS2	004F		DS3	0027		DS4	0013	
DS5	0009		DS6	0004		DURAT	4FEA	I
DV1	28F0		DV2	28F2		DVECT	22F0	
E1	0096		E2	004A		E3	0025	
E4	0012		EDFLG	4E5B		EDKEY	0066	
EDPTR	4E5C		ENDSCR	4FF4	I	ERROR	2949	
EX1	2517		EX2	2525		EX5	252D	
EXEC	2515		EXEC0	24EB		EXEC0A	2504	

.MAIN. -

+++++ SYMBOL TABLE +++++

EXECOB	2511		EXPR	2761	EXPR1	27A9	
EXPR2	27DB		EXPR3	2840	EXPRCP	22B6	
F1	008D		F2	0046	F3	0022	
F4	0011		F5	0008	FI1	2935	
FI2	299E		FILT1	2CDC	FILT2	2CDE	
FIN	292E		FINISH	2929	FIRSTC	2000	I
FIRSTL	2F68		FL1	29A0	FNDLN	2982	
FNDLP	298A		FNDNXT	299F	FNSKIP	29A1	
FNTSML	020D	I	FNTSYS	0206	FOR	262C	
FR1	2639		FR1A	263A	FR4	2644	
FR5	265D		FR6	265E	FR7	267A	
FRZGBK	2E59		FRZKEY	2E4F	FS1	0085	
FS2	0042		FS3	0020	FS4	0010	
FTBASE	0000	I	FTBYTE	0003	FTFSX	0001	I
FTFSY	0002	I	FTPTH	0006	FTPTL	0005	I
FTYSIZ	0004	I	G0	00FD	G1	007E	
G2	003E		G3	001F	G4	000F	
G5	0007		G6	0003	G7	0001	
G8	0000		GAMSTB	4FF8	GETBIT	20F8	
GETJX	28BE		GETJY	28CA	GETJY1	28D0	
GETJY2	28D2		GETJY3	28D4	GETK1	2DD0	
GETK2	2DDA		GETK3	2DE7	GETK4	2DF5	
GETK5	2DF6		GETKB	23CE	GETLN	2BD5	
GETPOT	28A6		GETTRG	28B6	GL1	2BDB	
GL2	2BDC		GL2A	2BE4	GL2B	2BFC	
GL2C	2C23		GL2D	2C2E	GL2E	2C05	
GL2F	2C0E		GL2G	2C1E	GL2J	2BF9	
GL3	2C31		GL4	2C65	GL5	2C74	
GL9	2C62		GLED	2BAE	GLEDA	2BC8	
GLEDB	2BCF		GOSUB	25F6	GOTO	2572	
GS0	00EE		GS1	0077	GS2	003B	
GS3	001D		GS4	000E	GSBEND	0007	I
GSBSCR	0001	I	GSBTIM	0000	GTMINS	4FEE	I
GTSECS	4FED	I	HOOKIN	2EAB	HORAF	000F	I
HORCB	0009	I	HOW	203B	IF1	26ED	
IFF	26EC		IGNBLK	293C	INCRO	4E5E	
INFBK	000D	I	INIDEV	201D	INIIMP	2E56	
INIT	2472		INIT0	246F	INLIN	000F	I
INMOD	000E	I	INPERR	26FB	INPUT	2705	
INTDDN	20F1		INTST	0008	IOFUN	23E8	
IP1	2705		IP10	2734	IP11	2737	
IP12	273E		IP2	270B	IP3	2710	
IP4	2722		IP5	2727	IP6	2728	
IP7	272D		IP8	272E	ITAB	2062	
JOKEP1	2EF4		JOKEP2	2EF5	JTAB	2060	
KEY0	0014	I	KEY1	0015	KEY2	0016	I
KEY3	0017	I	KEYSCN	2E5C	KEYSEX	4FE3	I
KEYTMR	4E55		KEYTRK	4E6D	KTBL1	2F7C	
KTBL2	2F91		KTBL3	2FA6	KTBL4	2FBB	
KYSCN1	2E65		KYSCN2	2E71	KYSCN3	2E7A	
KYSCN4	2E74		LDE	2FD0	LDE1	2FE5	
LDVDM1	2366		LDVDM2	2371	LDVDMC	235A	
LINED1	22C8		LINED2	22E3	LINED3	22E9	
LINED4	22ED		LINEDR	22BF	LIST	257D	
LOOPER	2DC5		LOPINC	4ECB	LOPLMT	4ECD	

