```
*****************************************************************
*                                                               *
*                    M U N C H - M A N  DISK VERSION            *
*                                                               *
*          BY:               JIM  DRAMIS                        *
*          DATE:             01/25/82                           *
*          FILE NAME:        MUNCHRAM                           *
*                                                               *
*                                                               *
*****************************************************************
        DEF   SFIRST, SLAST, SLOAD
SLOAD
SFIRST  B     @START
H00     BYTE  >00
H01     BYTE  >01
H02     BYTE  >02
H03     BYTE  >03
H04     BYTE  >04
H05     BYTE  >05
H06     BYTE  >06
H07     BYTE  >07
H08     BYTE  >08
H09     BYTE  >09
H0A     BYTE  >0A
H0B     BYTE  >0B
H0C     BYTE  >0C
H0D     BYTE  >0D
H0E     BYTE  >0E
H0F     BYTE  >0F
SCAN    EQU   >000E           ADDRESS OF SCAN ROUTINE
CPURAM  EQU   >8300           BEGINNING ADDRES OF CPU RAM
GPLWS   EQU   >83E0           GPL WORKSPACE
VDPRD   EQU   >8800           VDP READ DATA WINDOW ADDRESS
VDPWD   EQU   >8C00           VDP WRITE DATA WINDOW ADDRESS
        EVEN
H0000   DATA  >0000
H0001   DATA  >0001
H0002   DATA  >0002
H0003   DATA  >0003
H0004   DATA  >0004
H0005   DATA  >0005
H0006   DATA  >0006
H000A   DATA  >000A
H000C   DATA  >000C
H000D   DATA  >000D
H0010   DATA  >0010
H0011   DATA  >0011
H0017   DATA  >0017
H0018   DATA  >0018
H0020   DATA  >0020
H0031   DATA  >0031
H0032   DATA  >0032
H003C   DATA  >003C
H0040   DATA  >0040
H0046   DATA  >0046
H0060   DATA  >0060
H0064   DATA  >0064
H0070   DATA  >0070
H0090   DATA  >0090
H0098   DATA  >0098
H00A0   DATA  >00A0
H00B0   DATA  >00B0
```

```
H00C8   DATA >00C8
H0100   DATA >0100
H0190   DATA >0190
H01F4   DATA >01F4
H0200   DATA >0200
H0320   DATA >0320
H0700   DATA >0700
HFF00   DATA >FF00
H00F8   DATA >00F8      248 DOTS IN THE MAZE
H00FF   DATA >00FF
H2710   DATA >2710      10,000
H3230   DATA >3230      "20" ASCII SCREEN LEVEL
H4F4F   DATA >4F4F      CHAIN COLORS AT TITLE SCREEN TIME
H5F70   DATA >5F70      PIT LOCATIONS
H5F78   DATA >5F78      PIT LOCATIONS
H5F80   DATA >5F80      PIT LOCATIONS
H5F88   DATA >5F88      PIT LOCATIONS
HB1B1   DATA >B1B1      HOONO BOX CHARACTERS
HB2B2   DATA >B2B2      HOONO BOX CHARACTERS
HEA60   DATA >EA60      60,000 TIME DELAY FOR MONSTER OUT OF BOXES
CENTER  BYTE >B2,>B3,>B3,>B3,>B3,>B1    CENTER OF MAZE
H10     BYTE >10
H13     BYTE >13            SCREEN LEVEL #19
H14     BYTE >14            SCREEN LEVEL #20
H20     BYTE >20            DELAY PARAMETER
H23     BYTE >23            "#"
H2A     BYTE >2A            "*"
H30     BYTE >30            ASCII ZERO
H31     BYTE >31            ASCII ONE
H33     BYTE >33            ASCII THREE
H39     BYTE >39            ASCII NINE(HIGHEST DIGIT)
H3C     BYTE >3C            SOUND CHIMES TIME FACTOR
H40     BYTE >40            DELAY FOR "ARE YOU READY?" TIME
H42     BYTE >42            TIME-OUT FOR CHIME SOUNDING
H98     BYTE >98            MMAN SAB CHARACTER INIT
HC0     BYTE >C0            MONSTER SPLAT
HC5     BYTE >C5            CHAIN CHARACTER
HC3     BYTE >C3            CHAIN CHARACTER
HC6     BYTE >C6            CHAIN CHARACTER
HCC     BYTE >CC            CHAIN CHARACTER
HCA     BYTE >CA            CHAIN CHARACTER
HC9     BYTE >C9            CHAIN CHARACTER
HD0     BYTE >D0            SPRITE DELETE FOR YPT
HE0     BYTE >E0            TURN ON SCREEN
H81     BYTE >81            WRITE TO VPD REG1
H87     BYTE >87            WRITE TO VDP REG7
HFF     BYTE >FF            FOR JOYSTICK CHECK

*****************************************************************
* COLOR TABLE                                                   *
