

```

    .PREL
    .IDENT MAIN
    .INSERT HVG.LIB
    .INSERT MACRO.LIB
    .INSERT MUSIC.LIB
    .INSERT BBEQU.ASM
4E20 .LOC BOTSCR
; TAPE INPUT GOODIES
4E20 CONPRO ==: . ; TAG TO NAIL BOTH
4E20 PROPTR:: .BLKB 1 ; PRODUCERS POIN
    TER
4E21 CONPTR:: .BLKB 1 ; CONSUMERS POIN
    TER
4E22 TAPBUF:: .BLKB 48 ; ROTATING BUFFE
    R
4E52 TBFEND ==: . ; 1 PAST END OF BUFFER
4E52 TXTUNF:: .BLKB 2
4E54 VDMNLF:: .BLKB 1 ; VDM NEW LINE F
    LAG
4E55 KEYTMR:: .BLKB 1 ; KEYBOARD SCAN
    TIMER
4E56 MUZTMR:: .BLKB 1 ; MUSIC NOTE TIM
    ER
4E57 NEWTMR:: .BLKB 1 ; NEW MUSIC TIME
    R VALUE
4E58 MUZMO:: .BLKB 1 ; MASTER OSCILLATOR
4E59 MUZTON:: .BLKB 1 ; TONE VALUE
4E5A SHARPF:: .BLKB 1 ; SHARP-FLAT FLA
    G
;
4E5B EDFLG::
4E5B PIXVAL:: .BLKB 1 ; PIXEL TO DRAW
    VECTOR WITH
4E5C EDPTR::
4E5C MNMX:: .BLKB 2 ; MIN - MAX DELTAS FOR V
    ECTOR DRAW
4E5E INCRO:: .BLKB 2 ; COORDINATE INCREMENTS
    FOR VECTOR DRAW
4E60 NLLNLN:: .BLKB 2 ; LINE NUMBER BE
    ING TYPED
4E62 NLLNCT:: .BLKB 1 ; AUTO LINE # FL
    AG-COUNTER
4E63 NLLNZS:: .BLKB 1 ; AUTO LINE NUMB
    ER ZERO SURPRESS FLAG
4E64 OLDLN:: .BLKB 2 ; PREVIOUS LINE # TYPED
;
4E66 ALTFON:: .BLKB 7 ; ALTERNATE FONT
    DESCRIPTOR
4E6D KEYTRK:: .BLKB 1 ; KEYBOARD TRACK
    ER
4E6E VARBGN:: .BLKB 2*26
4EA2 DEVVAR:: ; DEVICE VARIABLES::
4EA2 DEVCL0:: .BLKB 2 ; BACKGROUND COL
    OR
4EA4 DEVCL1:: .BLKB 2 ; FOREGROUND COL
    OR
DEVCL2:: .BLKB 2 ; COLOR ?

```

MAIN -

```

OR
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 DIVIDE
4EB0 BCDA1:: .BLKB 9 ; FIRST ARG AREA
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 2024 .MAIN:: JMP BEGIN ; ** AUTOSTART CASSETTE
**
2003 00000000 .BYTE 0,0,0,0 ; FILLER TO LINE UP REST
ARTS
; TRANSFER VECTORS TO RESTART ROUTINES
2007 C3 0000:04 JMP TSTCH# ; * RST 8
200A C3 0000:05 JMP EXPR# ; * RST 16
200D C3 0000:06 JMP OUTCH# ; * RST 24
2010 C3 0000:07 JMP IGNBLK# ; * RST 32
2013 C3 0000:08 JMP PARN# ; * RST 40
2016 F1 POP PSW ; * RST 48
2017 C3 0000:09 JMP FINISH#
201A C3 0000:0A JMP CHKIO# ; ** LINK TO TAPE READ R
OUTLINE **
201D 000000 .BYTE 0,0,0 ; ALIGN TO WORD BOUNDARY
2020 20AF JTAB: .WORD SIDINT
2022 2069 ITAB: .WORD TBIINT
; ** TINY BASIC EXECUTION STARTS HERE **
; CLEAR WHOLE KIT AND KABOODLE
2024 AF BEGIN: XRA A
2025 D30C OUT MAGIC
2027 67 MOV H,A
2028 6F MOV L,A
2029 47 MOV B,A

```

MAIN -

```

202A 70          BEGIN1: MOV     M,B
202B 23          INX     H
202C 7C          MOV     A,H
202D FE50        CPI     50H
202F 20F9        JRNZ   BEGIN1
2031 31 4FCE     LXI     SP,sysram
2034 FF          ENTER[RST 7
2035 00          +.BYTE 0
0001          +.INTP.=1]
                SETOUT 1[INTP%[.IFE .INTP.,[RST 7]]
2036 17          +.BYTE 22+1]
2037 B0          .BYTE 176
2038 2C          .BYTE 44
2039 08          .BYTE 8
                EMUSIC 1      ; SHUT DOWN MUSIC[INTP%
                .IFE .INTP.,[RST 7]]
203A 15          +.BYTE 20+1 ]
                SETB 1[INTP%[.IFE .INTP.,[RST 7]]
203B 7B          +.BYTE 122+1]
203C 47          .BYTE 0A2      ; SET INITIAL MASTER OSC
                ILATOR
203D 4E58        .WORD  MUZMO
                ; INITIALIZE DEVICE VARIABLES
                MOVE 1[INTP%[.IFE .INTP.,[RST 7]]
203F 5F          +.BYTE 94+1]
2040 4EA2        .WORD  DEVVAR
2042 000A        .WORD  10
2044 0000:0B     .WORD  INIDEV#
                MOVE 1[INTP%[.IFE .INTP.,[RST 7]]
2046 5F          +.BYTE 94+1]
2047 4E66        .WORD  ALTFON
2049 0007        .WORD  7
204B 0206        .WORD  fntsys
                SETW 1[INTP%[.IFE .INTP.,[RST 7]]
204D 7D          +.BYTE 124+1]
204E 06A0        .WORD  6A0H
2050 4E66        .WORD  ALTFON
                SETW 1[INTP%[.IFE .INTP.,[RST 7]]
2052 7D          +.BYTE 124+1]
2053 A004        .WORD  TEXT+4
2055 4E52        .WORD  TXTUNF
2057 02          EXIT[.BYTE 2
0000          +.INTP.=0]
2058 21 A001     LXI     H,TEXT+1
205B 3EFF        MVI     A,OFFH
205D CD 0000:0C  CALL   STHL#
2060 CD 20ED     INIT0:: CALL  TBIEST
2063 CD 0000:0D  INIT::  CALL  CRLF#
2066 C3 0000:0E  JMP    TELL#
                ; TINY BASIC INTERRUPT ROUTINE
2069 F5          TBIINT: PUSH  PSW      ; SAVE REGISTERS
206A C5          PUSH  B
206B D5          PUSH  D
206C E5          PUSH  H
                ; DEAL WITH KEYBOARD SCAN TIMER

```

```

206D 21 4E55          LXI      H,KEYTMR
2070 7E              MOV      A,M
2071 A7              ANA      A
2072 2801           JRZ      TBINO
2074 35              DCR      M
2075 23              TBINO: INX      H
; HAS MUSIC TIMER COUNTED DOWN?
2076 7E              MOV      A,M
2077 A7              ANA      A
2078 2808           JRZ      TBIN1 ; YEP - PLAY NEXT NOTE
207A 35              DCR      M ; ELSE DECREMENT IT
207B 201F           JRNZ    TBIN3 ; JUMP IF NOT NOW ZERO
207D AF              XRA      A
207E D311           OUT     TONEA ; ELSE SILENCE
2080 1818           JMPR    TBIN2
; MUSIC TIMER IS AT ZERO - ARE NEW PARAMETERS RE
ADY?
2082 23              TBIN1: INX      H ; STEP TO NEW TIMER VALU
E
2083 B6              ORA      M ; IS IT NON ZERO?
2084 2816           JRZ      TBIN3 ; JUMP IF NOT
2086 2B              DCX      H ; ELSE SET OFFICIAL TIME
R
2087 77              MOV      M,A
2088 23              INX      H
2089 3600           MVI     M,0 ; ZERO NEW TIMER VALUE A
S FLAG
208B 23              INX      H
208C 7E              MOV      A,M ; GET NEW M.O.
208D D310           OUT     TONMO
208F 3647           MVI     M,0A2 ; RESET DEFAULT MASTER 0
SC
2091 23              INX      H
2092 7E              MOV      A,M ; AND NEW TONE
2093 D311           OUT     TONEA
2095 A7              ANA      A ; REST WANTED?
2096 2804           JRZ      TBIN3 ; YES - JUMP AROUND VOLU
ME UPDATE
2098 3E0F           MVI     A,15
209A D316           TBIN2: OUT     VOLAB
; SET COLOR REGISTERS TO VALUES IN PARAMETER VAR
S %0 AND %1
209C 3A 4EA2       TBIN3: LDA     DEVCL0
209F D304           OUT     COL0L
20A1 D305           LDA     DEVCL1
20A3 3A 4EA4       OUT     COL1L
20A6 D306           LDA     DEVCL2
20A8 D307           OUT     COL2L
20AA E1              INTDON: POP    H
20AB D1              POP     D
20AC C1              POP     B
20AD 1830           JMPR    ZRONK
; SERIAL INPUT DRIVER INTERRUPT ROUTINE
; THIS ROUTINE SAMPLES THE SERIAL INPUT BIT

```

*OUTPUT
 ALL
 FOUR*

MAIN -

```

; 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
20AF      F5      SIDINT: PUSH    PSW
20B0      D9      EXX
; SAMPLE THE INPUT BIT
20B1      DB12    GETBIT: IN      TAPEIO
20B3      1F      RAR
20B4      79      MOV      A,C      ; SHIFT INTO ACCUMULATOR
;
20B5      1F      RAR
20B6      4F      MOV      C,A
20B7      78      MOV      A,B      ; WHAT STATE ARE WE IN?
20B8      A7      ANA      A
20B9      FA 20C6 JM      SIDINO  ; 1 COUNTING STATE?
20BC      2011    JRNZ     SIDINI  ; JUMP IF IN MIDDLE OF C
HAR
; WE ARE WAITING FOR A START BIT
20BE      CB79    BIT      7,C      ; DID WE GET ONE?
20C0      201C    JRNZ     SIDIN3  ; JUMP IF RIGHT
20C2      0608    MVI      B,8      ; SET WAITING FOR 8 BITS
;
20C4      1818    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
20C6      04      SIDINO: INR    B      ; ASSUME WE GOT IT
20C7      CB79    BIT      7,C
20C9      2013    JRNZ     SIDIN3  ; JUMP IF GOOD ASSUMPTIO
N
; NO - RESET COUNTER AND WAIT SOME MORE
20CB      06FC    MVI      B,STPBCT
20CD      180F    JMPR     SIDIN3
; IN THE MIDDLE OF A CHARACTER...
; WAS THIS THE LAST BIT?
20CF      100D    SIDINI: DJNZ   SIDIN3  ; JUMP IF NOT
; WELCOME TO LAST BITSVILLE
20D1      2A 4E20 LHLD   CONPRO  ; GET POINTERS
20D4      7D      MOV      A,L
20D5      CD 20E2 CALL   BUMPTR  ; A=P+1
20D8      BC      CMP      H      ; IS C = P+1?
20D9      2803    JRZ      SIDIN3  ; YEP - FULL - IGNORE
20DB      264E    MVI      H,TAPBUF>8

```

MAIN -

```

20DD 71          MOV      M,C
20DE D9          SIDIN3: EXX
20DF F1          ZRONK: POP      PSW
20E0 FB          EI
20E1 C9          RET
; SUBROUTINE TO INCREMENT 1 BYTE POINTER TO CIRC
ULAR BUFFER
20E2 3C          BUMPTR:: INR      A      ; BUMP IT
20E3 FE52       CPI      TBFEND&OFFH
20E5 C0          RNZ          ; QUIT IF NOT AT END
20E6 3E22       MVI      A,TAPBUF&OFFH
20E8 C9          RET
; COMMAND TO RESTORE TINY BASIC INTERRUPT ROUTIN
E
20E9 CD 20ED    TRETUR:: CALL   TBIEST
20EC F7          BBRET:  RST      RSTFIN  ; GO HOME
20ED DB12       TBIEST:: IN      TAPEIO  ; INITIALIZE TAP
E INTERFACE
20EF E602       ANI      2
20F1 20FA       JRNZ     TBIEST  ; WAIT FOR THANG TO GO T
O ZERO
20F3 32 4EC2    STA      TAPEST
20F6 3E03       MVI      A,3
20F8 32 4EA6    STA      DEVTEM
; ACTIVATE TINY BASIC INTERRUPT ROUTINE
20FB F3          DI
20FC ED5E       IM2
20FE 3E20       MVI      A,ITAB>8
2100 ED47       STAI
2102 3E08       MVI      A,8
2104 D30E       OUT     INMOD
2106 3EC8       MVI      A,200
2108 D30F       OUT     INLIN
210A 3E22       MVI      A,ITAB&OFFH
210C D30D       OUT     INFBK
210E FB          EI
210F C9          RET
; SUBROUTINE TO ESTABLISH TAPE UNIT AS INPUT DEV
ICE
2110 F3          TINPES:: DI
2111 21 2222     LXI      H,((TAPBUF&OFFH)<8)+(TAPBUF&OFFH
)
2114 22 4E20     SHLD   CONPRO
2117 D9          EXX
2118 AF          XRA      A
2119 32 4EA6     STA      DEVTEM
211C 3C          INR      A
211D 32 4EC2     STA      TAPEST
2120 06FC       MVI      B,STPBCT
2122 D9          EXX
2123 3E18       MVI      A,18H
2125 D30E       OUT     INMOD
2127 FB          EI
2128 C9          RET
.END

```

LINE -

```

                                .PREL
                                .IDENT LINE
                                .INSERT HVG.LIB
                                .INSERT MACRO.LIB
                                .INSERT BBEQU.ASM
0000'                                .RELOC
                                ; DIRECT COMMAND - TEXT COLLECTOR
0000' TELL::
                                ;
                                ; LXI D,MSG
                                ; CALL PRTSTG
0000' STOP::
0000' 31 0000:04 RSTART:: LXI SP,STACK#
0003' 21 000A' LXI H,XXST1+1
0006' 22 0000:05 SHLD CURRNT#
0009' 21 0000 XXST1: LXI H,0
000C' 22 0000:06 SHLD LOPVAR#
000F' 22 0000:07 SHLD STKGOS#
0012' 3E3E XXST2: MVI A,'>'
0014' CD 00C7' CALL GETLN
0017' D5 PUSH D
0018' 11 0000:08 LXI D,BUFFER#
                                ; IGNORE ANY LEADING '>'
001B' 1A LDAX D
001C' FE3E CPI '>'
001E' 2001 JRNZ XXST4
0020' 13 INX D
0021' CD 0000:09 XXST4: CALL TSTNUM#
0024' E7 RST RSTIGN
0025' 7C MOV A,H
0026' B5 ORA L
0027' C1 POP B
0028' CA 0000:0A JZ EXECO#
002B' 22 0000:0B SHLD OLDLN#
002E' 1B DCX D
002F' 7C MOV A,H
0030' 12 STAX D
0031' 1B DCX D
0032' 7D MOV A,L
0033' 12 STAX D
0034' C5 PUSH B
0035' D5 PUSH D
0036' 79 MOV A,C
0037' 93 SUB E
0038' F5 PUSH PSW
0039' CD 0077' CALL FNDLN
003C' D5 PUSH D
003D' 2010 JRNZ XXST3
003F' D5 PUSH D
0040' CD 0094' CALL FNDNXT
0043' C1 POP B
0044' 2A 0000:0C LHLD TXTUNF#
0047' CD 0177' CALL MVUP
004A' 60 MOV H,B
004B' 69 MOV L,C
004C' 22 0000:0C SHLD TXTUNF#

```

FNDLN

LHLD	TXTUNF#
CALL	MVUP
MOV	H,B
MOV	L,C
SHLD	TXTUNF#

LINE -

```

004F' C1          XXST3: POP      B
0050' 2A 0000:0C LHL D  TXTUNF#
0053' F1          POP      PSW
0054' E5          PUSH     H
0055' FE03        CPI      3
0057' 28A7        JRZ     RSTART
0059' 85          ADD     L
005A' 5F          MOV     E,A
005B' 3E00        MVI    A,0
005D' 8C          ADC     H
005E' 57          MOV     D,A
005F' 21 A70C     LXI    H,DFTLMT
0062' EB          XCHG
0063' CD 0000:0D CALL   COMP#
0064' D2 0000:0E JNC    QSORRY#
0069' 22 0000:0C SHLD   TXTUNF#
006C' D1          POP     D
006D' CD 0189'    CALL   MVDOWN
0070' D1          POP     D
0071' E1          POP     H
0072' CD 0177'    CALL   MVUP
0075' 189B        JMPR   XXST2

; FNDLN
0077' 7C          FNDLN:: MOV    A,H
0078' B7          ORA    A
0079' FA 0000:0F JM     QHOW#
007C' 11 0000:10 LXI    D,TEXT#
007F' 13          FNDLP:: INX   D
0080' CD 019A'    CALL   LDE
0083' 4F          MOV    C,A
0084' 1B          DCX   D
0085' 87          ADD   A
0086' D8          RC
0087' CD 019A'    CALL   LDE
008A' 95          SUB   L
008B' 47          MOV   B,A
008C' 13          INX   D
008D' 79          MOV   A,C
008E' 9C          SBB   H
008F' 3804        JRC   FL1
0091' 1B          DCX   D
0092' B0          ORA   B
0093' C9          FI2:  RET
0094' 13          FNDNXT: INX  D
0095' 13          FL1:  INX  D
0096' CD 019A'    FNDSKP:: CALL LDE
0099' FE0D        CPI   CR
009B' 20F8        JRNZ  FL1
009D' 13          INX  D
009E' 18DF        JMPR  FNDLP
00A0' 3A 0000:11 GLED:  LDA   EDFLG#
00A3' A7          ANA   A
00A4' 2814        JRZ   GLEDA
00A6' 11 0000:08 LXI   D,BUFFER#
00A9' CD 0000:09 CALL  TSTNUM#
    
```