.MAIN. -

+++++ SYMBOL TABLE +++++

LOPLN	4ECF		LOPPT	4ED1	LOPVAR	4EC9	
LS2	258A		LS3	258B	LS4	2593	
LS5	259A		LSQUIT	25B1	MAGIC	000C	I
MD1	2A6D		MENUST	0218	MNMX	4E5C	
MPT1	2277		MPT2	2281	MPT3	2287	
MPT4	2290		MRFLOP	0006	MRLOCK	4FF7	I
MROR	0004	I	MRRDT	0002	MRSHFT	0003	I
MRXOR	0005	I	MRXPND	0003	MUZMAX	2F1C	
MUZMIN	2F03		MUZMO	4E58	MUZNOR	2F0A	
MUZFC	4FCE	I	MUZSP	4FD0	MUZTMR	4E56	
MUZTON	4E59		MVDOWN	2A65	MVUP	2A53	
MXSCR	021E	I	NEWL1	2D81	NEWLIN	2D4A	
NEWTMR	4E57		NEXT	267F	NLLN	0067	
NLLNCT	4E62		NLLNLN	4E60	NLLNZS	4E63	
NOCLK	2E1C		NOGAME	0235	NOPLAY	0228	I
NORMEM	4000	I	NOTES	2F11	NOTF	2846	
NUMBA	00FE		NUMPLY	4FF3	NX1	2688	
NX2	26A0		NX3	26BA	NX4	26D0	
NX5	26E1		NX6	26E3	OA1	008F	
OA2	0047		OA3	0023	OA4	0011	
OA5	0008		OBO	00FE	OCO	00F1	
OD1	00D6		OE1	00BF	OF1	00B4	
OG1	00A0		OLDLN	4E64	OLDXY	4EAC	
OPOT0	4FDF	I	OPOT1	4FE0	OPOT2	4FE1	I
OPOT3	4FE2	I	OSWO	4FE4	OSW1	4FE5	I
OSW2	4FE6	I	OSW3	4FE7	OUTCH	2C7C	
PARN	286B		PARNP	286E	PARNUM	0007	
PCURS1	2D8C		PIXDUD	21E5	PIXFUN	21BE	
PIXTBL	2003		PIXVAL	4E5B	PN3	2AE8	
PN4	2AF3		PN5	2AFA	PN6	2B09	
PN7	2B0A		PN8	2B15	PN9	2B1B	
PNOTC1	2EBF		PNOTCL	2EBB	PNOTD1	2EC5	
PNOTDV	2EC3		PNOTE	2E86	PNOTML	2ECA	
PNOTMN	2ED1		PNOTNO	2EE1	PNOTPL	2ECE	
PNOTZ	2ED5		POPA	2A77	POT0	001C	I
POT1	001D	I	POT2	001E	POT3	001F	I
PP1	2A90		PR1	25BF	PR2	25C7	
PR3	25CA		PR4	25D5	PR5	25DA	
PR6	25DD		PR7	25E5	PR8	25EA	
PR9	25EE		PRINT	25B5	PRIOR	4FF9	I
PROPTR	4E20		PRTCHS	2ADE	PRTLN	2B25	
PRTNUM	2AE8		PRTSTG	2AB7	PRWAIT	2E8A	
PS1	2AB8		PS2	2AB9	PSHARP	2EBC	
PSWCY	0000	I	PSWPV	0002	PSWSGN	0007	I
PSWZRO	0006	I	PU1	2AB4	PUSHA	2A92	
PUTCD	23FC		PUTCD2	2403	PUTIO	23D9	
PUTMU	23F3		PVOLAB	4FD2	PVOLMC	4FD3	I
QH0W	2A4C		QSORRY	297C	QT1	2ACA	
QT2	2ACD		QT3	2AD6	QT5	2ADD	
QTSTG	2AC5		QWHAT	2945	R2A	23BE	
R2ACL	23AE		RANEND	2BAE	RANSHT	4FEF	I
REM	26E7		REMAIN	4EAE	RESTO	261F	
RETURN	2616		RND	2876	RSTART	2475	
RSTEXP	0002		RSTFIN	0006	RSTIGN	0004	
RSTOCH	0003		RSTPAR	0005	RUBOUT	001F	