SPECIFIED LN

TXJUNE

LINE -

```

00AC'   CD 0077'           CALL   FNDLN
00AF'   3E3F             MVI    A,'?'
00B1'   C0              RNZ
00B2'   13              INX    D
00B3'   CD 00C1'       CALL   GLEDB
00B6'   AF              XRA    A
00B7'   32 0000:11     STA    EDFLG#
00BA'   ED5B 0000:12 GLEDA: LDED   EDPTR#
00BE'   CD 019A'       CALL   LDE
00C1'   13              GLEDB: INX    D
00C2'   ED53 0000:12   SDED   EDPTR#
00C6'   C9              RET
00C7'   11 0000:08     GETLN:: LXI   D,BUFFER#
00CA'   32 0000:11     STA    EDFLG#
00CD'   DF              GL1:   RST   RSTOCH  ; PROMPT OR ECHO
00CE'   C5              GL2:   PUSH  B
00CF'   E5              PUSH  H
00D0'   D5              PUSH  D
                        ; PLACE UP CURSOR BLOCK
00D1'   0EAA           MVI    C,OAAH
00D3'   CD 0000:13     CALL   CURSE#
                        ; RETURN CHAR FROM NEXT LINE #
00D6'   21 0000:14     GL2A: LXI   H,NLLNCT#
00D9'   7E              MOV   A,M      ; SENSE FLAG
00DA'   A7              ANA   A
00DB'   2838           JRZ   GL2C
00DD'   35              DCR   M
                        ; FIRST TIME THRU?
00DE'   FE05           CPI    5
00E0'   200C           JRNZ  GL2B      ; JUMP IF NOT
                        ; GET PREVIOUS LINE # AND BUMP IT
00E2'   2A 0000:0B     LHLD  OLDLN#
00E5'   11 000A       LXI   D,10
00E8'   19              DAD   D
00E9'   CBBC           RES   7,H      ; ALLOW NEG
00EB'   22 0000:15     GL2J: SHLD  NLLNLN ; MOVE TO WORKING RAM CE
                        LL#
                        ; COMPUTE DIVISION SUBTRACTOR
00EE'   FF              GL2B:  INDEXW 1[INTP%.IFE .INTP.,[RST 7]]
00EF'   5B              +.BYTE 90+1]
00F0'   01C8'         .WORD  TBLDIV-2
00F2'   2A 0000:15     LHLD  NLLNLN#
00F5'   0600           MVI   B,0
00F7'   A7              GL2E:  ANA   A
00F8'   ED52           DSBC  D
00FA'   FA 0100'       JM    GL2F
00FD'   04              INR   B
00FE'   18F7           JMPR  GL2E
0100'   19              GL2F:  DAD   D
0101'   22 0000:15     SHLD  NLLNLN#
0104'   21 0000:16     LXI   H,NLLNZS#
0107'   78              MOV   A,B
0108'   A7              ANA   A
0109'   2005           JRNZ  GL2G
010B'   7E              MOV   A,M

```

LINE -

```

010C'  A7          ANA      A
010D'  28C7       JRZ      GL2A   ; YES - JUMP BACK
010F'  AF         XRA      A
0110'  C630       GL2G:  ADI      '0'   ; MAKE ASCII
0112'  77         MOV      M,A     ; SET NONZERO FLAG
0113'  180B       JMPR     GL2D
                ; NOTHIN FANCY
0115'  CD 0000:17 GL2C:  CALL     CHKIO# ; GET NORMAL CHARACTER#
0118'  D1         POP      D
0119'  12         STAX     D       ; STUFF CHAR AS DELIMITE
                R
011A'  D5         PUSH     D
011B'  FE66       CPI      EDKEY
011D'  CC 00A0'   CZ       GLED
0120'  D1         GL2D:  POP      D
0121'  E1         POP      H
0122'  C1         POP      B
0123'  12         GL3:   STAX     D
0124'  FE1F       CPI      RUBOUT
0126'  2033       JRNZ     GL4
0128'  7B         MOV      A,E
0129'  E5         PUSH     H
012A'  21 0000:08 LXI      H,BUFFER#
012D'  BD         CMP      L
012E'  E1         POP      H
012F'  289D       JRZ      GL2
0131'  1B         DCX      D
0132'  1A         LDAX    D
0133'  FE68       CPI      68H    ; TOKEN TO RUB OUT?
0135'  3007       JRNC    TOKIN
0137'  CD 0000:18 CALL     PNOTE#
013A'  3E1F       MVI      A,RUBOUT
013C'  188F       JMPR     GL1
013E'  D5         TOKIN:  PUSH     D
013F'  CD 0000:19 CALL     TOKEPT#
0142'  7E         TOKER:  MOV      A,M
0143'  E5         PUSH     H
0144'  E67F       ANI      7FH
0146'  CD 0000:18 CALL     PNOTE#
0149'  3E1F       MVI      A,RUBOUT
014B'  CD 0000:1A CALL     VDM#
014E'  E1         POP      H
014F'  7E         MOV      A,M
0150'  23         INX      H
0151'  07         RLC
0152'  30EE       JRNC    TOKER
0154'  3E1F       TOKEQ:  MVI      A,RUBOUT
0156'  DF         RST      RSTOCH ; ECHO ONE RUBOUT CHAR
0157'  D1         POP      D
0158'  C3 00CE'   GL9:   JMP      GL2
015B'  FE0D       GL4:   CPI      CR
015D'  CA 016E'   JZ      GL5
0160'  7B         MOV      A,E
0161'  E5         PUSH     H
0162'  21 0000:1B LXI      H,BUFEND#

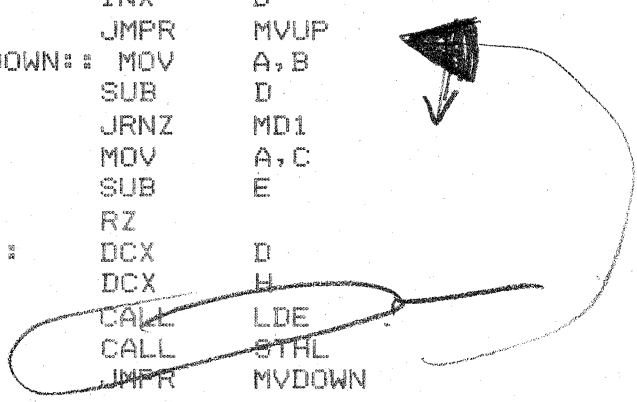
```

LINE -

```

0165'   BD          CMP      L
0166'   E1          POP      H
0167'   28EF       JRZ      GL9
0169'   1A          LDAX    D
016A'   13          INX     D
016B'   C3 00CD'   JMP     GL1
016E'   13          GL5:   INX     D
016F'   13          INX     D
0170'   3EFF       MVI     A,0FFH
0172'   12          STAX    D
0173'   1B          DCX     D
0174'   C3 0000:1C JMP     CRLF#
; MVUP AND MVDOWN
0177'   CD 0000:0D MVUP:  CALL    COMP#
017A'   C8          RZ
017B'   CD 019A'   CALL    LDE
017E'   E5          PUSH   H
017F'   60          MOV     H,B
0180'   69          MOV     L,C
0181'   CD 01B2'   CALL    STHL
0184'   E1          POP     H
0185'   13          INX     D
0186'   03          INX     B
0187'   18EE       JMPR   MVUP
0189'   78          MVDOWN:: MOV    A,B
018A'   92          SUB     D
018B'   2003       JRNZ   MD1
018D'   79          MOV     A,C
018E'   93          SUB     E
018F'   C8          RZ
0190'   1B          MD1:  DCX     D
0191'   2B          DCX     H
0192'   CD 019A'   CALL    LDE
0195'   CD 01B2'   CALL    STHL
0198'   18EF       JMPR   MVDOWN
019A'   08          LDE::  EXAF
019B'   CB7A       BIT     7,D
019D'   2810       JRZ     LDE1
019F'   D5          PUSH   D
01A0'   EB          XCHG
01A1'   29          DAD     H
01A2'   7E          MOV     A,M
01A3'   07          RLC
01A4'   23          INX     H
01A5'   AE          XRA     M
01A6'   E6AA       ANI     10101010B
01A8'   AE          XRA     M
01A9'   67          MOV     H,A
01AA'   08          EXAF
01AB'   7C          MOV     A,H
01AC'   EB          XCHG
01AD'   D1          POP     D
01AE'   C9          RET
01AF'   08          LDE1:  EXAF
01B0'   1A          LDAX    D

```



LINE -

```

01B1'   C9                RET
                        ; SUBROUTINE TO STORE A IN (HL)
01B2'   E5                STHL::  PUSH   H
01B3'   F5                PUSH   PSW
01B4'   CB7C              BIT     7,H
01B6'   280E              JRZ    STHL1
01B8'   29                DAD    H
01B9'   0F                RRC
01BA'   AE                XRA    M
01BB'   E655              ANI    01010101B
01BD'   AE                XRA    M
01BE'   77                MOV    M,A
01BF'   23                INX    H
01C0'   F1                POP    PSW
01C1'   F5                PUSH   PSW
01C2'   AE                XRA    M
01C3'   E655              ANI    01010101B
01C5'   AE                XRA    M
01C6'   77                STHL1:  MOV    M,A
01C7'   F1                POP    PSW
01C8'   E1                POP    H
01C9'   C9                RET
01CA'   0001              TBLDIV: .WORD 1
01CC'   000A              .WORD 10
01CE'   0064              .WORD 100
01D0'   03E8              .WORD 1000
01D2'   2710              .WORD 10000
                        .END

```


EXEC -

```

        .PREL
        .IDENT EXEC
        .INSERT MACRO.LIB
        .INSERT BBEQU.ASM
; TABLE GIVING JUMP TO ADDRESS FOR COMMANDS
0000' 0000:04 TOKJT: .WORD LIST#
0002' 0000:05 .WORD CLRSCR#
0004' 00&F' .WORD RUN
0006' 0000:06 .WORD NEXT#
0008' 0000:07 .WORD LINEDR#
000A' 0000:08 .WORD IFF#
000C' 0000:09 .WORD GOTO#
000E' 0000:0A .WORD GOSUB#
0010' 0000:0B .WORD RETURN#
0012' 0000:0C .WORD BOXDRW#
0014' 0000:0D .WORD FOR#
0016' 0000:0E .WORD INPUT#
0018' 0000:0F .WORD PRINT#

; DIRECT AND EXEC
001A' E7 EXECO:: RST RSTIGN ; GET FIRST NONBLANK

001B' D5 PUSH D
001C' FE68 CPI 68H ; IS SHE A TOKEN?
001E' 3813 JRC EXEC0A ; NO
0020' FE75 CPI 75H
0022' 300F JRNC EXEC0A ; NO

; WE FOUND A TOKEN - LOOKUP IN TABLE AND JUMP TO
IT
0024' 07 RLC
0025' 5F MOV E,A
0026' 1600 MVI D,0
0028' 21 FF30' LXI H,TOKJT-208
002B' 19 DAD D
002C' 5E MOV E,M
002D' 23 INX H
002E' 56 MOV D,M
002F' EB XCHG
0030' D1 POP D
0031' 13 INX D
0032' E9 PCHL

; NOT A TOKEN - A VARIABLE PERHAPS?
0033' CD 0000:10 EXEC0A: CALL TSTV#
0036' 3808 JRC EXEC0B ; NO - SEARCH 1
0038' CF TSTC '=' ,EXEC0B ; MAYBE!

RST 1
0039' 3D + .BYTE '='
003A' 05 + .BYTE EXEC0B --1
+]

003B' C1 POP B ; THROW OUT OLD PTR
003C' CD 0000:11 CALL SETV1# ; ASSIGNMENT 1
003F' F7 RST RSTFIN
0040' D1 EXEC0B: POP D
0041' 21 00A7' LXI H,TAB2
0044' E7 EXEC:: RST RSTIGN
0045' D5 EXO: PUSH D

```

```

0046'  CD 0000:12  EX1:  CALL  LDE#
0049'  CB7E        BIT    7,M
004B'  2016        JRNZ  ATEND
004D'  BE          CMP    M
004E'  2004        JRNZ  MISMAT
0050'  23          INX   H
0051'  13          INX   D
0052'  18F2       JMPR  EX1
0054'  23          MISMAT: INX  H
0055'  CB7E        BIT    7,M
0057'  28FB       JRZ   MISMAT
0059'  23          SKPBAD: INX H
005A'  23          INX  H
005B'  23          INX  H
005C'  7E         MOV   A,M
005D'  3C         INR   A
005E'  2808       JRZ   EOL
0060'  D1         POP   D
0061'  18E2       JMPR  EXO
0063'  F680       ATEND:  ORI   80H
0065'  BE         CMP   M
0066'  20F1       JRNZ  SKPBAD
0068'  23          EOL:   INX  H
0069'  7E         MOV   A,M
006A'  23          INX  H
006B'  66         MOV   H,M
006C'  6F         MOV   L,A
006D'  F1         POP   PSW
006E'  E9         PCHL
006F'  11 0000:13  RUN:   LXI   D,TEXT#
0072'  21 0000    RUNNXL: LXI   H,0
0075'  CD 0000:14  CALL  FNDLF#
0078'  DA 0000:15  JC    RSTART#
007B'  ED53 0000:16  RUNTSL: SDIED  CURRNT#
007F'  CD 0000:17  CALL  WHATSU# ; CHECK FOR HALT
0082'  2008        JRNZ  ..OK ; AND FOR TRACE KEY
0084'  D5          PUSH  D ; TRACE IS WANTED
0085'  CD 0000:18  CALL  PRTLN#
0088'  CD 0000:19  CALL  PRTSTG#
008B'  D1          POP   D
008C'  ..OK:
008C'  13          INX   D
008D'  13          INX   D
008E'  CD 0000:17  RUNSML: CALL  WHATSU# ; CHECK FOR INTERRUPT KE
Y
0091'  C3 001A'    JMP   EXECO
; END A LINE ROUTINE
0094'  CD 009A'    FINISH: CALL  FIN
0097'  C3 0000:1A  JMP   QWHAT#
009A'  CF          FIN:   TSTCC 3BH,FI10 RST 1
009B'  3B          +     .BYTE 3BH
009C'  03          +     .BYTE FI1--1
+ ]
009D'  F1          POP   PSW
009E'  18EE       JMPR  RUNSML

```

EXEC -

00A0	CF	FI1:	TSTCC	CR,FI2[RST	1	
00A1	0D	+	.BYTE	CR		
00A2	03	+	.BYTE	FI2-,-1		
		+]]				
00A3	F1		POP	PSW		
00A4	18CC		JMPR	RUNNXL		
00A6	C9	FI2:	RET			
00A7	54D6	TAB2:	ITEM	'TV',PUTCD#[.ASCIS	'TV'
00A9	0000:1B	+	.WORD	PUTCD#		
		+]]				
00AB	4DD5		ITEM	'MU',PUTMU#[.ASCIS	'MU'
00AD	0000:1C	+	.WORD	PUTMU#		
		+]]				
00AF	A6		ITEM	'&',PUTIO#[.ASCIS	'&'
00B0	0000:1D	+	.WORD	PUTIO#		
		+]]				
00B2	43414CCC		ITEM	'CALL',DOCALL	#[.ASCIS
			'CALL'			
00B6	0000:1E	+	.WORD	DOCALL #		
		+]]				
00B8	AE		ITEM	'.',REM#[.ASCIS	'.'
00B9	0000:1F	+	.WORD	REM#		
		+]]				
00BB	A4		ITEM	'\$',BCDMAT#[.ASCIS	'\$'
00BC	0000:20	+	.WORD	BCDMAT#		
		+]]				
00BE	3A		.BYTE	'!'		
00BF	E8		TOKEN	68H,TLIST#[.BYTE	68H+80H
00C0	0000:21	+	.WORD	TLIST#		
		+]]				
00C2	3A		.BYTE	'!'		
00C3	F4		TOKEN	74H,TOUTPU#[.BYTE	74H+80H
00C4	0000:22	+	.WORD	TOUTPU#		
		+]]				
00C6	3A		.BYTE	'!'		
00C7	F3		TOKEN	73H,TINPUT#[.BYTE	73H+80H
00C8	0000:23	+	.WORD	TINPUT#		
		+]]				
00CA	3A		.BYTE	'!'		
00CB	F0		TOKEN	70H,TRETUR#[.BYTE	70H+80H
00CC	0000:24	+	.WORD	TRETUR#		
		+]]				
00CE	3A		.BYTE	'!'		
00CF	EA		TOKEN	6AH,TLOAD#[.BYTE	6AH+80H
00D0	0000:25	+	.WORD	TLOAD#		
		+]]				
00D2	2A		.BYTE	'*'		
00D3	F4		TOKEN	74H,TPRINT#[.BYTE	74H+80H
00D4	0000:26	+	.WORD	TPRINT#		
		+]]				
00D6	53544FD0		ITEM	'STOP',STOP#[.ASCIS	'STOP'
		/				
00DA	0000:27	+	.WORD	STOP#		
		+]]				
00DC	FF		TOKEN	07FH,DEFLT#[.BYTE	07FH+80H

```
00DD' 0000:28 + .WORD DEFLT#  
+]  
.END
```

EXPR -

```

.PREL
.IDENT EXPR
.INSERT BBEQU.ASM

; ** EXPR **
0000' CD 0048' EXPR:: CALL EXPR1
0003' E5          PUSH  H
0004' 21 013F'   LXI   H,TAB6
0007' C3 0000:04 JMP   EXEC#
000A' CD 0033'   XPR1: CALL XPR8
000D' D8          RC
000E' 6F          MOV   L,A
000F' C9          RET
0010' CD 0033'   XPR2: CALL XPR8
0013' C8          RZ
0014' 6F          MOV   L,A
0015' C9          RET
0016' CD 0033'   XPR3: CALL XPR8
0019' C8          RZ
001A' D8          RC
001B' 6F          MOV   L,A
001C' C9          RET
001D' CD 0033'   XPR4: CALL XPR8
0020' 6F          MOV   L,A
0021' C8          RZ
0022' D8          RC
0023' 6C          MOV   L,H
0024' C9          RET
0025' CD 0033'   XPR5: CALL XPR8
0028' C0          RNZ
0029' 6F          MOV   L,A
002A' C9          RET
002B' CD 0033'   XPR6: CALL XPR8
002E' D0          RNC
002F' 6F          MOV   L,A
0030' C9          RET
0031' E1          XPR7: POP  H
0032' C9          RET
0033' 79          XPR8: MOV  A,C
0034' E1          POP  H
0035' C1          POP  B
0036' E5          PUSH H
0037' C5          PUSH B
0038' 4F          MOV  C,A
0039' CD 0048'   CALL  EXPR1
003C' EB          XCHG
003D' E3          XTHL
003E' CD 0000:05 CALL  CKHLDE#
0041' D1          POP  D
0042' 21 0000   LXI  H,0
0045' 3E01      MVI  A,1
0047' C9          RET
0048' CF          XPR1: TSTC  /-/,XP11[ RST  1
0049' 2D          +      .BYTE /-/'
004A' 05          +      .BYTE XP11-,-1
+ ]
    
```

2B18

*BUMPED BY 1
Leaving EXEC*

2B11

EXPR -

```

004B' 21 0000          LXI      H,0
004E' 1821            JMPR      XP16
0050' CF              XP11:  TSTC      '+',XP12[   RST      1
0051' 2B              +      .BYTE    '+'
0052' 00              +      .BYTE    XP12--,-1
                    +]
0053' CD 007A'        XP12:  CALL      EXPR2
0056' CF              XP13:  TSTC      '+',XP15[   RST      1
0057' 2B              +      .BYTE    '+'
0058' 15              +      .BYTE    XP15--,-1
                    +]
0059' E5              PUSH     H
005A' CD 007A'        CALL      EXPR2
005D' EB              XP14:  XCHG
005E' E3              XTHL
005F' 7C              MOV      A,H
0060' AA              XRA      D
0061' 7A              MOV      A,D
0062' 19              DAD      D
0063' D1              POP      D
0064' FA 0056'        JM       XP13
0067' AC              XRA      H
0068' F2 0056'        JP       XP13
006B' C3 0000:06      JMP      QHOW#
006E' CF              XP15:  TSTC      '- ',XPR9[   RST      1
006F' 2D              +      .BYTE    '- '
0070' A0              +      .BYTE    XPR9--,-1
                    +]
0071' E5              XP16:  PUSH     H
0072' CD 007A'        CALL      EXPR2
0075' CD 0000:07      CALL      CHGSGN#
0078' 18E3            JMPR      XP14
007A' CD 00DF'        EXPR2:  CALL      EXPR3
007D' CF              XP21:  TSTCC    62H,XP24[   RST      1
007E' 62              +      .BYTE    62H
007F' 29              +      .BYTE    XP24--,-1
                    +]
0080' E5              PUSH     H
0081' CD 00DF'        CALL      EXPR3
0084' 0600            MVI      B,0
0086' CD 0000:08      CALL      CHKSGN#
0089' E3              XTHL
008A' CD 0000:08      CALL      CHKSGN#
008D' EB              XCHG
008E' E3              XTHL
008F' 7C              MOV      A,H
0090' B7              ORA      A
0091' 2806            JRZ      XP22
0093' 7A              MOV      A,D
0094' B2              ORA      D
0095' EB              XCHG
0096' C2 0000:09      JNZ      AHOW#
0099' 7D              XP22:  MOV      A,L
009A' 21 0000          LXI      H,0
009D' B7              ORA      A

```

EXPR -

009E'	2832		JRZ	XP25		
00A0'	19	XP23:	DAD	D		
00A1'	DA 0000:09		JC	AHOW#		
00A4'	3D		DCR	A		
00A5'	20F9		JRNZ	XP23		
00A7'	1829		JMPR	XP25		
00A9'	CF	XP24:	TSTCC	63H, XPR9[RST	1
00AA'	63	+	.BYTE	63H		
00AB'	65	+	.BYTE	XPR9-, -1		
		+]				
00AC'	E5		PUSH	H		
00AD'	CD 00DF'		CALL	EXPR3		
00B0'	0600		MVI	B, 0		
00B2'	CD 0000:08		CALL	CHKSGN#		
00B5'	E3		XTHL			
00B6'	CD 0000:08		CALL	CHKSGN#		
00B9'	EB		XCHG			
00BA'	E3		XTHL			
00BB'	EB		XCHG			
00BC'	7A		MOV	A, D		
00BD'	B3		ORA	E		
00BE'	CA 0000:09		JZ	AHOW#		
00C1'	C5		PUSH	B		
00C2'	CD 0000:0A		CALL	DIVIDE#		
00C5'	D1		POP	D	;	SIGN STUFF TO DE
00C6'	C5		PUSH	B	;	SAVE DIVIDE RESULT
00C7'	CB7A		BIT	7, D	;	WAS SIGN SET?
00C9'	C4 0000:07		CNZ	CHGSGN#	;	YEP - CHANGE#
00CC'	22 0000:0B		SHLD	REMAIN#	;	STUFF IT
00CF'	E1		POP	H	;	RESULT TO HL
00D0'	42		MOV	B, D	;	COPY OVER SIGN STUFF
00D1'	4B		MOV	C, E		
00D2'	D1	XP25:	POP	D		
00D3'	7C		MOV	A, H		
00D4'	B7		ORA	A		
00D5'	FA 0000:06		JM	AHOW#		
00D8'	78		MOV	A, B		
00D9'	B7		ORA	A		
00DA'	FC 0000:07		CM	CHGSGN#		
00DD'	189E		JMPR	XP21		
00DF'	21 0115'	EXPR3:	LXI	H, TAB3		
00E2'	C3 0000:04		JMP	EXEC#		
00E5'	CD 0000:0C	NOTF:	CALL	TSTV#		
00E8'	380E		JRC	XP32		
00EA'	EB		XCHG			
00EB'	CD 0000:0D		CALL	LDE#		
00EE'	F5		PUSH	PSW		
00EF'	13		INX	D		
00F0'	CD 0000:0D		CALL	LDE#		
00F3'	EB		XCHG			
00F4'	67		MOV	H, A		
00F5'	F1		POP	PSW		
00F6'	6F		MOV	L, A		
00F7'	C9		RET			
00F8'	CD 0000:0E	XP32:	CALL	TSTNUM#		

2B11
2B E5
FL

EXPR -

```

00FB' 78          MOV     A,B
00FC' B7          ORA     A
00FD' C0          RNZ
; SINGLE CHAR STRING CONSTANT?
00FE' CF          TSTC   "'",PARN      ; HAVE WE GOT QU
      OTES?L RST     1
00FF' 22          +     .BYTE  "'"
0100' 09          +     .BYTE  PARN     -,-1
      +]
0101' CD 0000:0D CALL   LDE#
0104' 6F          MOV     L,A      ; FAILED TSTNUM SET H TO
      ZERO
0105' 13          INX     D
0106' CF          TSTC   "'",XPRO      ; ERROR IF NO TR
      AILINGL RST     1
0107' 22          +     .BYTE  "'"
0108' 09          +     .BYTE  XPRO     -,-1
      +]
0109' C9          RET
010A' CF          PARN:: TSTC   '(',XPROL   RST     1
010B' 28          +     .BYTE  '('
010C' 05          +     .BYTE  XPRO-,-1
      +]
010D' D7          PARNP: RST   RSTEXP
010E' CF          TSTC   ')',XPROL   RST     1
010F' 29          +     .BYTE  ')'
0110' 01          +     .BYTE  XPRO-,-1
      +]
0111' C9          XPR9:  RET
0112' C3 0000:0F XPR0:  JMP     QWHAT#
; FUNCTION AND COMPARISON OPERATION LOOKUP TABLE
S
0115' F6          TAB3:  TOKEN   76H,RND#L   .BYTE   76H+80H
0116' 0000:10     +     .WORD   RND#
      +]
0118' 4BCE        ITEM   'KN',GETPOT#L .ASCIS   'KN'
011A' 0000:11     +     .WORD   GETPOT#
      +]
011C' 54D2        ITEM   'TR',GETTRG#L .ASCIS   'TR'
011E' 0000:12     +     .WORD   GETTRG#
      +]
0120' 4AD8        ITEM   'JX',GETJX#L .ASCIS   'JX'
0122' 0000:13     +     .WORD   GETJX#
      +]
0124' 4AD9        ITEM   'JY',GETJY#L .ASCIS   'JY'
0126' 0000:14     +     .WORD   GETJY#
      +]
0128' 4BD0        ITEM   'KP',GETKB#L .ASCIS   'KP'
012A' 0000:15     +     .WORD   GETKB#
      +]
012C' 50D8        ITEM   'PX',PIXFUN#L .ASCIS   'PX'
012E' 0000:16     +     .WORD   PIXFUN#
      +]
0130' A6          ITEM   '&',IOFUN#L .ASCIS   '&'
0131' 0000:17     +     .WORD   IOFUN#

```



```

0133' 4142D3      +]      ITEM      'ABS',ABS#I      .ASCIS      'ABS'
0136' 0000:18    +      .WORD      ABS#
                   +]
0138' 53DA      ITEM      'SZ',SIZE#I      .ASCIS      'SZ'
013A' 0000:19    +      .WORD      SIZE#
                   +]
013C' FF        TOKEN      7FH,NOTFI      .BYTE      7FH+80H
013D' 00E5'      +      .WORD      NOTF
                   +]
013F' 3EBD      TAB6:  ITEM      '>=',XPR1I      .ASCIS      '>='
0141' 000A'      +      .WORD      XPR1
                   +]
0143' A3        ITEM      '#',XPR2I      .ASCIS      '#'
0144' 0010'      +      .WORD      XPR2
                   +]
0146' BE        ITEM      '>',XPR3I      .ASCIS      '>'
0147' 0016'      +      .WORD      XPR3
                   +]
0149' BD        ITEM      '= ',XPR5I      .ASCIS      '= '
014A' 0025'      +      .WORD      XPR5
                   +]
014C' 3CBD      ITEM      '<=',XPR4I      .ASCIS      '<='
014E' 001D'      +      .WORD      XPR4
                   +]
0150' BC        ITEM      '<',XPR6I      .ASCIS      '<'
0151' 002B'      +      .WORD      XPR6
                   +]
0153' FF        TOKEN      7FH,XPR7I      .BYTE      7FH+80H
0154' 0031'      +      .WORD      XPR7
                   +]
                                .END

```

VARS -

```

                                .PREL
                                .IDENT VARS
                                .INSERT HVG.LIB
                                .INSERT MACRO.LIB
                                .INSERT BBEQU.ASM
0000'                                .RELOC
; DEVICE VARIABLE TABLE
; THIS TABLE IS IN INVERSE ORDER OF APPEARANCE I
N MEMORY
0007'                                PARNUM ==          7          ; 7 GUYS
0000'                                0007' INIDEV:: .WORD 7
0002'                                0000' .WORD 0
0004'                                0003' .WORD 3
0006'                                FFB3' .WORD -77
0008'                                0028' .WORD 40
000A'                                0000' .WORD 0
000C'                                DEVLST:
000C'                                12' .BYTE 'R'-'@'
000D'                                4D' .BYTE 'M'
000E'                                18' .BYTE 'X'-'@'
000F'                                59' .BYTE 'Y'
0010'                                03' .BYTE 'C'-'@'
0011'                                59' .BYTE 'Y'
0012'                                03' .BYTE 'C'-'@'
0013'                                58' .BYTE 'X'
0014'                                0E' .BYTE 'N'-'@'
0015'                                54' .BYTE 'T'
0016'                                06' .BYTE 'F'-'@'
0017'                                43' .BYTE 'C'
0018'                                02' .BYTE 'B'-'@'
0019'                                43' .BYTE 'C'
; DEVICE VARIABLE TO OUTPUT TO REFERENCED IO POR
T
001A'                                EF' PUTIO:: RST RSTPAR ; GET PORT #
001B'                                CF' TSTC '=@,PUTIO2 ; GET EQUALSE
RST 1
001C'                                3D' + .BYTE '=@'
001D'                                0B' + .BYTE PUTIO2 --,-1
+ ]
001E'                                E5' PUSH H ; SAVE PORT #
001F'                                D7' RST RSTEXP ; EVALUATE EXPRESSION FO
LLOWING
0020'                                7D' MOV A,L ; A=VALUE TO OUTPUT
0021'                                E1' POP H ; RESTORE PORT #
0022'                                C5' PUSH B
0023'                                44' MOV B,H
0024'                                4D' MOV C,L
0025'                                ED79' OUTP A ; IT 1
0027'                                C1' POP B
0028'                                F7' BBRET: RST RSTFIN ; GO HOME
0029'                                C3 0000:04' PUTIO2: JMP QWHAT#
; FUNCTION TO RETURN VALUE OF A GIVEN IO PORT
002C'                                EF' IOFUN:: RST RSTPAR ; GET PORT NUMBA
002D'                                C5' PUSH B
002E'                                44' MOV B,H

```

VARS -

```

002F' 4D          MOV     C,L
0030' ED78       INP     A
0032' 6F         MOV     L,A
0033' 2600       MVI     H,0
0035' C1         POP     B
0036' C9         RET

; ROUTINE TO TRANSFER CONTROL TO ASSEMBLY LANGUAGE SUBROUTINE
0037' 21 0028'   DOCALL:: LXI     H,BBRET ; PUSH RETURN ADDR ON STACK
003A' E5         PUSH    H
003B' D7         RST     RSTEXP ; GET ADDRESS
003C' E9         PCHL    ; AND JUMP TO IT
; FUNCTION TO SENSE DIAL VALUE
003D' 3E1B       GETPOT:: MVI     A,1BH
003F' CD 006B'   CALL    CHKRNG ; GET DATA
0042' 2F         CMA
0043' D680       SUI     80H
0045' 6F         MOV     L,A
0046' C3 0000:05 JMP     SGNEXT#
; FUNCTION TO SENSE STATE OF TRIGGER
0049' CD 0069'   GETTRG:: CALL    CHKRN1
004C' E610       ANI     10H
004E' C8         RZ
004F' 2C         INR     L
0050' C9         RET
; FUNCTIONS TO RETURN JOYSTICK VALUE
; THESE FUNCTIONS RETURN EITHER +1, ,0 OR -1, DEPENDING
; ON JOYSTICK STATE
0051' CD 0069'   GETJX:: CALL    CHKRN1 ; FARM IN RANGE?
0054' 0F         RRC
0055' 0F         RRC
0056' 0F         RRC
0057' 380E       JRC     GETJY3
0059' 0F         RRC
005A' 3807       JRC     GETJY1
005C' C9         RET
; ENTRY FOR Y JOYSTICK VALUE
005D' CD 0069'   GETJY:: CALL    CHKRN1
0060' 0F         RRC
0061' 3002       JRNC   GETJY2
0063' 23         GETJY1: INX     H
0064' C9         RET
0065' 0F         GETJY2: RRC
0066' D0         RNC
0067' 2B         GETJY3: DCX    H
0068' C9         RET
; SUBROUTINE TO GET PARAMETER BETWEEN 1 AND 4
0069' 3E0F       CHKRN1: MVI     A,0FH
006B' C5         CHKRNG: PUSH    B
006C' F5         PUSH    PSW
006D' EF         RST     RSTPAR
006E' F1         POP     PSW
006F' 85         ADD     L

```

VARS -

```

0070' 4F          MOV      C,A
0071' ED78       INP      A
0073' C1         POP      B
0074' 21 0000    LXI      H,0
0077' C9         RET

; SETVAL, FIN, AND ERROR
0078' CD 008E'   SETVAL:: CALL  TSTVFF
007B' CF         TSTC    '=' ,SVWHATE      RST      1
007C' 3D         +      .BYTE  '='
007D' 15         +      .BYTE  SVWHAT--1
+ ]

2303 007E' E5     SETV1:: PUSH   H
007F' D7         RST    RSTEXP
0080' 44         MOV    B,H
0081' 4D         MOV    C,L
0082' E1         POP    H
0083' F5         PUSH   PSW
0084' 79         MOV    A,C
0085' CD 0000:06 CALL  STHL#
0088' 23         INX    H
0089' 78         MOV    A,B
008A' CD 0000:06 CALL  STHL#
008D' F1         POP    PSW

; SUBROUTINE TO GET A VARIABLE MAKING SURE IT IS
ONE
008E' CD 0096'   TSTVFF:: CALL  TSTV
0091' D0         RNC
0092' C9         RET
0093' C3 0000:04 SVWHAT: JMP    QWHAT#
0096' E7         TSTV:: RST    RSTIGN
0097' FE25       CPI    '%'      ; PEEK-POKE?
0099' 281E       JRZ    TSTVO
009B' D640       SUI    '@'
009D' D8         RC
009E' 201D       JRNZ   TV1
00A0' 13         INX    D
00A1' EF         RST    RSTPAR
00A2' 29         DAD    H
00A3' DA 0000:07 JC    QHOW#
00A6' D5         TSTB:  PUSH   D
00A7' EB         XCHG
00A8' CD 0000:08 CALL  SIZE#
00AB' CD 0000:09 CALL  COMP#
00AE' DA 0000:0A JC    ASORRY#
00B1' 2A 0000:0B LHLD  TXTUNF#
00B4' 2B         DCX    H
00B5' 2B         DCX    H
00B6' 19         DAD    D
00B7' D1         POP    D
00B8' C9         RET

; %(ADDR) PEEK-POKE CALL
00B9' 13         TSTVO: INX    D
00BA' EF         RST    RSTPAR ; GET ADDR
00BB' AF         XRA    A      ; CLEAR CY
00BC' C9         RET      ; AND GO BACK

```

VARS -

```

00BD' FE1B      TV1:   CPI      27
00BF' 3F        CMC
00C0' D8        RC
00C1' 13        INX      D
                ; IS SECOND CHARACTER ALSO ALPHA?
00C2' 6F        MOV      L,A      ; SAVE FIRST      ONE
00C3' CD 0000:0C CALL    LDE#
00C6' FE41      CPI      'A'
00C8' 3826      JRC      DEVV4   ; IF NOT IN RANGE A-Z
00CA' FE5B      CPI      'Z'+1
00CC' 3022      JRNC    DEVV4   ; THEN SEARCH
00CE' C5        PUSH    B
00CF' D5        PUSH    D
00D0' 67        MOV      H,A      ; SECOND CHAR TO H
00D1' 0607      MVI      B,PARNUM   ; B - ITERATION

                CTR
00D3' 11 000C' LXI      D,DEVLST   ; DE - SEARCH TA
                BLE
00D6' 1A        DEVV1: LDAX   D      ; GET FIRST      ENTRY
00D7' 13        INX      D
00D8' BD        CMP      L
00D9' 1A        LDAX   D
00DA' 13        INX      D
00DB' 200B      JRNZ    DEVV2
00DD' BC        CMP      H
00DE' 2008      JRNZ    DEVV2
                ; MATCH FOUND - FIGURE OUT LOOKUP INDEX
00E0' 78        MOV      A,B
00E1' C61A      ADI      26
00E3' 6F        MOV      L,A
00E4' D1        POP      D
00E5' 13        INX      D      ; BUMP CHAR PTR
00E6' 1807      JMPR    DEVV3
                ; MISMATCH - LOOP BACK IF POSS
00E8' 10EC      DEVV2: DJNZ    DEVV1
                ; NOT POSSIBLE - RETURN NOT A VAR
00EA' D1        POP      D
00EB' C1        POP      B
00EC' 1B        DCX      D      ; BACKUP TO CHAR START
00ED' 37        STC      ; SET CARRY
00EE' C9        RET
00EF' C1        DEVV3: POP      B
00F0' 7D        DEVV4: MOV      A,L
00F1' 21 FFFE:0D LXI      H,VARBGN#-2
00F4' 07        RLC
00F5' 85        ADD      L
00F6' 6F        MOV      L,A
00F7' 3E00      MVI      A,0
00F9' 8C        ADC      H
00FA' 67        MOV      H,A
00FB' C9        RET
                .END