.MAIN. -

+++++ SYMBOL TABLE +++++

RUN	254D		RUNNXL	2550		RUNSML	256C	
RUNTSL	2559		SABS	226A		SCREEN	0000	I
SCRL9	2D5A		SCRLP	2D66		SCRUP	2D70	
SCT0	0001	I	SCT1	0002	I	SCT2	0003	I
SCT3	0004	I	SCT4	0005	I	SCT5	0006	I
SCT6	0007	I	SCT7	0008	I	SEMI4S	4FDE	I
SENFLG	4FFA	I	SETV1	2918		SETVAL	2912	
SF0	0009	I	SF1	000A	I	SF2	000B	I
SF3	000C	I	SF4	000D	I	SF5	000E	I
SF6	000F	I	SF7	0010	I	SGNEXT	28AF	
SHARPF	4E5A		SIDINO	210D		SIDIN1	2116	
SIDIN3	2128		SIDINT	20F6		SIZE	2899	
SJ0	0015	I	SJ1	0017	I	SJ2	0019	I
SJ3	001B	I	SKYD	0013	I	SKYU	0012	I
SNDBX	0018	I	SNUL	0000	I	SDRRY	2040	
SP0	001C	I	SP1	001D	I	SP2	001E	I
SP3	001F	I	SSEC	0011	I	ST0	0014	I
ST1	0016	I	ST2	0018	I	ST3	001A	I
STACK	4FEF		STHL	2FE8		STHL1	2FFC	
STIMER	0200	I	STKGOS	4EC5		STKINP	4EC7	
STKLMT	4F5C		STOP	2475		STPBCT	00FC	
STRIPE	2294		STRP1	229D		STRP2	22A4	
STVDMC	2374		SW0	0010	I	SW1	0011	I
SW2	0012	I	SW3	0013	I	SYSRAM	4FCE	I
TAB2	2B38		TAB3	2B6F		TAB6	2B98	
TAPBUF	4E22		TAPCH0	2C8F		TAPCH1	2C97	
TAPCH2	2C99		TAPCH3	2CA2		TAPCH4	2CB1	
TAPEIO	0012		TAPEST	4EC2		TBFEND	4E52	
TBIEST	2410		TBIINT	20B0		TBIN0	20BC	
TBIN1	20C9		TBIN2	20E1		TBIN3	20E3	
TBLDIV	2F12		TC1	2A1D		TELL	2475	
TEXT	A000		TIMOUT	4FEC	I	TINPES	2133	
TINPUT	2151		TLIST	2155		TLIST1	2158	
TLOAD	2167		TLOAD1	216E		TMR60	4FEB	I
TN1	2A26		TOKEP	2D31		TOKEP1	2D34	
TOKEP2	2D41		TOKEPT	2EEF		TOKEQ	2C5E	
TOKER	2C4C		TOKIN	2C48		TOKJT	2046	
TOKTXT	2064		TONEA	0011	I	TONEB	0012	I
TONEC	0013	I	TONMO	0010	I	TOPSCR	4FEF	
TOUTPU	2161		TPRINT	215E		TRETUR	240C	
TSTB	29BB		TSTCH	2A10		TSTNUM	2A21	
TSTV	29AB		TSTV0	29CD		TSTVFF	21E1	
TURDC1	2C88		TURDC2	2C8C		TV1	29D1	
TXTUNF	4E52		UCURSE	2D46		UMARGT	4FFB	I
URINAL	0FFF	I	USERTB	4FFD	I	VARBGN	4E6E	
VARNXT	4EC7		VBBLNK	0006	I	VBCCHK	0004	I
VBCH	0003	I	VBCL	0002	I	VBCLAT	0003	I
VBCLMT	0000	I	VBCREV	0001	I	VBDCH	0001	I
VBDCL	0000	I	VBDXH	0004	I	VBDXL	0003	I
VBDYH	0009	I	VBDYL	0008	I	VBMR	0000	I
VBOAH	000E	I	VBOAL	000D	I	VBSACT	0007	I
VBSTAT	0001	I	VBTIMB	0002	I	VBXCHK	0007	I
VBXH	0006	I	VBXL	0005	I	VBYCHK	000C	I
VBYH	000B	I	VBYL	000A	I	VDM	2CCE	
VDMOCR	2D00		VDM1	2CE5		VDM2	2CF8	

.MAIN. -

+++++ SYMBOL TABLE +++++

VDM3	2D0E		VDMCAL	2C8C		VDMDN1	2D29	
VDMDN2	2D2D		VDMDOM	2D2C		VDMFRG	2CB7	
VDMNLF	4E54		VDMTOK	2CC5		VDMX	4EAE	
VDMY	4BAA		VECT1	230B		VECT2	2316	
VECT2A	2335		VECT3	2345		VECT4	234A	
VECT5	2355		VECT9	2330		VERAF	000E	I
VERBL	000A	I	VIBRA	0014	I	VOICES	4FD4	I
VOLAB	0016	I	VOLC	0015	I	VOLN	0017	I
WASTER	0FFF	I	WCLICK	2E29		WHAT	2035	
WHATSU	2E3F		XP11	27B1		XP12	27B4	
XP13	27B7		XR14	27BE		XP15	27CF	
XP16	27D2		XP21	27DE		XP22	27FA	
XP23	2801		XP24	280A		XP25	2833	
XP32	2859		XPAND	0019	I	XPR0	2873	
XPR1	276B		XPR2	2771		XPR3	2777	
XPR4	277E		XPR5	2786		XPR6	278C	
XPR7	2792		XPR8	2794		XPR9	2872	
XXST1	247E		XXST2	2487		XXST3	24C3	
XXST4	2496		ZRONK	2129		\$IND	007E	
.BLNK.	0000	*G3 X	.DATA.	0000*	X	.INTP:	0000	
.PROC.	0000	X						