```

MATH -

```

                                .PREL
                                .IDENT MATH
                                .INSERT HVG.LIB
                                .INSERT MACRO.LIB
                                .INSERT BBEGU.ASM
0000'                                .RELOC
                                ; BCD MATH ROUTINE INTERFACE
                                ; IMPLEMENTS $+, $-, $*, */
0000'    E7                                BCDMAT:: RST      RSTIGN
0001'    13                                INX      D
0002'    F5                                PUSH     PSW      ; SAVE CODE SCANNED
0003'    CD 0000:04                        CALL     TSTVFF# ; GET FIRST   VAR
0006'    D5                                PUSH     D
0007'    11 0000:05                        LXI     D,BCDA1# ; CONVERT TO BCD

000A'    CD 005D'                          CALL     ASCBCD
000D'    D1                                POP      D
000E'    CF                                TSTCC   COMMA,BCDDUD ; INSIST ON COMM
                                ASE
000F'    2C                                +        .BYTE   COMMA
0010'    33                                +        .BYTE   BCDDUD  -,-1
                                +]

0011'    CD 0000:04                        CALL     TSTVFF# ; GET 2ND VAR
0014'    F1                                POP      PSW
0015'    D5                                PUSH     D
0016'    11 0000:06                        LXI     D,BCDA2# ; SAME STORY
0019'    F5                                PUSH     PSW
001A'    CD 005D'                          CALL     ASCBCD
001D'    11 0000:05                        LXI     D,BCDA1#
0020'    F1                                POP      PSW
0021'    0609                              MVI     B,9
0023'    CD 0047'                          CALL     DOBCDD
0026'    D1                                POP      D ; SCAN FOR PLACE
                                TO STORE
0027'    CF                                TSTCC   COMMA,BCDDUDE RST      1
0028'    2C                                +        .BYTE   COMMA
0029'    1A                                +        .BYTE   BCDDUD-,-1
                                +]

002A'    CD 0000:04                        CALL     TSTVFF#
002D'    D5                                PUSH     D
002E'    11 0000:05                        LXI     D,BCDA1#
                                ; ARG1 CONTAINS THE RESULT - STICK IT LAST VAR
0031'    01 1200                            LXI     B,1200H
0034'    EB                                BCDASC: XCHG
0035'    FF                                INDEXN[INTP%[.IFE .INTP.,[RST 7]]
0036'    56                                +.BYTE  86+0]
0037'    C630                              ADI     '0'
0039'    EB                                XCHG
003A'    CD 0000:07                        CALL     STHL#
003D'    23                                INX     H
003E'    23                                INX     H
003F'    0C                                INR     C
0040'    10F2                              DJNZ   BCDASC
0042'    D1                                POP     D
0043'    F7                                RST     RSTFIN

```

MATH -

```

0044'   C3 0000:08   BCDDUD: JMP      QWHAT#
; SUBROUTINE TO IT      1
0047'   FE62       DOBCDO: CPI      62H      ; < MULT?
0049'   300A       JRNC      BCD3      ; NO
004B'   FE2D       CPI      '-'      ; YES IS IT MINUS?
004D'   2803       JRZ      BCD2      ; JUMP IF SO
004F'   FF         BCDADD  ; NO - ITS ADD THEN[INTP%[.IFE .
; INTP.,[RST 7]]
0050'   62         +.BYTE  98+0]
0051'   C9         RET
0052'   FF         BCD2:  BCDSUB  ; SUBTR 1[INTP%[.IFE .INTP.,[RST
; 7]]
0053'   64         +.BYTE  100+0]
0054'   C9         RET
0055'   2003       BCD3:  JRNZ      BCD4      ; JUMP IF NOT = TIMES
0057'   FF         BCDMUL  ; ELSE MULT[INTP%[.IFE .INTP.,[R
; ST 7]]
0058'   66         +.BYTE  102+0]
0059'   C9         RET
005A'   FF         BCD4:  BCDDIV[INTP%[.IFE .INTP.,[RST 7]]
005B'   68         +.BYTE  104+0]
005C'   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
005D'   EB         ASCBCD: XCHG
005E'   01 1200    LXI      B,1200H
0061'   CD 0000:09  ASCBC1: CALL    LDE#
0064'   FF         STOREN[INTP%[.IFE .INTP.,[RST 7]]
0065'   58         +.BYTE  88+0]
0066'   0C         INR      C
0067'   13         INX      D
0068'   13         INX      D
0069'   10F6      DJNZ    ASCBC1
006B'   C9         RET
; RANDOM NUMBER FUNCTION
006C'   EF         RND::  RST      RSTPAR
006D'   7C         MOV      A,H
006E'   B7         ORA      A
006F'   FA 0000:0A  JM      QHOW#
0072'   B5         ORA      L
0073'   CA 0000:0A  JZ      QHOW#
0076'   D5         PUSH    D
0077'   EB         XCHG      ; DE = RANGE
0078'   AF         XRA      A
0079'   FF         RANGED[INTP%[.IFE .INTP.,[RST 7]]
007A'   76         +.BYTE  118+0]
007B'   6F         MOV      L,A
007C'   AF         XRA      A
007D'   FF         RANGED[INTP%[.IFE .INTP.,[RST 7]]
007E'   76         +.BYTE  118+0]
007F'   67         MOV      H,A
; HL = RANDOM #
0080'   C5         PUSH    B

```

MATH -

```

0081'   CD 00A3'           CALL    DIVIDE
0084'   C1                POP     B
0085'   D1                POP     D
0086'   23                INX     H
0087'   C9                RET

; ABSOLUTE VALUE FUNCTION
0088'   EF               ABS::  RST    RSTPAR
0089'   1B                DCX     D
008A'   CD 00B8'         CALL    CHKSGN
008D'   13                INX     D
008E'   C9                RET

; SIZE FUNCTION - RETURNS BYTES NOT YET USED
008F'   2A 0000:0B      SIZE:: LHL D  TXTUNF#
0092'   D5                PUSH   D
0093'   EB                XCHG
0094'   21 A70C          LXI     H, DFTLMT
0097'   A7                ANA     A
0098'   ED52            DSBC    D
009A'   D1                POP     D
009B'   C9                RET

; SIGN EXTEND SUBROUTINE
009C'   2600            SGNEXT:: MVI   H, 0
009E'   7D                MOV    A, L
009F'   A7                ANA     A
00A0'   F0                RP
00A1'   25                DCR    H
00A2'   C9                RET

; DIVIDE, SUBDE, CHKSGN, CHGSGN, CKHLDE
00A3'   E5               DIVIDE:: PUSH  H
00A4'   6C                MOV    L, H
00A5'   2600            MVI   H, 0
00A7'   CD 00AE'         CALL    DV1
00AA'   41                MOV    B, C
00AB'   7D                MOV    A, L
00AC'   E1                POP     H
00AD'   67                MOV    H, A
00AE'   0EFF            DV1:  MVI   C, -1
00B0'   0C               DV2:  INR   C
00B1'   A7                ANA     A
00B2'   ED52            DSBC    D
00B4'   30FA            JRNC   DV2
00B6'   19                DAD    D
00B7'   C9                RET
00B8'   7C               CHKSGN:: MOV   A, H
00B9'   B7                ORA    A
00BA'   F0                RP
00BB'   7C               CHGSGN:: MOV   A, H
00BC'   B5                ORA    L
00BD'   C8                RZ
00BE'   7C                MOV    A, H
00BF'   F5                PUSH   PSW
00C0'   2F                CMA
00C1'   67                MOV    H, A
00C2'   7D                MOV    A, L
00C3'   2F                CMA

```


MATH -

```
00C4' 6F          MOV     L,A
00C5' 23          INX     H
00C6' F1          POP     PSW
00C7' AC          XRA     H
00C8' F2 0000:0A  JP      @HLOW#
00CB' 78          MOV     A,B
00CC' EE80        XRI     80H
00CE' 47          MOV     B,A
00CF' 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
;          .END
```

STDBAS -

```

        .PREL
        .IDENT STDBAS
        .INSERT HVG.LIB
        .INSERT MACRO.LIB
        .INSERT BBEQU.ASM
0000'   .RELOC
0000'   D7          GOTO:: RST   RSTEXP
0001'   D5          PUSH   D
0002'   CD 0000:04 CALL   FNDLN#
0005'   C2 0000:05 JNZ   AHOW#
0008'   F1          POP    PSW
0009'   C3 0000:06 JMP    RUNTSL#
; LIST AND PRINT
; NEW - IMPROVED LIST COMMAND
; LETS YOU PUT IT IN A PROGRAM
000C'   21 0000    LIST:: LXI   H,0      ; ASSUME AT EOL
000F'   E7          RST    RSTIGN   ; IGNORE
0010'   CD 0000:07 CALL   ATNL#
0013'   2805      JRZ    LS3
0015'   FE2C      CPI    ', '      ; LEADING COMMA?
0017'   2801      JRZ    LS3      ; YEP - SKIP FIRST
        EXPR GET
; NOT AT END - GET FIRST          EXPR
0019'   D7          LS2:   RST    RSTEXP
001A'   E5          LS3:   PUSH   H
001B'   21 FFFF      LXI    H,0FFFFH
001E'   CF          TSTCC  COMMA,LS4[ RST    1
001F'   2C          +     .BYTE COMMA
0020'   01          +     .BYTE LS4-.-1
        +]
0021'   D7          RST    RSTEXP
0022'   D5          LS4:   PUSH   D
0023'   FDE1      POP    Y
0025'   E3          XTHL
0026'   CD 0000:04 CALL   FNDLN#
0029'   3815      LS5:   JRC    LSQUIT
002B'   E3          XTHL
002C'   7C          MOV    A,H
002D'   B5          ORA    L
002E'   2810      JRZ    LSQUIT
0030'   2B          DCX   H
0031'   E3          XTHL
0032'   CD 0000:08 CALL   PRTLN#
0035'   CD 0000:09 CALL   PRTSTG#
0038'   CD 0000:0A CALL   WHATSU#
003B'   CD 0000:0B CALL   FNDLP#
003E'   18E9      JMPR  LS5
0040'   FDE5      LSQUIT: PUSH   Y
0042'   D1          POP    D
0043'   F7          RST    RSTFIN
0044'   0E08      PRINT:: MVI   C,8
0046'   CF          TSTCC  59,PR1[ RST    1
0047'   3B          +     .BYTE 59
0048'   06          +     .BYTE PR1-.-1
        +]

```

STDBAS -

```

0049/  CD 0000:0C          CALL    CRLF#
004C/  C3 0000:0D          JMP     RUNSML#
004F/  CF                PR1:  TSTCC   CR,PR6[RST    1
0050/  0D                +     .BYTE   CR
0051/  1C                +     .BYTE   PR6--1
                +]

0052/  CD 0000:0C          CALL    CRLF#
0055/  C3 0000:0E          JMP     RUNNXL#
0058/  CF                PR2:  TSTC   '#',PR4[RST    1
0059/  23                +     .BYTE   '#
005A/  0B                +     .BYTE   PR4--1
                +]

005B/  D7                PR3:  RST     RSTEXP
005C/  3E00             MVI    A,0C0H
005E/  A5                ANA    L
005F/  B4                ORA    H
0060/  C2 0000:0F          JNZ    QHOW#
0063/  4D                MOV    C,L
0064/  1805             JMPR   PR5
0066/  CD 0000:26        PR4:  CALL   QTSTG
0069/  1814             JMPR   PR9
006B/  CF                PR5:  TSTCC  COMMA,PR8[RST    1
006C/  2C                +     .BYTE   COMMA
006D/  0D                +     .BYTE   PR8--1
                +]

006E/  CF                PR6:  TSTCC  COMMA,PR7[RST    1
006F/  2C                +     .BYTE   COMMA
0070/  05                +     .BYTE   PR7--1
                +]

0071/  3E20             MVI    A,' '
0073/  DF                RST    RSTOCH
0074/  18F8             JMPR   PR6
0076/  CD 0000:10        PR7:  CALL   FIN#
0079/  18DD             JMPR   PR2
007B/  CD 0000:0C        PR8:  CALL   CRLF#
007E/  F7                RST    RSTFIN
007F/  D7                PR9:  RST    RSTEXP
0080/  C5                PUSH   B
0081/  CD 0000:11        CALL   PRTNUM#
0084/  C1                POP    B
0085/  18E4             JMPR   PR5

                ; GOSUB AND RETURN
0087/  CD 020D/         GOSUB: CALL  PUSHA
008A/  D7                RST    RSTEXP
008B/  D5                PUSH   D
008C/  CD 0000:04        CALL   FNDLN#
008F/  C2 0000:05        JNZ    AHOW#
0092/  2A 0000:12        LHLD  CURRNT#
0095/  E5                PUSH   H
0096/  2A 0000:13        LHLD  STKGOS#
0099/  E5                PUSH   H
009A/  21 0000           LXI    H,0
009D/  22 0000:14        SHLD  LOPVAR#
00A0/  39                DAD   SP
00A1/  22 0000:13        SHLD  STKGOS#

```

STDBAS -

```

00A4'  C3 0000:06          JMP      RUNTSL#
00A7'  2A 0000:13  RETURN: LHL  STKGOS#
00AA'  7C          MOV      A,H
00AB'  B5          ORA      L
00AC'  CA 0000:15          JZ       QWHAT#
00AF'  F9          SPHL
00B0'  E1          RESTO: POP      H
00B1'  22 0000:13          SHLD    STKGOS#
00B4'  E1          POP      H
00B5'  22 0000:12          SHLD    CURRNT#
00B8'  D1          POP      D
00B9'  CD 01F2'        CALL    POPA
00BC'  F7          RST     RSTFIN
          ; FOR AND NEXT
00BD'  CD 020D'        FOR::  CALL    PUSHA
00C0'  CD 0000:16          CALL    SETVAL#
00C3'  2B          DCX     H
00C4'  22 0000:14          SHLD    LOPVAR#
00C7'  CF          TSTCC   77H,FR1A      ; TO?E RST
          1
00C8'  77          +      .BYTE  77H
00C9'  01          +      .BYTE  FR1A      --1
          +1
00CA'  D7          FR1:   RST     RSTEXP
00CB'  22 0000:17          FR1A:  SHLD    LOPLMT#
00CE'  21 0001          LXI     H,1
00D1'  CF          TSTCC   75H,FR4 ; STEP?E RST
          1
00D2'  75          +      .BYTE  75H
00D3'  01          +      .BYTE  FR4      --1
          +1
00D4'  D7          RST     RSTEXP
00D5'  22 0000:18          FR4:   SHLD    LOPINC #
00D8'  2A 0000:12          LHL    CURRNT#
00DB'  22 0000:19          SHLD    LOPLN#
00DE'  EB          XCHG
00DF'  22 0000:1A          SHLD    LOPPT#
00E2'  01 000A          LXI     B,10
00E5'  2A 0000:14          LHL    LOPVAR#
00E8'  EB          XCHG
00E9'  60          MOV     H,B
00EA'  68          MOV     L,B
00EB'  39          DAD    SP
00EC'  1801          JMPR   FR6
00EE'  09          FR5:   DAD    B
00EF'  7E          FR6:   MOV     A,M
00F0'  23          INX     H
00F1'  B6          ORA     M
00F2'  2817          JRZ    FR7
00F4'  7E          MOV     A,M
00F5'  2B          DCX     H
00F6'  BA          CMP     D
00F7'  20F5          JRNZ   FR5
00F9'  7E          MOV     A,M
00FA'  AB          XRA     E

```

00FB'	20F1		JRNZ	FR5
00FD'	EB		XCHG	
00FE'	67		MOV	H,A
00FF'	6F		MOV	L,A
0100'	39		DAD	SP
0101'	44		MOV	B,H
0102'	4D		MOV	C,L
0103'	21 000A		LXI	H,10
0106'	19		DAD	D
0107'	CD 0000:1B		CALL	MVDOWN#
010A'	F9		SPHL	
010B'	2A 0000:1A	FR7:	LHLD	LOPPT#
010E'	EB		XCHG	
010F'	F7		RST	RSTFIN
0110'	CD 0000:1C	NEXT::	CALL	TSTV#
0113'	DA 0000:15		JC	QWHAT#
0116'	22 0000:1D		SHLD	VARNXT#
0119'	D5	NX1:	PUSH	D
011A'	EB		XCHG	
011B'	2A 0000:14		LHLD	LOPVAR#
011E'	7C		MOV	A,H
011F'	B5		ORA	L
0120'	CA 0000:1E		JZ	AWHAT#
0123'	CD 0000:1F		CALL	COMP#
0126'	2809		JRZ	NX2
0128'	D1		POP	D
0129'	CD 01F2'		CALL	POPA
012C'	2A 0000:1D		LHLD	VARNXT#
012F'	18E8		JMPR	NX1
0131'	EB	NX2:	XCHG	
0132'	CD 0000:20		CALL	LDE#
0135'	6F		MOV	L,A
0136'	13		INX	D
0137'	CD 0000:20		CALL	LDE#
013A'	67		MOV	H,A
013B'	EB		XCHG	
013C'	2A 0000:18		LHLD	LOPINC #
013F'	E5		PUSH	H
0140'	7C		MOV	A,H
0141'	AA		XRA	D
0142'	7A		MOV	A,D
0143'	19		DAD	D
0144'	FA 014B'		JM	NX3
0147'	AC		XRA	H
0148'	FA 0172'		JM	NX5
014B'	EB	NX3:	XCHG	
014C'	2A 0000:14		LHLD	LOPVAR#
014F'	7B		MOV	A,E
0150'	CD 0000:21		CALL	STHL#
0153'	23		INX	H
0154'	7A		MOV	A,D
0155'	CD 0000:21		CALL	STHL#
0158'	2A 0000:17		LHLD	LOPLMT#
015B'	F1		POP	PSW
015C'	B7		ORA	A

015D'	F2 0161'		JP	NX4		
0160'	EB		XCHG			
0161'	CD 0000:22	NX4:	CALL	CKHLDE#		
0164'	D1		POP	D		
0165'	380D		JRC	NX6		
0167'	2A 0000:19		LHLD	LOPLN#		
016A'	22 0000:12		SHLD	CURRNT#		
016D'	2A 0000:1A		LHLD	LOPPT#		
0170'	EB		XCHG			
0171'	F7		RST	RSTFIN		
0172'	E1	NX5:	POP	H		
0173'	D1		POP	D		
0174'	CD 01F2'	NX6:	CALL	POPA		
0177'	F7		RST	RSTFIN		
			; REM, IF, INPUT, LET			
0178'	21 0000	REM::	LXI	H,0		
017B'	1801		JMPR	IF1		
017D'	D7	IFF::	RST	RSTEXP		
017E'	7C	IF1:	MOV	A,H		
017F'	B5		ORA	L		
0180'	C2 0000:0D		JNZ	RUNSML#		
0183'	CD 0000:23		CALL	FNDSKP#		
0186'	D2 0000:06		JNC	RUNTSL#		
0189'	C3 0000:24		JMP	RSTART#		
018C'	2A 0000:25	INPERR::	LHLD	STKINP#		
018F'	F9		SPHL			
0190'	E1		POP	H		
0191'	22 0000:12		SHLD	CURRNT#		
0194'	D1		POP	D		
0195'	D1		POP	D		
0196'		INPUT	==:	.		
0196'	D5	IP1:	PUSH	D		
0197'	CD 0000:26		CALL	QTSTG#		
019A'	1823		JMPR	IP8		
019C'	CD 0000:1C	IP2:	CALL	TSTV#		
019F'	3817		JRC	IP5		
01A1'	CD 01CF'	IP3:	CALL	IP12		
01A4'	11 0000:27		LXI	D, BUFFER#		
01A7'	D7		RST	RSTEXP		
01A8'	D1		POP	D		
01A9'	EB		XCHG			
01AA'	7B		MOV	A,E		
01AB'	CD 0000:21		CALL	STHL#		
01AE'	23		INX	H		
01AF'	7A		MOV	A,D		
01B0'	CD 0000:21		CALL	STHL#		
01B3'	E1	IP4:	POP	H		
01B4'	22 0000:12		SHLD	CURRNT#		
01B7'	D1		POP	D		
01B8'	F1	IP5:	POP	PSW		
01B9'	CF	IP6:	TSTCC	COMMA, IP7L	RST	1
01BA'	2C	+	.BYTE	COMMA		
01BB'	02	+	.BYTE	IP7-, -1		
		+]				
01BC'	18D8		JMPR	INPUT		

STDBAS -

01BE'	F7	IP7:	RST	RSTFIN
01BF'	D5	IP8:	PUSH	D
01C0'	CD 0000:1C		CALL	TSTV#
01C3'	3003		JRNC	IP11
01C5'	C3 0000:15	IP10:	JMP	QWHAT#
01C8'	43	IP11:	MOV	B,E
01C9'	D1		POP	D
01CA'	CD 0000:28		CALL	PRTCHS#
01CD'	18D2		JMPR	IP3
01CF'	C1	IP12:	POP	B
01D0'	D5		PUSH	D
01D1'	EB		XCHG	
01D2'	2A 0000:12		LHLD	CURRNT#
01D5'	E5		PUSH	H
01D6'	21 0196'		LXI	H,IP1
01D9'	22 0000:12		SHLD	CURRNT#
01DC'	21 0000		LXI	H,0
01DF'	39		DAD	SP
01E0'	22 0000:25		SHLD	STKINP#
01E3'	D5		PUSH	D
01E4'	C5		PUSH	B
01E5'	3E20		MVI	A,' '
01E7'	C3 0000:29		JMP	GETLN#
01EA'	1A	DEFLT::	LDAX	D
01EB'	FE0D		CPI	CR
01ED'	28CF		JRZ	IP7
01EF'	C3 0000:15		JMP	QWHAT#
01F2'	C1	POPA:	POP	B
01F3'	E1		POP	H
01F4'	22 0000:14		SHLD	LOPVAR#
01F7'	7C		MOV	A,H
01F8'	B5		ORA	L
01F9'	2810		JRZ	FP1
01FB'	E1		POP	H
01FC'	22 0000:18		SHLD	LOPINC#
01FF'	E1		POP	H
0200'	22 0000:17		SHLD	LOPLMT#
0203'	E1		POP	H
0204'	22 0000:19		SHLD	LOPLN#
0207'	E1		POP	H
0208'	22 0000:1A		SHLD	LOPPT#
020B'	C5	PP1:	PUSH	B
020C'	C9		RET	
020D'	21 0000:2A	PUSHA:	LXI	H,STKLMT#
0210'	C1		POP	B
0211'	A7		ANA	A
0212'	ED72		DSBC	SP
0214'	D2 0000:2B		JNC	QSORRY#
0217'	2A 0000:14		LHLD	LOPVAR#
021A'	7C		MOV	A,H
021B'	B5		ORA	L
021C'	2813		JRZ	PU1
021E'	2A 0000:1A		LHLD	LOPPT#
0221'	E5		PUSH	H
0222'	2A 0000:19		LHLD	LOPLN#

STDBAS -

```
0225'  E5          PUSH      H
0226'  2A 0000:17  LHLD     LOPLMT#
0229'  E5          PUSH      H
022A'  2A 0000:18  LHLD     LOPINC  #
022D'  E5          PUSH      H
022E'  2A 0000:14  LHLD     LOPVAR#
0231'  E5          PU1:     PUSH      H
0232'  C5          PUSH      B
0233'  C9          RET
          .END
```


GRAPHI -

```

        .PREL
        .IDENT  GRAPHICS
        .INSERT HVG.LIB
        .INSERT MACRO.LIB
        .INSERT BBEQU.ASM
0000'   .RELOC
0000'   80       FIXTBL: .BYTE  080H
0001'   20       .BYTE  020H
0002'   08       .BYTE  08H
0003'   02       .BYTE  2H
                ; FUNCTION TO RETURN STATE OF ADDRESSED PIXEL
                ; IE... PIX(X,Y)= 1 IF PIXEL IS 1, 0 IF 0
0004'   CF       PIXFUN:: TSTC  '(',PIXDUDI  RST
                1
0005'   28       + .BYTE  '<<'
0006'   20       + .BYTE  PIXDUD--1
                +]
0007'   C5       PUSH  B
0008'   D7       RST   RSTEXP
0009'   E5       PUSH  H
000A'   CF       TSTCC COMMA,PIXDUDI  RST  1
000B'   2C       + .BYTE  COMMA
000C'   1A       + .BYTE  PIXDUD--1
                +]
000D'   D7       RST   RSTEXP
000E'   CF       TSTC  ')',PIXDUDI  RST  1
000F'   29       + .BYTE  '>'
0010'   16       + .BYTE  PIXDUD--1
                +]
0011'   C1       POP   B
0012'   D5       PUSH  D      ; SAVE PTR
0013'   55       MOV   D,L
0014'   59       MOV   E,C
0015'   CD 01A9' CALL  R2A
0018'   EB       XCHG
0019'   FF       INDEXB 1[INTP%[.IFE .INTP.,[RST 7]]
001A'   5D       +.BYTE  92+1]
001B'   0000'   .WORD  FIXTBL
001D'   1A       LDAX  D      ; GET BYTE FROM SCREEN
001E'   A6       ANA   M      ; MASK OFF NONSENSE
001F'   2600    MVI   H,0
0021'   6C       MOV   L,H
0022'   D1       POP   D
0023'   C1       POP   B
0024'   C8       RZ
0025'   23       INX   H
0026'   C9       RET
0027'   C3 0000:04 PIXDUD: JMP  QWHAT
                ; BOX DRAW ROUTINE
002A'   D7       BOXDRW:: RST  RSTEXP  ; GET X
002B'   E5       PUSH  H
002C'   CF       TSTCC COMMA,BOXDUD  ; FIND COMMAI
                RST  1
002D'   2C       + .BYTE  COMMA
002E'   55       + .BYTE  BOXDUD  --1
    
```

```

002F'   D7           +]           RST      RSTEXP ; GET Y
0030'   E5           PUSH      H
0031'   CF           TSTCC    COMMA,BOXDUDE   RST      1
0032'   2C           +         .BYTE    COMMA
0033'   50           +         .BYTE    BOXDUD--1
                                +]
0034'   CD 00D3'    CALL      EXPRCP ; XS
0037'   F5           PUSH      PSW
0038'   CF           TSTCC    COMMA,BOXDUDE   RST      1
0039'   2C           +         .BYTE    COMMA
003A'   49           +         .BYTE    BOXDUD--1
                                +]
003B'   CD 00D3'    CALL      EXPRCP ; YS
003E'   F5           PUSH      PSW
003F'   CF           TSTCC    COMMA,BOXDUDE   RST      1
0040'   2C           +         .BYTE    COMMA
0041'   42           +         .BYTE    BOXDUD--1
                                +]
0042'   D7           RST      RSTEXP
0043'   D5           PUSH      D
0044'   DDE1        POP       X
0046'   F1           POP       PSW      ; RESTORE YS
0047'   47           MOV      B,A
0048'   F1           POP       PSW      ; AND XS
0049'   4F           MOV      C,A
004A'   7D           MOV      A,L      ; PRESERVE FLAG
004B'   E1           POP       H
004C'   55           MOV      D,L
004D'   E1           POP       H
004E'   5D           MOV      E,L
004F'   6F           MOV      L,A
                                ; NOW WE HAVE: B=YS, C=XS, D=Y, E=X, L=FLAG
                                ; LIMIT CHECK Y
0050'   60           MOV      H,B
0051'   25           DCR      H
0052'   CB3C        SRLR     H
0054'   7A           MOV      A,D
0055'   CD 0087'    CALL      SABS
0058'   84           ADD      H
0059'   FE2C        CPI      44
005B'   3023        JRNC    BOXNDR
005D'   7A           MOV      A,D
005E'   84           ADD      H
005F'   57           MOV      D,A
                                ; AND X
0060'   61           MOV      H,C
0061'   CB3C        SRLR     H
0063'   7B           MOV      A,E
0064'   CD 0087'    CALL      SABS
0067'   84           ADD      H
0068'   FE51        CPI      81
006A'   3014        JRNC    BOXNDR
006C'   7B           MOV      A,E
006D'   94           SUB      H

```

```

006E' 5F          MOV      E,A
          ; DIDDLE WITH FLAG BYTE
006F' 7D          MOV      A,L
0070' E603        ANI      3          ; MODULO 4
0072' 280C        JRZ     BOXNDR ; SKIP DRAW IF ZERO
0074' D602        SUI     2          ; ELSE SUBTRACT 2 FOR MA
          SK
0076' F5          BOXNDR: PUSH   PSW
0077' CD 01A9'    CALL   R2A
          ; HL = ABS ADDR, A = SA, B=YS, C=XS
007A' D30C        OUT     MAGIC
007C' F1          POP     PSW
007D' CD 008C'    CALL   BOXPUT
0080' DDE5        BOXNDR: PUSH   X
0082' D1          POP     D
0083' F7          RST     RSTFIN
0084' C3 0000:04 BOXDUD: JMP    QWHAT#
0087' A7          SABS:  ANA     A
0088' F0          RP
0089' ED44        NEG
008B' C9          RET
          ; SUBROUTINE TO DRAW A BOX ON SCREEN
008C' 5F          BOXPUT:  MOV    E,A
008D' 79          MOV    A,C      ; D = X / 4
008E' 0F          RRC
008F' 0F          RRC
0090' E63F        ANI     3FH
0092' 3C          INR     A
0093' 57          MOV    D,A
          ; PAINT FULL BYTE STRIPES
0094' 15          MPT1:  DCR    D
0095' 2807        JRZ     MPT2
0097' 3EAA        MVI     A,10101010B
0099' CD 00B1'    CALL   STRIPE
009C' 18F6        JMPR   MPT1
009E' 79          MPT2:  MOV    A,C
009F' E603        ANI     3
00A1' 3C          INR     A
00A2' 4F          MOV    C,A
00A3' AF          XRA     A
00A4' 0D          MPT3:  DCR    C
00A5' 2806        JRZ     MPT4
00A7' 0F          RRC
00A8' 0F          RRC
00A9' F680        ORI     10000000B
00AB' 18F7        JMPR   MPT3
00AD' CD 00B1'    MPT4:  CALL   STRIPE
00B0' AF          XRA     A
          ; FALL INTO ...
          ; SUBROUTINE TO PAINT A STRIPE
00B1' E5          STRIPE: PUSH   H
00B2' C5          PUSH   B
00B3' 32 0FFF     STA     URINAL
00B6' 3A 4FFF     LDA     URINAL+4000H
00B9' 4F          MOV    C,A
  
```

```

00BA✓ 7B          STRP1:  MOV    A,E
00BB✓ FE01        CPI     1
00BD✓ 2002        JRNZ   STRP2
00BF✓ 7E          MOV    A,M
00C0✓ A9          XRA    C
00C1✓ AE          STRP2:  XRA    M
00C2✓ A1          ANA    C
00C3✓ AE          XRA    M
00C4✓ 77          MOV    M,A
00C5✓ 7D          MOV    A,L
00C6✓ C628        ADI    BYTEPL
00C8✓ 6F          MOV    L,A
00C9✓ 7C          MOV    A,H
00CA✓ CE00        ACI    0
00CC✓ 67          MOV    H,A
00CD✓ 10EB        DJNZ   STRP1
00CF✓ C1          POP    B
00D0✓ E1          POP    H
00D1✓ 23          INX    H
00D2✓ C9          RET

: ROUTINE TO GET EXPRESSION, MAKING SURE IT IS POSITIVE
: AND NONZERO
00D3✓ D7          EXPRCP: RST    RSTEXP
00D4✓ 7C          MOV    A,H
00D5✓ B7          ORA    A
00D6✓ 2032        JRNZ   LINED4
00D8✓ B5          ORA    L
00D9✓ 282F        JRZ    LINED4
00DB✓ C9          RET

: LINE DRAWER
LINEDR::
00DC✓ D7          RST    RSTEXP
00DD✓ 7D          MOV    A,L
00DE✓ F5          PUSH   PSW
00DF✓ CF          TSTCC COMMA,LINED4[ RST    1
00E0✓ 2C          +     .BYTE COMMA
00E1✓ 28          +     .BYTE LINED4-,-1
+1

00E2✓ D7          RST    RSTEXP
00E3✓ 7D          MOV    A,L
00E4✓ F5          PUSH   PSW
00E5✓ CF          LINED1: TSTCC COMMA,LINED4[ RST    1
00E6✓ 2C          +     .BYTE COMMA
00E7✓ 22          +     .BYTE LINED4-,-1
+1

00E8✓ D7          RST    RSTEXP
00E9✓ 44          MOV    B,H
00EA✓ 4D          MOV    C,L
00EB✓ D5          PUSH   D
00EC✓ DDE1        POP    X
00EE✓ ED5B 0000:05 LDDED OLDXY#
00F2✓ F1          POP    PSW
00F3✓ 67          MOV    H,A
00F4✓ F1          POP    PSW
00F5✓ 6F          MOV    L,A
  
```

GRAPHI -

```

00F6' 22 0000:05          SHLD  OLDXY#  ; SET NEW LAST PLACE
                        ; DIDDLE WITH FLAG BYTE
00F9' 79                  MOV   A,C
00FA' E603                ANI   3
00FC' 2808                JRZ   LINED3
00FE' D602                SUI   2
0100' 32 0000:06        LINED2: STA  PIXVAL# ; SET PIXVAL
0103' CD 010D'          CALL  DVECT
0106' DDE5                LINED3: PUSH X
0108' D1                  POP   D
0109' F7                  RST   RSTFIN
010A' C3 0000:07        LINED4: JMP  QHOW#
                        ; LARRY LIVERMORE'S VECTOR DRAWING ALGORITHM
                        ; H=Y1, L=X1, D=Y2, E=X2
010D' D5                  DVECT: PUSH D
010E' 45                  MOV   B,L
010F' 4B                  MOV   C,E
0110' CD 0177'          CALL  CDELTA
0113' 58                  MOV   E,B
0114' 69                  MOV   L,C
0115' 44                  MOV   B,H
0116' 4A                  MOV   C,D
0117' CD 0177'          CALL  CDELTA
011A' 61                  MOV   H,C
011B' 50                  MOV   D,B
                        ; WE NOW HAVE: H=SGN(DY), L=SGN(DX)
                        ; D=ABS(DY), E=ABS(DX)
011C' 22 0000:08        SHLD  INCRO#
                        ; DECIDE WHICH DELTA IS LARGER
                        ; CALL BIGGER MX, SMALLER MN
011F' 0E00                MVI   C,0
0121' 7A                  MOV   A,D
0122' BB                  CMP   E
0123' 3803                JRC   VECT1
0125' 53                  MOV   D,E
0126' 5F                  MOV   E,A
0127' 0C                  INR   C
0128' 7A                  VECT1: MOV  A,D      ; MX TO A
0129' CB3F                SRLR  A
012B' 47                  MOV   B,A
012C' EB                  XCHG
012D' 22 0000:09        SHLD  MNMX#
0130' D1                  POP   D
0131' 7D                  MOV   A,L
0132' 3C                  INR   A      ; MAKE SURE LAST PIXEL W
                        RITTEN
                        ; THE INFAMOUS PIXEL PAINTING LOOP
0133' F5                  VECT2: PUSH PSW
0134' CD 0199'          CALL  R2ACLP
0137' 3819                JRC   VECT2A
0139' C5                  PUSH  B
013A' E5                  PUSH  H
013B' 4F                  MOV   C,A
013C' 0600                MVI   B,0
013E' 21 0000'          LXI   H,PIXTBL

```

```

0141' 09          DAD      B
0142' 46          MOV      B,M
0143' E1          POP      H
0144' 3A 0000:06 LDA      PIXVAL#
0147' FE01        CPI      1
0149' 2002        JRNZ     VECT9
014B' 7E          MOV      A,M
014C' A8          XRA      B
014D' AE          VECT9:  XRA      M
014E' A0          ANA      B
014F' AE          XRA      M
0150' 77          MOV      M,A
0151' C1          POP      B
          ; INCREMENT COORDINATES
0152' 2A 0000:09 VECT2A:  LHLD     MNMX#
0155' 78          MOV      A,B
0156' 84          ADD      H
0157' BD          CMP      L
0158' 380D        JRC      VECT4
015A' 95          SUB      L
015B' 47          MOV      B,A
015C' 2A 0000:08 LHLD     INCRO#
015F' 7A          MOV      A,D
0160' 84          ADD      H
0161' 57          MOV      D,A
0162' 7B          VECT3:  MOV      A,E
0163' 85          ADD      L
0164' 5F          MOV      E,A
0165' 180B        JMPR     VECT5
0167' 47          VECT4:  MOV      B,A
0168' 2A 0000:08 LHLD     INCRO#
016B' 79          MOV      A,C
016C' 0F          RRC
016D' 30F3        JRNC     VECT3
016F' 7A          MOV      A,D
0170' 84          ADD      H
0171' 57          MOV      D,A
          ; END OF LOOP
0172' F1          VECT5:  POP      PSW
0173' 3D          DCR      A
0174' 20BD        JRNZ     VECT2
0176' C9          RET
          ; SUBROUTINE TO COMPUTE DELTA AND INCREMENT FOR
          TWO COORDINATES
0177' E5          CDELTA:  PUSH     H
0178' D5          PUSH     D
0179' 69          MOV      L,C
017A' CD 0000:0A CALL     SGNEXT#
017D' EB          XCHG
017E' 68          MOV      L,B
017F' CD 0000:0A CALL     SGNEXT#
0182' AF          XRA      A
0183' ED52        DSBC
          ; COMPUTE SGN(DELTA) AND ABS(DELTA)
0185' B4          ORA      H

```

GRAPHI -

```

0186/ 2807          JRZ      CDELT1
0188/ 4F           MOV      C,A
0189/ 7D           MOV      A,L
018A/ ED44        NEG
018C/ 47           MOV      B,A
018D/ 1807        JMPR     CDELT3
018F/ B5          CDELT1: ORA      L      ; POS CASE 0?
0190/ 2802        JRZ      CDELT2
0192/ 3E01        MVI      A,1
0194/ 45          CDELT2: MOV      B,L
0195/ 4F           MOV      C,A
0196/ D1          CDELT3: POP      D
0197/ E1          POP      H
0198/ C9          RET

; RELATIVE TO ABSOLUTE CONVERSION WITH CLIPPING
0199/ 7B          R2ACLP: MOV     A,E
019A/ FE50        CPI      80
019C/ 3803        JRC      ..OKY
019E/ FEB0        CPI      0BOH   ; ** -80 **
01A0/ D8          RC
01A1/ 7A          ..OKY: MOV     A,D
01A2/ FE2C        CPI      44
01A4/ 3803        JRC      R2A
01A6/ FED4        CPI      0D4H
01A8/ D8          RC

; ...
; RELATIVE TO ABSOLUTE CONVERSION
01A9/ D5          R2A::  PUSH     D
01AA/ 7A          MOV      A,D
01AB/ 2F          CMA
01AC/ C62C        ADI      44
01AE/ 57          MOV      D,A
01AF/ 7B          MOV      A,E
01B0/ C650        ADI      80
01B2/ 5F          MOV      E,A
01B3/ AF          XRA      A
01B4/ FF          RELAB1[INTP%[.IFE .INTP.,[RST 7]]
01B5/ 3A          +.BYTE 58+0]
01B6/ EB          XCHG
01B7/ D1          POP      D
01B8/ C9          RET

; COMMAND TO CLEAR SCREEN
01B9/             CLRSCR::
01B9/ 21 4000     LXI      H,4000H
01BC/ 7E          CLRRLP: MOV     A,M
01BD/ E655        ANI      01010101B
01BF/ 77          MOV      M,A
01C0/ 23          INX      H
01C1/ 7C          MOV      A,H
01C2/ FE4E        CPI      4EH
01C4/ 20F6        JRNZ     CLRRLP

; RESET VDM GOODIES
01C6/ D5          PUSH     D
01C7/ FF          MOVE     1[INTP%[.IFE .INTP.,[RST 7]]
01C8/ 5F          +.BYTE 94+1]

```

change

```
01C9' 0000:0B      .WORD  VDMX#  
01CB' 0006        .WORD  6  
01CD' 0006:0C      .WORD  INIDEV#+6  
01CF' D1          POP    D  
01D0' F7          RST    RSTFIN  
                .END
```


CHAR -

```

        .PREL
        .IDENT CHAR
        .INSERT HVG.LIB
        .INSERT MACRO.LIB
        .INSERT BBEQU.ASM
0000'   .RELOC
        ; TABLE GIVING ASCII CHARS FOR TOKENS
0000'   TOKTXT:
0000'   4C4953 .ASCII 'LIS'
0003'   D4     .BYTE 'T'+80H
0004'   434C4541 .ASCII 'CLEA'
0008'   D2     .BYTE 'R'+80H
0009'   5255   .ASCII 'RU'
000B'   CE     .BYTE 'N'+80H
000C'   4E4558 .ASCII 'NEX'
000F'   D4     .BYTE 'T'+80H
0010'   4C494E .ASCII 'LIN'
0013'   C5     .BYTE 'E'+80H
0014'   49     .BYTE 'I'
0015'   C6     .BYTE 'F'+80H
0016'   474F54 .ASCII 'GOT'
0019'   CF     .BYTE 'O'+80H
001A'   474F5355 .ASCII 'GOSU'
001E'   C2     .BYTE 'B'+80H
001F'   5245545552 .ASCII 'RETUR'
0024'   CE     .BYTE 'N'+80H
0025'   424F   .ASCII 'BO'
0027'   D8     .BYTE 'X'+80H
0028'   464F   .ASCII 'FO'
002A'   D2     .BYTE 'R'+80H
002B'   494E5055 .ASCII 'INPU'
002F'   D4     .BYTE 'T'+80H
0030'   5052494E .ASCII 'PRIN'
0034'   D4     .BYTE 'T'+80H
0035'   535445 .ASCII 'STE'
0038'   D0     .BYTE 'P'+80H
0039'   524E   .ASCII 'RN'
003B'   C4     .BYTE 'D'+80H
003C'   54     .BYTE 'T'
003D'   CF     .BYTE 'O'+80H
    
```

```

003E'   F5
003F'   3A 0000:04
0042'   2F
0043'   C629
0045'   FE51
0047'   3801
0049'   AF
004A'   67
004B'   3A 0000:05
004E'   C64D
0050'   FE9D
0052'   3801
0054'   AF
    
```

```

        ; SUBROUTINE TO LOAD HL WITH VDM COORDINATES
        ; FROM DEVICE VARIABLES
LDVDMC: PUSH    PSW
        LDA     VDMY#
        CMA
        ADI     41
        CPI     81      ; OUT OF RANGE?
        JRC     LDVDM1 ; NO
        XRA     A
LDVDM1: MOV     H,A
        LDA     VDMX#  ; DIDDLE WITH X
        ADI     77
        CPI     157
        JRC     LDVDM2
        XRA     A
    
```

OK

CHAR -

```

0055' 6F LDVDM2: MOV L,A
0056' F1 POP PSW
0057' C9 RET
; SUBROUTINE TO STORE HL INTO VDM COORDINATE DEV
ICE VARIABLES
0058' E5 STVDMC: PUSH H
0059' 7C MOV A,H
005A' D629 SUI 41
005C' 2F CMA
005D' 6F MOV L,A
005E' CD 0000:06 CALL SGNEXT#
0061' 22 0000:04 SHLD VDMY#
0064' E1 POP H
0065' 7D MOV A,L
0066' D64D SUI 77
0068' 6F MOV L,A
0069' CD 0000:06 CALL SGNEXT#
006C' 22 0000:05 SHLD VDMX#
006F' C9 RET
; DEVICE VARIABLE TO OUTPUT CHARACTER ON VDM
0070' CF PUTCD: TSTC '<=>',PUTCD2[RST 1
0071' 3D + .BYTE '<=>'
0072' 04 + .BYTE PUTCD2-,-1
+J
0073' D7 RST RSTEXP
0074' 7D MOV A,L
0075' DF RST RSTOCH
0076' F7 RST RSTFIN
0077' C3 0000:07 PUTCD2: JMP QWHAT#
007A' 3E0D CRLF: MVI A,CR
; SUBROUTINE TO SIMULATE A CHARACTER DISPLAY IN
; THE ARCADE FRAME BUFFER. THE SIMULATED VDM HA
S
; DIMENSIONS 26 CHARS BY 11 LINES. THE CHARACTE
R GRAPHICS ARE 5 X 7
; IN A 6 X 8 FRAME. ALTERNATE FONT IS USED TO G
ET THIS.
; THE 64 UPPER CASE ASCII CHARACTERS ARE DISPLAY
ED BY THIS
; HANDLER. THE ASCII CONTROL CHARACTERS CARRIAG
E RETURN AND
; RUBOUT ARE ALSO PROCESSED BY THIS HANDLER. CR
CAUSES
; THE DISPLAY TO GO TO THE NEXT LINE OF THE DISP
LAY, WITH
; SCROLL UP IF NECESSARY. RUBOUT CAUSES THE CUR
SOR TO MOVE
; BACKWARDS ONE CHARACTER POSITION.
; CHARACTER TO DISPLAY IS IN A. THE ALTERNATE R
EGISTER SET
; IS USED.
007C' E5 OUTCH: PUSH H
007D' D5 PUSH D
007E' C5 PUSH B
007F' F5 PUSH PSW

```

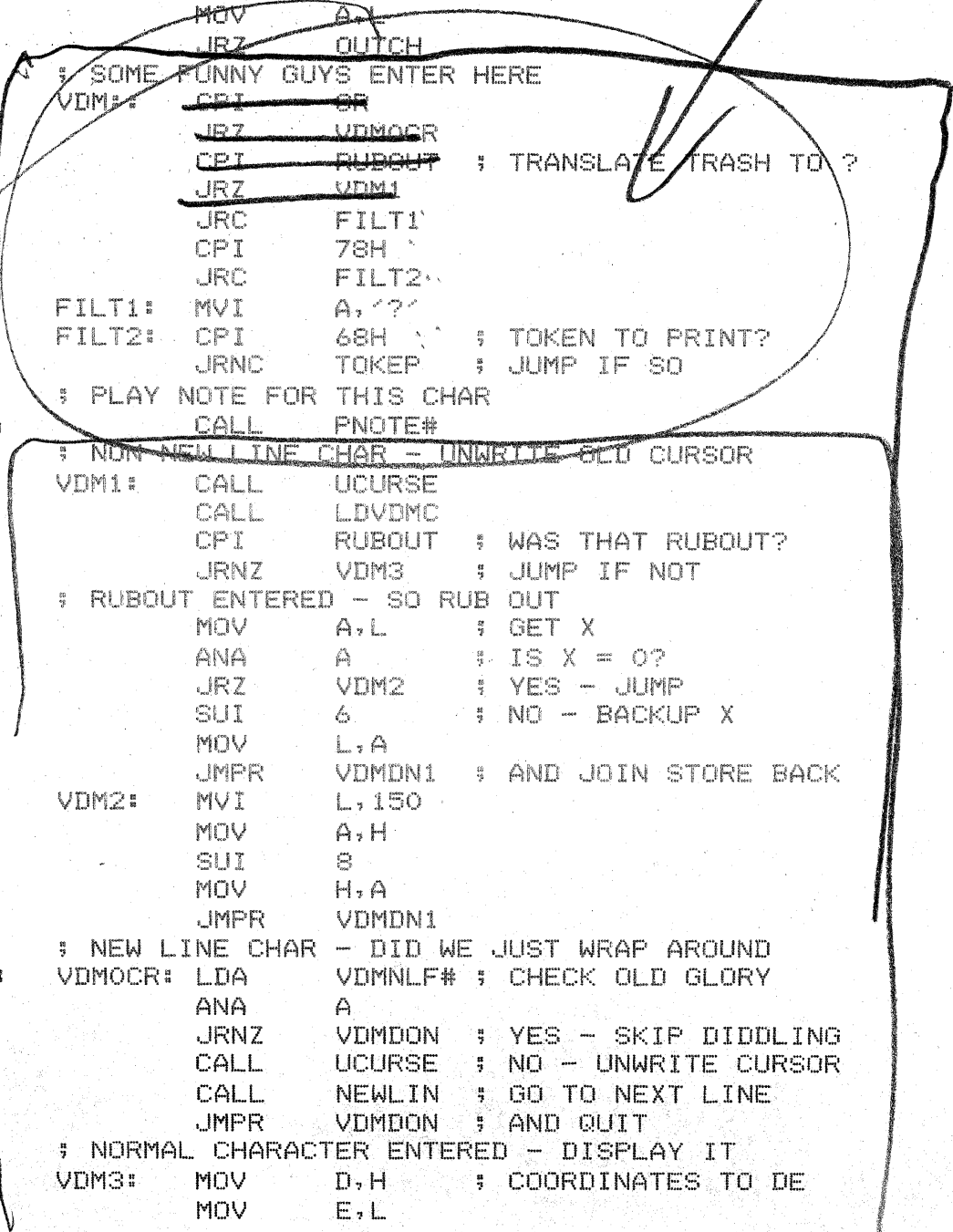
CHAR -

```

0080' 57          MOV      D,A
0081' CD 0000:08  CALL    TOUTCK#
0084' 7A          VDMCAL: MOV    A,D
0085' CD 0096'   CALL    VDM
0088' F1          POP     PSW
0089' C1          POP     B
008A' D1          POP     D
008B' E1          POP     H
008C' C9          RET
; EVEN STRANGER GUYS ENTER HERE ...
008D' 6F          VDMTOK: MOV    L,A
008E' 3A 0000:09 LDA    TAPEST#
0091' FE06        CPI     6
0093' 7D          MOV    A,L
0094' 28E6        JRZ    OUTCH
; SOME FUNNY GUYS ENTER HERE
0096' FE0D        VDM:  CPI     0F
0098' 282E        JRZ    VDMOCR
009A' FE1F        CPI     RUBOUT ; TRANSLATE TRASH TO ?
009C' 280F        JRZ    VDM1
009E' 3804        JRC    FILT1'
00A0' FE78        CPI     78H
00A2' 3802        JRC    FILT2'
00A4' 3E3F        FILT1: MVI   A,'?'
00A6' FE68        FILT2: CPI   68H ; TOKEN TO PRINT?
00A8' 304F        JRNC  TOKEP ; JUMP IF SO
; PLAY NOTE FOR THIS CHAR
00AA' CD 0000:0A CALL   PNOTE#
; NON-NEW LINE CHAR - UNWRITE OLD CURSOR
00AD' CD 010E'   VDM1:  CALL   UCURSE
00B0' CD 003E'   CALL   LDVDMC
00B3' FE1F        CPI     RUBOUT ; WAS THAT RUBOUT?
00B5' 201F        JRNZ   VDM3 ; JUMP IF NOT
; RUBOUT ENTERED - SO RUB OUT
00B7' 7D          MOV    A,L ; GET X
00B8' A7          ANA    A ; IS X = 0?
00B9' 2805        JRZ    VDM2 ; YES - JUMP
00BB' D606        SUI    6 ; NO - BACKUP X
00BD' 6F          MOV    L,A
00BE' 1831        JMPR   VDMDN1 ; AND JOIN STORE BACK
00C0' 2E96        VDM2:  MVI   L,150
00C2' 7C          MOV    A,H
00C3' D608        SUI    8
00C5' 67          MOV    H,A
00C6' 1829        JMPR   VDMDN1
; NEW LINE CHAR - DID WE JUST WRAP AROUND
00C8' 3A 0000:0B VDMOCR: LDA   VDMNLF# ; CHECK OLD GLORY
00CB' A7          ANA    A
00CC' 2026        JRNZ   VDMDON ; YES - SKIP DIDDLING
00CE' CD 010E'   CALL   UCURSE ; NO - UNWRITE CURSOR
00D1' CD 0112'   CALL   NEWLIN ; GO TO NEXT LINE
00D4' 181E        JMPR   VDMDON ; AND QUIT
; NORMAL CHARACTER ENTERED - DISPLAY IT
00D6' 54          VDM3:  MOV    D,H ; COORDINATES TO DE
00D7' 5D          MOV    E,L

```

Integrate



CHAR -

```

00D8' F680          ORI      80H      ; ALT FONT THE CHAR
00DA' OE18          MVI      C,011000B ; OR WRITE THE C
                                HAR
00DC' DD21 0000:00 LXI      X,ALTFON# ; USING ALTERNAT
                                E CHAR FONT
00E0' FF           CHRDIS ; IT      1|INTP%L.IFE .INTP.,|RST
                                7|]
00E1' 32           +.BYTE 50+0|
00E2' 7D           MOV      A,L      ; ADVANCE X POINTER
00E3' C606         ADI      6
00E5' 6F           MOV      L,A
00E6' FE9C         CPI      156     ; END OF LINE?
00E8' 2007         JRNZ     VMDN1    ; NO - JUMP
00EA' CD 0112'     CALL     NEWLIN  ; YES - NEW      1 LINE
00ED' 3E01         MVI      A,1     ; AND SET NEW LINE FORCE
                                D FLAG
00EF' 1804         JMPR     VMDN2
00F1' CD 0058'     VMDN1: CALL     STVDMC
00F4' AF           VMDN0: XRA      A      ; CLEAR NEW LINE FORCED
                                FLAG
00F5' 32 0000:00 VMDN2: STA      VDMNLF#
00F8' C9           RET
                                ; ROUTINE TO DISPLAY A TOKEN IN FULL FORM
00F9' CD 0164'     TOKEP: CALL     TOKEPT
00FC' 7E           TOKEP1: MOV     A,M
00FD' E67F         ANI      7FH
00FF' E5           PUSH     H
0100' CD 008D'     CALL     VDMTOK
0103' E1           POP      H
0104' 7E           MOV     A,M
0105' 23           INX     H
0106' 07           RLC
0107' 30F3         JRNC     TOKEP1
0109' 3E20         TOKEP2: MVI     A,' ' ; PUT SPACE AFTER TOKEN
010B' C3 008D'     JMP      VDMTOK ; AND RETURN
                                ; SUBROUTINE TO UNWRITE THE CURSOR
010E' 0E00         UCURSE: MVI     C,0
0110' 183E         JMPR     CURSE
                                ; SUBROUTINE TO DISPLAY NEW LINE
0112' CD 003E'     NEWLIN: CALL     LDVDMC
                                ; IS SCROLL UP NEEDED?
0115' 2E00         MVI     L,0
0117' 7C           MOV     A,H
0118' FE50         CPI     80
011A' 202D         JRNZ     NEWL1 ; JUMP IF NOT NEEDED
                                ; SCROLL UP IS NEEDED
011C' CD 0058'     CALL     STVDMC
011F' 21 4DC0     LXI     H,4DC0H
0122' 7E           SCRL9: MOV     A,M
0123' E655         ANI     01010101B
0125' 77           MOV     M,A
0126' 23           INX     H
0127' 7D           MOV     A,L
0128' FE20         CPI     20H
012A' 20F6         JRNZ     SCRL9

```

CHAR -

```

012C' 0604          MVI      B,4
012E'  C5          SCRLP:  PUSH   B
012F'  21 4000     LXI      H,NORMEM
0132'  11 4050     LXI      D,NORMEM+80
0135'  01 980E     LXI      B,0980EH
0138'  1A          SCRUP:  LDAX   D
0139'  AE          XRA     M
013A'  E6AA       ANI     10101010B
013C'  AE          XRA     M
013D'  77          MOV     M,A
013E'  23          INX     H
013F'  13          INX     D
0140'  10F6       DJNZ   SCRUP
0142'  0D          DCR     C
0143'  20F3       JRNZ   SCRUP
0145'  C1          POP     B
0146'  10E6       DJNZ   SCRLP
0148'  C9          RET

;
0149'  C608       NEWL1:  ADI     8
014B'  67          MOV     H,A
014C'  CD 0058'   CALL   STDPMC
014F'  C9          RET

; SUBROUTINE TO PAINT CURSOR
; C = DATA TO PAINT 00 OR AA
CURSE:: PUSH   PSW
        CALL   LDVDMC
PCURS1: XCHG
        XRA     A
        RELAB1[INTF%].IFE .INTP.,[RST 7]]
+.BYTE 58+01
        OUT    MAGIC
        XCHG
        MOV     A,C
015C'  01 0806     LXI     B,0806H
015F'  CD 0000:0D CALL   BOXPUT#
0162'  F1          POP     PSW
0163'  C9          RET

; SUBROUTINE TO POINT AT STRING FOR TOKEN
0164'  21 0000'   TOKEPT: LXI     H,TOKTXT ; POINT
        AT LIST

0167'  D668       SUI     68H
0169'  C8          JOKEP1: RZ
016A'  CB7E       JOKEP2: BIT     7,M ; MOVE PAST NEXT WORD
016C'  23          INX     H
016D'  28FB       JRZ     JOKEP2
016F'  3D          DCR     A
0170'  18F7       JMPR   JOKEP1
        .END
    
```

EZKP -

```

                                .PREL
                                .IDENT  EZKP
                                .INSERT  HVG.LIB
                                .INSERT  MACRO.LIB
                                .INSERT  MUSIC.LIB
                                .INSERT  BBEQU.ASM
0000'                                .RELOC
                                ; KB - FUNCTION TO RETURN NEXT CHARACTER FROM KE
                                YBOARD
0000'    C5    GETKB::  PUSH    B
0001'    D5    PUSH    D
0002'    CD 000B' CALL    CHKID
0005'    D1    POP     D
0006'    C1    POP     B
0007'    6F    MOV     L,A
0008'    2600 MVI     H,0
000A'    C9    RET

                                ; NEW KEYBOARD HANDLER
                                ; WITH SHIFT KEY ROLLOVER
000B'    3A 0000:04 CHKID:: LDA    TAPEST# ; TAPE INPUT WANTED?
000E'    3D    DCR     A
000F'    201A  JRNZ   CHKIDQ ; NO - SKIP IT THEN
                                ; YES - CHECK FOR ABORT KEYSTROKE
0011'    CD 00CC' CALL    KEYSCN
0014'    C2 0000:05 JNZ    INITO#
                                ; GET CHARACTER FROM THE BUFFER
0017'    2A 0000:06 LHLD   CONPRO#
001A'    7C    MOV     A,H
001B'    BD    CMP     L      ; ARE POINTERS EQUAL?
001C'    28ED  JRZ    CHKID  ; YEP - LOOP WAITING
001E'    21 0000:07 LXI    H,TAPBUF#
0021'    6F    MOV     L,A      ; MAKE OFFICIAL POINTER
0022'    4E    MOV     C,M
0023'    CD 0000:08 CALL    BUMPTR#
0026'    32 0000:09 STA    CONPTR# ; UPDATE MY POINTER
0029'    79    MOV     A,C      ; PASS BACK THE CHARACTE
                                R
002A'    C9    RET
002B'    CD 00CC' CHKIDQ: CALL   KEYSCN ; MAKE SURE PREVIOUS KEY
                                RELEASED
002E'    20FB  JRNZ   CHKIDQ
                                ; AWAIT DEBOUNCE TIMER COUNTDOWN
0030'    21 0000:0A CHKIDQ: LXI    H,KEYTMR#
0033'    3606 MVI     M,6      ; SET IT
0035'    7E    LOOPER: MOV    A,M
0036'    A7    ANA    A
0037'    20FC  JRNZ   LOOPER
                                ; SAVE BACKGROUND COLOR
0039'    3A 0000:0B LDA    DEVCL0#
003C'    F5    PUSH   PSW
                                ; ASSUME FIRST LEVEL KEYCODE
003D'    21 00F6' LXI    H,FIRSTL
0040'    E5    GETK1: PUSH  H      ; SAVE TABLE PTR
                                ; SCAN ONLY FOR SHIFT KEYS
0041'    21 0149' LXI    H,KTBL4

```

```

0044' 11 FFEB          LXI    D,-21    ; ** SIZE OF LOOKUP TABL
                                E
0047' 01 0414          LXI    B,0414H
004A' ED78            GETK2: INP    A          ; INPUT FROM PORT
004C' E620            ANI    20H      ; SHIFT KEY DOWN?
004E' 2007            JRNZ   GETK3    ; JUMP IF YEP
0050' 19              DAD    D          ; ELSE TO NEXT TABLE
0051' 0C              INR    C          ; AND PORT
0052' 10F6            DJNZ   GETK2
                                ; NO SHIFT KEY IS DOWN - USE WHATEVER WE HAD BEF
                                ORE
0054' E1              POP    H
0055' 180F            JMPR  GETK5
                                ; A SHIFT KEY IS DOWN - SAME OLD STORY?
0057' D1              GETK3: POP    D          ; DISCARD OLD BELIEFS
0058' 3A 0000:0B      LDA    DEVCL0# ; IS CURRENT SCREEN COLO
                                R
005B' BE              CMP    M          ; THE SAME AS WHAT WE WO
                                ULD SET?
005C' 2807            JRZ    GETK4    ; YEP - DON'T BOTHER WIT
                                H FEEDBACK
005E' 7E              MOV    A,M      ; ELSE SET NEW COLOR
005F' 32 0000:0B      STA    DEVCL0#
0062' CD 0099'        CALL  WCLICK   ; AND GO CLICK
0065' 23              GETK4: INX    H          ; SKIP COLOR BYTE
                                ; NOW SCAN FOR ANY 'NORMAL' KEY DEPRESSION
0066' CD 00CC'        GETK5: CALL  KEYS0N
0069' 28D5            JRZ    GETK1    ; JUMP IF NO KEY DOWN
                                ; WE GOT ONE - CONVERT TO ASCII
006B' 3D              DCR    A          ; BY TABLE LOOKUP
006C' 4F              MOV    C,A
006D' 0600            MVI    B,0
006F' 09              DAD    B
0070' F1              POP    PSW      ; RESTORE COLOR
0071' 32 0000:0B      STA    DEVCL0#
0074' 7E              MOV    A,M      ; GET CODE
0075' A7              ANA    A          ; A HLT PERCHANCE?
0076' 284E            JRZ    INIJMP   ; YEP - RESET
0078' FE01            CPI    1          ; AN ERROR?
007A' CA 000B'        JZ     CHKIO    ; YEP - GO DOIT AGAIN
                                ; GOOD KEY...
007D' F5              CHKIO2: PUSH  PSW      ; NEED WE GO 'CLICK'?
007E' FE67            CPI    NLLN     ; REJECT TOKENS
0080' 300A            JRNC  NOCLK
0082' FF              INDEXB 1[INTP%[.IFE .INTP.,[RST 7]]
0083' 5D              +.BYTE 92+1]
0084' FFF3:0C        .WORD  NOTES#-CR
0086' 3C              INR    A
0087' 2003            JRNZ   NOCLK
0089' CD 0099'        CALL  WCLICK
008C' F1              NOCLK: POP    PSW
008D' FE67            CPI    NLLN
008F' C0              RNZ
0090' 21 0005          LXI    H,5
0093' 22 0000:0D      SHLD  NLLNCT# ; SET FLAG AND ZERO SUPR
  
```

```

0096' 3E0D          ESS
                                MVI    A,CR    ; PASS BACK CR AS FIRST
                                CHAR
0098' C9          RET
0099' 3A 0000:0E  WCLICK: LDA    NEWTMR#
009C' A7          ANA    A
009D' 20FA        JRNZ   WCLICK
009F' 3EFD        MVI    A,GO
00A1' 32 0000:0F  STA    MUZTON#
00A4' 3A 0000:10  LDA    DEVTEM# ; DON'T CLICK IF NT=0
00A7' A7          ANA    A
00A8' C8          RZ
00A9' 3E01        MVI    A,1
00AB' 32 0000:0E  STA    NEWTMR#
00AE' C9          RET

;
; SUBROUTINE TO CHECK FOR HLT KEY WHILE PGM RUNN
; ING
00AF' C5          WHATSU:: PUSH   B
00B0' D5          PUSH   D
00B1' CD 00CC'    CALL   KEYSN  ; GET KEY CODE
00B4' D602        SUI    2      ; FREEZE?
00B6' 2807        JRZ    FRZKEY
00B8' 3D          DCR    A
00B9' 280B        JRZ    INIUMP
00BB' FE01        CPI    1      ; RETURN Z STATUS FOR LI
                                ST KEY
00BD' 180A        JMPR   FRZGBK ; ELSE GO BACK TO CALLER
00BF' CD 00CC'    FRZKEY: CALL   KEYSN  ; SCAN FOR NONZERO KEY T
                                O REL
00C2' 28FB        JRZ    FRZKEY
00C4' FE03        CPI    3      ; HLT NAILED?
00C6' CA 0000:11  INIUMP: JZ     INIT#
00C9' D1          FRZGBK: POP    D
00CA' C1          POP    B
00CB' C9          RET
; SUBROUTINE TO SCAN TINY BASIC KEYBOARD
00CC' 01 0414    KEYSN: LXI   B,0414H ; B = CNT, C = PORT #
00CF' 11 0000:12  LXI   D,KEYTRK# ; DE = KEYBOARD
                                MEMORY
00D2' AF          XRA    A
00D3' FF          RANGED[INTP%[.IFE .INTP.,[RST 7]]
00D4' 76          +.BYTE 118+0]
00D5' ED78        KYSCN1: INP    A      ; LOOK AT COLUMN
00D7' E61F        ANI    1FH    ; ISOLATE THE RELEVANT
00D9' 2006        JRNZ   KYSCN2 ; JUMP IF BITS HIGH
00DB' 0C          INR    C      ; BUMP PORT #
00DC' 10F7        DJNZ   KYSCN1
00DE' AF          XRA    A      ; SET ZERO STATUS
00DF' 12          STAX   D      ; NOTHIN - SAY ZIP
00E0' C9          RET
; DEPRESSION FOUND - JUMP UP AND DOWN
00E1' 05          KYSCN2: DCR    B
00E2' 0E00        MVI    C,0    ; COME UP WITH BIT #

```


EZKP -

```

00E4' 0F          KYSCN4: RRC      ; SHIFT BIT OVER
00E5' 3803       JRC      KYSCN3  ; JUMP IF THE ONE
00E7' 0C          INR      C      ; ELSE COUNT UP
00E8' 18FA       JMPR     KYSCN4  ; AND TRY AGAIN
; FOUND BIT - ASSEMBLE KEYCODE
00EA' 79          KYSCN3: MOV      A,C      ; BIT # TO A
00EB' 07          RLC          ; * 4
00EC' 07          RLC          ;
00ED' B0          ORA      B      ; COMBINE WITH COL #
00EE' 3C          INR      A
00EF' 47          MOV      B,A
00F0' 1A          LDAX     D
00F1' A8          XRA      B
00F2' 78          MOV      A,B
00F3' C8          RZ          ; QUIT IF THE SAME
00F4' 12          STAX     D      ; ELSE UPDATE TRACKER
00F5' C9          RET
; TABLE OF FIRST LEVEL KEYCODES
00F6'             FIRRTL:
00F6' 0D          .BYTE     CR
00F7' 01          .BYTE     1
00F8' 00          .BYTE     0
00F9' 63          .BYTE     63H
00FA' 37          .BYTE     '7'
00FB' 38          .BYTE     '8'
00FC' 39          .BYTE     '9'
00FD' 62          .BYTE     62H
00FE' 34          .BYTE     '4'
00FF' 35          .BYTE     '5'
0100' 36          .BYTE     '6'
0101' 2D          .BYTE     '-'
0102' 31          .BYTE     '1'
0103' 32          .BYTE     '2'
0104' 33          .BYTE     '3'
0105' 2B          .BYTE     '+'
0106' 20          .BYTE     ' '
0107' 30          .BYTE     '0'
0108' 1F          .BYTE     RUBOUT
0109' 3D          .BYTE     '='
; FIRST SHIFT KEY
010A'             KTBL1:
010A' A7          .BYTE     0A7H   ; FIRST SHIFT KEY COLOR
010B' 0D          .BYTE     CR
010C' 01          .BYTE     1
010D' 00          .BYTE     0
010E' 01          .BYTE     1
010F' 41          .BYTE     'A'
0110' 44          .BYTE     'D'
0111' 47          .BYTE     'G'
0112' 4A          .BYTE     'J'
0113' 4D          .BYTE     'M'
0114' 50          .BYTE     'P'
0115' 53          .BYTE     'S'
0116' 56          .BYTE     'V'
0117' 59          .BYTE     'Y'

```

EZKF -

```

0118' 5F .BYTE 5FH
0119' 5E .BYTE 5EH
011A' 26 .BYTE '&'
011B' 24 .BYTE '$'
011C' 3C .BYTE '<'
011D' 28 .BYTE '('
011E' 23 .BYTE '#'
; SECOND SHIFT KEY
011F'
KTBL2:
011F' 5F .BYTE 05FH ; SECOND SHIFT KEY COLOR

0120' 0D .BYTE CR
0121' 2F .BYTE 2FH
0122' 00 .BYTE 0
0123' 5B .BYTE 5BH
0124' 42 .BYTE 'B'
0125' 45 .BYTE 'E'
0126' 48 .BYTE 'H'
0127' 4B .BYTE 'K'
0128' 4E .BYTE 'N'
0129' 51 .BYTE 'Q'
012A' 54 .BYTE 'T'
012B' 57 .BYTE 'W'
012C' 5A .BYTE 'Z'
012D' 27 .BYTE 27H
012E' 2E .BYTE '.'
012F' 40 .BYTE '@'
0130' 2C .BYTE ','
0131' 22 .BYTE 22H
0132' 3B .BYTE ';'
0133' 25 .BYTE '%'
; TABLE THE THIRD
0134'
KTBL3:
0134' 0F .BYTE 0FH ; THIRD SHIFT KEY COLOR
0135' 0D .BYTE CR
0136' 5C .BYTE 5CH
0137' 00 .BYTE 0
0138' 5D .BYTE 5DH
0139' 43 .BYTE 'C'
013A' 46 .BYTE 'F'
013B' 49 .BYTE 'I'
013C' 4C .BYTE 'L'
013D' 4F .BYTE 'O'
013E' 52 .BYTE 'R'
013F' 55 .BYTE 'U'
0140' 58 .BYTE 'X'
0141' 21 .BYTE '!'
0142' 61 .BYTE 61H
0143' 60 .BYTE 60H
0144' 2A .BYTE '*'
0145' 3F .BYTE '?'
0146' 3E .BYTE '>'
0147' 29 .BYTE ')'
0148' 3A .BYTE ':'
; TOKEN KEY

```

```
0149'          KTBL4:
0149'  77      .BYTE  77H      ; WORDS KEY COLOR
014A'  67      .BYTE  NLLN
014B'  66      .BYTE  EDKEY
014C'  6A      .BYTE  6AH
014D'  68      .BYTE  68H
014E'  72      .BYTE  72H
014F'  77      .BYTE  77H
0150'  75      .BYTE  75H
0151'  6B      .BYTE  6BH
0152'  6F      .BYTE  6FH
0153'  70      .BYTE  70H
0154'  76      .BYTE  76H
0155'  6D      .BYTE  6DH
0156'  69      .BYTE  69H
0157'  6C      .BYTE  6CH
0158'  71      .BYTE  71H
0159'  6E      .BYTE  6EH
015A'  66      .BYTE  EDKEY
015B'  73      .BYTE  73H
015C'  01      .BYTE  1
015D'  74      .BYTE  74H
          .END
```

MUSIC -

```

                                .PREL
                                .IDENT MUSIC
                                .INSERT HVG.LIB
                                .INSERT MACRO.LIB
                                .INSERT MUSIC.LIB
                                .INSERT BBEQU.ASM
0000'                                .RELOC
                                ; DEVICE VARIABLE TO PLAY NOTE WITHOUT PRINTING
0000'    CF    PUTMU:: TSTC    '=' ,LQWHAT[    RST    1
0001'    3D    +    .BYTE    '='
0002'    06    +    .BYTE    LQWHAT-.-1
                                +]
0003'    D7                                RST    RSTEXP
0004'    7D                                MOV    A,L
0005'    CD 000C'    CALL    PNOTE
0008'    F7                                RST    RSTFIN
0009'    C3 0000:04    LQWHAT: JMP    QWHAT#
                                ; SUBROUTINE TO PLAY A NOTE
000C'    E5    PNOTE:: PUSH    H
000D'    D5                                PUSH    D
000E'    F5                                PUSH    PSW
000F'    67                                MOV    H,A
                                ; WAIT FOR PREVIOUS PARAMETERS TO BE EATEN
0010'    3A 0000:05    PRWAIT: LDA    NEWTMR#
0013'    A7                                ANA    A
0014'    20FA    JRNZ    PRWAIT ; LOOP
0016'    7C                                MOV    A,H
0017'    FE63    CPI    63H ; DIVIDE?
0019'    282E    JRZ    PNOTDV
001B'    FE62    CPI    62H ; MULTIPLY?
001D'    2831    JRZ    PNOTML
001F'    FE2B    CPI    '+'
0021'    2831    JRZ    PNOTPL
0023'    FE2D    CPI    '-'
0025'    2830    JRZ    PNOTMN
0027'    FE30    CPI    '0'
0029'    2830    JRZ    PNOTZ
002B'    21 007C'    LXI    H,NOTES-CR
002E'    FF    HOOKIN: INDEXBLINTP%[.IFE .INTP.,[RST 7]]
002F'    5C    +.BYTE    92+0]
0030'    3C                                INR    A ; CLICK?
0031'    280E    JRZ    PNOTCL
0033'    3C                                INR    A ; AINSWORTH NUMBA?
0034'    2831    JRZ    PNOTNO
0036'    3D                                DCR    A
0037'    3D                                DCR    A
0038'    32 0000:06    STA    MUZTON#
003B'    3A 0000:07    LDA    DEVTEM#
003E'    32 0000:05    STA    NEWTMR#
0041'    AF    PNOTCL: XRA    A
0042'    32 0000:08    PSHARP: STA    SHARPF#
0045'    F1    PNOTC1: POP    PSW
0046'    D1                                POP    D
0047'    E1                                POP    H
0048'    C9                                RET

```

MUSIC -

```

0049' 3E8F      PNOTDV: MVI      A,0A1
004B' 32 0000:09 PNOTD1: STA      MUZMO#
004E' 18F5              JMPR      PNOTC1
0050' 3E23      PNOTML: MVI      A,0A3
0052' 18F7              JMPR      PNOTD1
0054' 3E01      PNOTPL: MVI      A,1
0056' 11              .BYTE     11H      ; OPCODE TO MUNCH NEXT T
                        WD BYTES INTO DE
0057' 3E02      PNOTMN: MVI      A,2
0059' 18E7              JMPR      PSHARP
005B' 21 0000:0A PNOTZ:  LXI      H,MUZTMR#
005E' 3A 0000:07      LDA      DEVTEM#
0061' F3              DI
0062' 86              ADD      M
0063' 77              MOV      M,A
0064' FB              EI
0065' 18DE              JMPR      PNOTC1
                        ; PLAY AINSWORTH NOTE
0067' 3A 0000:08 PNOTNO: LDA      SHARPF# ; RESTORE CHARACTER
006A' FF              INDEXW  1[INTP%[.IFE .INTP.,[RST 7]]
006B' 5B      +.BYTE  90+1]
006C' 0075'      .WORD     DICKY
006E' EB              XCHG
006F' F1              POP      PSW
0070' F5              PUSH     PSW
0071' D631          SUI      '1'
0073' 18B9              JMPR      HOOKIN
                        ; MUSIC TABLES
0075' 0082'      DICKY:  .WORD     MUZNOR
0077' 0094'      .WORD     MUZMAX
0079' 007B'      .WORD     MUZMIN
007B' MUZMIN:
007B' 64              .BYTE     B1
007C' 59              .BYTE     CS2
007D' 4F              .BYTE     DS2
007E' 4A              .BYTE     E2
007F' 42              .BYTE     FS2
0080' 3B              .BYTE     GS2
0081' 34              .BYTE     AS2
0082' MUZNOR:
0082' 5E              .BYTE     C2
0083' 54              .BYTE     D2
0084' 4A              .BYTE     E2
0085' 46              .BYTE     F2
0086' 3E              .BYTE     G2
0087' 37              .BYTE     A2
0088' 31              .BYTE     B2
                        ; TONE GENERATION TABLE - ORDERED BY ASCII CHARA
                        CTER
00FF      CLICK  ==      OFFH
00FE      NUMBA  ==      OFEH
0089' FF      NOTES: .BYTE  CLICK  ; CR
                        ; IRRELEVANT STUFF TO EAT TABLE SPACE
008A' 0000      0000      WORD  0,0,0,0,0
0094' MUZMAX:

```

MUSIC -

0094	59	.BYTE	CS2
0095	4F	.BYTE	DS2
0096	46	.BYTE	F2
0097	42	.BYTE	FS2
0098	3B	.BYTE	GS2
0099	34	.BYTE	AS2
009A	2E	.BYTE	C3
; AND THE TABLE CONTINUES			
009B	FF	.BYTE	CLICK
009C	00	.BYTE	0
009D	E1	.BYTE	A0
009E	D4	.BYTE	AS0
009F	C8	.BYTE	B0
00A0	BD	.BYTE	C1
00A1	B2	.BYTE	CS1
00A2	A8	.BYTE	D1
00A3	9F	.BYTE	DS1
00A4	96	.BYTE	E1
00A5	8D	.BYTE	F1
00A6	85	.BYTE	FS1
00A7	FF	.BYTE	CLICK
00A8	77	.BYTE	GS1
00A9	FF	.BYTE	CLICK
00AA	6A	.BYTE	AS1
00AB	64	.BYTE	B1
00AC	FF	.BYTE	CLICK
00AD	FE	.BYTE	NUMBA
00AE	FE	.BYTE	NUMBA
00AF	FE	.BYTE	NUMBA
00B0	FE	.BYTE	NUMBA
00B1	FE	.BYTE	NUMBA
00B2	FE	.BYTE	NUMBA
00B3	FE	.BYTE	NUMBA
00B4	2E	.BYTE	C3
00B5	2C	.BYTE	CS3
00B6	29	.BYTE	D3
00B7	27	.BYTE	DS3
00B8	25	.BYTE	E3
00B9	22	.BYTE	F3
00BA	20	.BYTE	FS3
00BB	1F	.BYTE	G3
00BC	1D	.BYTE	GS3
00BD	1B	.BYTE	A3
00BE	1A	.BYTE	AS3
00BF	18	.BYTE	B3
00C0	17	.BYTE	C4
00C1	15	.BYTE	CS4
00C2	14	.BYTE	D4
00C3	13	.BYTE	DS4
00C4	12	.BYTE	E4
00C5	11	.BYTE	F4
00C6	10	.BYTE	FS4
00C7	0F	.BYTE	G4
00C8	0E	.BYTE	GS4
00C9	0D	.BYTE	A4

MUSIC -

00CA'	0B	.BYTE	C5
00CB'	0A	.BYTE	CS5
00CC'	09	.BYTE	DS5
00CD'	08	.BYTE	F5
00CE'	07	.BYTE	G5
00CF'	06	.BYTE	A5
00D0'	05	.BYTE	C6
00D1'	04	.BYTE	DS6
00D2'	03	.BYTE	G6
00D3'	02	.BYTE	C7
00D4'	01	.BYTE	G7
00D5'	64	.BYTE	B1
00D6'	5E	.BYTE	C2
00D7'	59	.BYTE	CS2
00D8'	54	.BYTE	D2
00D9'	4F	.BYTE	DS2
00DA'	4A	.BYTE	E2
00DB'	46	.BYTE	F2
00DC'	42	.BYTE	FS2
00DD'	3E	.BYTE	G2
00DE'	FF	.BYTE	CLICK
00DF'	FF	.BYTE	CLICK
		.END	

TAPE -

```

        .PREL
        .IDENT TAPE
        .INSERT HVG.LIB
        .INSERT MACRO.LIB
        .INSERT BBEQU.ASM
0000'   .RELOC
        ; COMMAND TO ESTABLISH TAPE UNIT AS INPUT DEVICE

0000'   CD 0000:04   TINPUT:: CALL   TINPES#
0003'   F7          RST     RSTFIN  ; GO HOME
        ; COMMAND TO LIST STUFF ON CRT

0004'   CD 0000:04   TLIST:: CALL   TINPES#
0007'   CD 0000:05   TLIST1: CALL   CHKIO#  ; GET CHARACTER
000A'   DF          RST     RSTOCH  ; PRINT IT
000B'   18FA       JMPR    TLIST1  ; FOREVER      1
        ; COMMAND TO ESTABLISH PRINTING THRU THE TAPE IN
        ; TERFACE

000D'   3E06       TPRINT:: MVI    A,6    ; SET PRINT FLAG
000F'   01        .BYTE   01H    ; LXI    B,#### OPCODE TO
        ; EFFECT SKIP
        ; COMMAND TO ESTABLISH OUTPUT TO TAPE

0010'   3E02       TOUTPU:: MVI    A,2    ; SET OUTPUT FLAG
0012'   32 0000:06   STA     TAPEST#
0015'   F7          RST     RSTFIN
        ; COMMAND TO LOAD 128 BYTE BOOTSTRAP FROM TAPE

0016'   CD 0000:04   TLOAD:: CALL   TINPES# ; START TAPE READING
0019'   21 4000     LXI     H,NORMEM
001C'   E5         PUSH    H
001D'   E5         TLOAD1: PUSH    H
001E'   CD 0000:05   CALL   CHKIO#
0021'   E1         POP     H
0022'   77         MOV     M,A
0023'   23         INX     H
0024'   CB7D       BIT     7,L
0026'   28F5       JRZ    TLOAD1
0028'   C9         RET
        ; ENTER VIA RETURNING
0029'   TOUTCK::
0029'   3A 0000:06   LDA     TAPEST#
002C'   FE06       CPI     6
002E'   282D       JRZ    VDMFRG
0030'   E602       TURDC1: ANI    2    ; WRITE TO TAPE WANTED?
0032'   C8         RZ
0033'   4A         TURDC2: MOV     C,D
0034'   CB01       RLCR    C
        ; YEP
        ; WRITE START BIT

0036'   DB12       TAPCHO: IN     TAPEIO  ; INPUT FF STATE
0038'   E602       ANI     2    ; WAIT FOR IT TO GO NONZ
        ERO
003A'   28FA       JRZ     TAPCHO  ; MEANING START BIT WRIT
        TEN
003C'   060A       MVI     B,10   ; B = # OF BITS TO WRITE

        ; WAIT ABOUT 1.8 MILLISECONDS TO MOVE BEYOND CHA
        ; NGE WINDOW

```


TAPE -

```

003E' 3EC0      TAPCH1: MVI      A,192
0040' 3D       TAPCH2: DCR      A
0041' 20FD      JRNZ      TAPCH2
; ARE WE DONE WRITING?
0043' 05       DCR      B
0044' C8       RZ
; (POSITION IS BIT 1 )
0045' DB12     IN       TAPEIO ; KEEP READING INPUT
0047' 5F       MOV      E,A
0048' DB12     TAPCH3: IN      TAPEIO ; UNTIL IT TOGGLES
004A' AB       XRA      E
004B' E602     ANI      2
004D' 28F9     JRZ      TAPCH3
004F' 7B       MOV      A,E ; E = WHAT WE GOT NOW
0050' A9       XRA      C ; C = WHAT WE WANT
0051' E602     ANI      2 ; DID WE 'GET IT?'
0053' 2802     JRZ      TAPCH4 ; JUMP IF SO
0055' DB12     IN       TAPEIO ; NO - READ AGAIN TO TOG
GLE
0057' CBC9     TAPCH4: SET      1,C ; SET EACH BIT WRITTEN T
O 1
0059' CB09     RRCR      C ; SHIFT INTO NEXT POSITI
ON
005B' 18E1     JMPR      TAPCH1 ; SO STOP BIT WILL GO OU
T
; MODE 6 - TOKEN COMING THRU?
005D' 7A       VDMFRG: MOV      A,D
005E' FE68     CPI      68H
0060' 38D1     JRC      TURDC2 ; NO - TWEEDLE IT TO TAP
E
0062' C9       RET
.END

```

SUBS -

```

                                .PREL
                                .IDENT  SUBS
                                .INSERT BBEQU.ASM
0000' 7C          CKHLDE:: MOV    A,H
0001' AA          XRA     D
0002' F2 0006'   JP      COMP
0005' EB          XCHG

                                ; ...
0006' 7C          COMP:: MOV    A,H
0007' BA          CMP     D
0008' C0          RNZ
0009' 7D          MOV    A,L
000A' BB          CMP     E
000B' C9          RET

                                ; SUBROUTINE TO RETURN ZERO STATUS IF CHARACTER
                                ; IN A IS NL OR
                                ; '/'
000C' FE3B       ATNL:: CPI     '/'      ; CHECK FOR CONTINUATION

000E' C8          RZ
000F' FE0D       CPI     CR      ; AND FOR CR
0011' C9          RET
0012' CD 0000:04  IGNBLK:: CALL  LDE#
0015' FE20       CPI     '/'
0017' C0          RNZ
0018' 13         INX     D
0019' 18F7       JMPR    IGNBLK

                                ; TSTCH AND TSTNUM
001B' E3         TSTCH:: XTHL
001C' E7         RST     RSTIGN
001D' BE         CMP     M
001E' 23         INX     H
001F' 2807       JRZ     TC1
0021' C5         PUSH   B
0022' 4E         MOV    C,M
0023' 0600       MVI    B,0
0025' 09         DAD    B
0026' C1         POP    B
0027' 1B         DCX   D
0028' 13         TC1:   INX   D
0029' 23         INX   H
002A' E3         XTHL
002B' C9         RET
002C' 21 0000   TSTNUM:: LXI   H,0
002F' 44         MOV    B,H
0030' E7         RST   RSTIGN
0031' FE30       TN1:   CPI   '0'
0033' D8         RC
0034' FE3A       CPI   3AH
0036' D0         RNC
0037' 3EF0       MVI   A,0F0H
0039' A4         ANA   H
003A' C2 0000:05  JNZ   QHOW#
003D' 04         INR   B
003E' C5         PUSH  B

```

SUBS -

003F'	44	MOV	B,H
0040'	4D	MOV	C,L
0041'	29	DAD	H
0042'	29	DAD	H
0043'	09	DAD	B
0044'	29	DAD	H
0045'	CD 0000:04	CALL	LDE#
0048'	13	INX	D
0049'	E60F	ANI	00FH
004B'	85	ADD	L
004C'	6F	MOV	L,A
004D'	3E00	MVI	A,0
004F'	8C	ADC	H
0050'	67	MOV	H,A
0051'	C1	POP	B
0052'	CD 0000:04	CALL	LDE#
0055'	F2 0031'	JP	TN1
0058'	C3 0000:05	JMP	QH0W#
		.END	

PRTSUB -

```

                .PREL
                .IDENT PRTSUBS
                .INSERT BBEGU.ASM
; PRTSTG, QTSTG, PRTNUM, PRTLN
0000' 97          PRTSTG:: SUB      A
0001' 47          PS1:   MOV      B,A
0002' CD 0000:04  PS2:   CALL     LDE#
0005' 13          INX      D
0006' B8          CMP      B
0007' C8          RZ
0008' DF          RST      RSTOCH
0009' FE0D       CPI      CR
000B' 20F5       JRNZ     PS2
000D' C9          RET
000E' CF          QTSTG:: TSTC   "'",QT3[      RST      1
000F' 22          +      .BYTE   "'"
0010' 0E          +      .BYTE   QT3--1
                +]
0011' 3E22       MVI      A,'"
0013' CD 0001'   QT1:   CALL     PS1
0016' FE0D       QT2:   CPI      CR
0018' E1          POP      H
0019' CA 0000:05 JZ      RUNNXL#
001C' 23          INX      H
001D' 23          INX      H
001E' E9          PCHL
001F' CF          QT3:   TSTCC   027H,QT5[      RST      1
0020' 27          +      .BYTE   027H
0021' 04          +      .BYTE   QT5--1
                +]
0022' 3E27       MVI      A,027H
0024' 18ED       JMPR     QT1
0026' C9          QT5:   RET
0027' 7B          PRTCHS:: MOV    A,E
0028' B8          CMP      B
0029' C8          RZ
002A' CD 0000:04 CALL     LDE#
002D' DF          RST      RSTOCH
002E' 13          INX      D
002F' 18F6       JMPR     PRTCHS
0031'             PRTNUM  ==:   .
0031' 0600       PN3:   MVI      B,0
0033' CD 0000:06 CALL     CHKSGN#
0036' F2 003C'   JP      PN4
0039' 062D       MVI      B,'-'
003B' 0D          DCR      C
003C' D5          PN4:   PUSH     D
003D' 11 000A   LXI      D,10
0040' D5          PUSH     D
0041' 0D          DCR      C
0042' C5          PUSH     B
0043' CD 0000:07 PN5:   CALL     DIVIDE#
0046' 78          MOV      A,B
0047' B1          ORA      C
0048' CA 0052'   JZ      PN6

```

PRTSUB -

004B✓	E3		XTHL	
004C✓	2D		DCR	L
004D✓	E5		PUSH	H
004E✓	60		MOV	H,B
004F✓	69		MOV	L,C
0050✓	18F1		JMPR	PN5
0052✓	C1	PN6:	POP	B
0053✓	0D	PN7:	DCR	C
0054✓	79		MOV	A,C
0055✓	B7		ORA	A
0056✓	FA 005E✓		JM	PN8
0059✓	3E20		MVI	A,✓
005B✓	DF		RST	RSTOCH
005C✓	18F5		JMPR	PN7
005E✓	78	PN8:	MOV	A,B
005F✓	B7		ORA	A
0060✓	C4 0000:08		CNZ	OUTCH#
0063✓	5D		MOV	E,L
0064✓	7B	PN9:	MOV	A,E
0065✓	FE0A		CPI	10
0067✓	D1		POP	D
0068✓	C8		RZ	
0069✓	C630		ADI	'0'
006B✓	DF		RST	RSTOCH
006C✓	18F6		JMPR	PN9
006E✓	CD 0000:04	PRTLN::	CALL	LDE#
0071✓	6F		MOV	L,A
0072✓	13		INX	D
0073✓	CD 0000:04		CALL	LDE#
0076✓	67		MOV	H,A
0077✓	13		INX	D
0078✓	0E04		MVI	C,4
007A✓	CD 0031✓		CALL	PRTNUM
007D✓	3E20		MVI	A,✓
007F✓	DF		RST	RSTOCH
0080✓	C9		RET	
			.END	

ERROR -

```

                                .PREL
                                .IDENT  ERROR
                                .INSERT MACRO.LIB
                                .INSERT BBEQU.ASM
0000' 574841543F  WHAT:  .ASCII  'WHAT?'
0005' 0D                                .BYTE  CR
0006' 484F573F  HOW:   .ASCII  'HOW?'
000A' 0D                                .BYTE  CR
000B' 534F525259  SORRY: .ASCII  'SORRY'
0010' 0D                                .BYTE  CR
0011' D5                                QWHAT:: PUSH  D
0012' 11 0000'  AWHAT:: LXI  D,WHAT
0015' CD 0000:04  ERROR:  CALL  CRLF#
0018' CD 0000:05                CALL  PRTSTG#
001B' 2A 0000:06                LHL  CURRNT#
001E' E5                                PUSH  H
001F' EB                                XCHG
0020' CD 0000:07                CALL  LDE#
0023' 67                                MOV   H,A
0024' 13                                INX  D
0025' CD 0000:07                CALL  LDE#
0028' B4                                ORA  H
0029' EB                                XCHG
002A' D1                                POP  D
002B' CA 0000:08                JZ   TELL#
002E' EB                                XCHG
002F' CD 0000:07                CALL  LDE#
0032' EB                                XCHG
0033' B7                                ORA  A
0034' FA 0000:09                JM  INPERR#
0037' CD 0000:0A                CALL  PRTLN#
003A' C1                                POP  B
003B' 41                                MOV  B,C
003C' CD 0000:0B                CALL  PRTCHS#
003F' 3E3F                        MVI  A,'?'
0041' DF                        RST  RSTOCH
0042' CD 0000:05                CALL  PRTSTG#
0045' C3 0000:08                JMP  TELL#
0048' D5                                QSORRY:: PUSH  D
0049' 11 000B'  ASORRY:: LXI  D,SORRY
004C' 18C7                        JMPR ERROR
004E' D5                                QHOW::  PUSH  D
004F' 11 0006'  AHOW::  LXI  D,HOW
0052' C3 0015'  JMP  ERROR
                                .END

```