*****************************************************************
CLEAR   EQU  H00
BLACK   EQU  H01
MGREEN  EQU  H02
LGREEN  EQU  H03
DBLUE   EQU  H04
LBLUE   EQU  H05
DRED    EQU  H06
CYAN    EQU  H07
MRED    EQU  H08
LRED    EQU  H09
DYELL   EQU  H0A
```

```
LYELL   EQU   H0B
DGREEN  EQU   H0C
MAGENT  EQU   H0D
GRAY    EQU   H0E
WHITE   EQU   H0F

*****************************************
*  SOUND LIST AT HIGH VDP(>3000 - )  *
*****************************************
SNDEAT  EQU   >3000          MONSTER EATEN SOUND
SNDENZ  EQU   >3014          ENERGIZER EATEN SOUND
SNDREW  EQU   >3025          BELL REWARD SOUND
SNDWIN  EQU   >3042          MUNCH-MAN WINS SOUND
SNDEND  EQU   >3082          MAN EATEN SOUND
SNDOFF  EQU   >310B          ALL SOUNDS OFF
SNDMUN  EQU   >3111          MUNCH SOUND
SNDCHM  EQU   >3500          START CHIME SOUND
SNDCH1  EQU   >3538          END CHIME SOUND
SNDTIL  EQU   >3543          START CRUNCH SOUND
SNDMOV  EQU   >356B          START TITLE SCREEN SOUND

**********************
*  CPU RAM EQUATES  *
**********************
TIME    EQU   CPURAM+>00   TIME OUT FACTOR FOR SCAN DELAY ROUTINE
YXLOCP  EQU   CPURAM+>00   YPT,XPT LOCATION OF MMAN
YXLOC0  EQU   CPURAM+>04   YPT,XPT LOCATION OF MONSTER0
YXLOC1  EQU   CPURAM+>08   YPT,XPT LOCATION OF MONSTER1
YXLOC2  EQU   CPURAM+>0C   YPT,XPT LOCATION OF MONSTER2
YXLOC3  EQU   CPURAM+>10   YPT,XPT LOCATION OF MONSTER3
YPTSAV  EQU   CPURAM+>14   YPT LOCATION OF MMAN SAVED
XPTSAV  EQU   CPURAM+>16   XPT LOCATION OF MMAN SAVED
SCRSAV  EQU   CPURAM+>18   CURRENT SCORE SAVED
FLAGPC  EQU   CPURAM+>1A   CHECK-DISTANCE FLAG BETWEEN MONSTER&MMAN
CAPPTS  EQU   CPURAM+>1B   NUMBER OF CAPTURE POINTS(100,200,400,800)
YMONSV  EQU   CPURAM+>1C   YPT LOCATION OF MONSTER SAVED
XMONSV  EQU   CPURAM+>1E   XPT LOCATION OF MONSTER SAVED
SAVRM0  EQU   CPURAM+>20   RANDOM NUMBER SAVED MONSTER0
SAVRM1  EQU   CPURAM+>24   RANDOM NUMBER SAVED MONSTER1
SAVRM2  EQU   CPURAM+>28   RANDOM NUMBER SAVED MONSTER2
SAVRM3  EQU   CPURAM+>2C   RANDOM NUMBER SAVED MONSTER3
DELTA0  EQU   CPURAM+>22   DELTA Y AND X FOR   MONSTER0
DELTA1  EQU   CPURAM+>26   DELTA Y AND X FOR   MONSTER1
DELTA2  EQU   CPURAM+>2A   DELTA Y AND X FOR   MONSTER2
DELTA3  EQU   CPURAM+>2E   DELTA Y AND X FOR   MONSTER3
INCMAN  EQU   CPURAM+>30   YPT,XPT INCREMENT OF MMAN
SAVINC  EQU   CPURAM+>32   YPT,XPT INCREMENT OF MMAN SAVED(0-3)
DOTCNT  EQU   CPURAM+>34   NUMBER OF DOTS EATEN BY MMAN
WINFLG  EQU   CPURAM+>36   WIN GAME FLAG
SCRNUM  EQU   CPURAM+>37   NUMBER OF SCREENS COMPLETED
WAITPC  EQU   CPURAM+>38   STALL COUNTER TO MOVE MMAN
WAITMN  EQU   CPURAM+>3A   STALL COUNTER TO MOVE MONSTERS
STALPC  EQU   CPURAM+>3C   STALL COUNTER SAVED AFTER MMAN ENERGIZED
ENGSAV  EQU   CPURAM+>3E   TIME OUT FACTOR FOR ENERGIZED MMAN
INCR0   EQU   CPURAM+>40   YPT,XPT INCREMENT OF MONSTER0 SAVED
INCR1   EQU   CPURAM+>44   YPT,XPT INCREMENT OF MONSTER1 SAVED
INCR2   EQU   CPURAM+>48   YPT,XPT INCREMENT OF MONSTER2 SAVED
INCR3   EQU   CPURAM+>4C   YPT,XPT INCREMENT OF MONSTER3 SAVED
PROB00  EQU   CPURAM+>42   PROBABILITY OF CHASE BREAK-MONSTER0
PROB01  EQU   CPURAM+>46   PROBABILITY OF CHASE BREAK-MONSTER1
PROB02  EQU   CPURAM+>4A   PROBABILITY OF CHASE BREAK-MONSTER2
PROB03  EQU   CPURAM+>4E   PROBABILITY OF CHASE BREAK-MONSTER3
```

```
SAVR11  EQU  CPURAM+>50    SAVE RETURN ADDRESS FROM R11
RAND16  EQU  CPURAM+>52    16 BIT RANDOM NUMBER
COUNT1  EQU  CPURAM+>54    NUMBER OF MMAN REMAINING
INDX    EQU  CPURAM+>55    TEMP VARIABLE
SAVWPC  EQU  CPURAM+>56    SAVE WAIT AMOUNT FOR MMAN
SAVWMN  EQU  CPURAM+>58    SAVE WAIT AMOUNT FOR MONSTERS
TITLFG  EQU  CPURAM+>5A    TITLE SCREEN UP FLAG
SCRNUB  EQU  CPURAM+>5B    SCREEN NUMBER (0 - 19)
POINT   EQU  CPURAM+>5C    REWARD MSG POINTER
SCRTCH  EQU  CPURAM+>60    SCRATCH AREA
SAVR10  EQU  CPURAM+>60    SAVE REG10 AREA FOR SCORE PASSED
DIGTHB  EQU  CPURAM+>62    HIGH BYTE FOR DIGIT
DIGTLB  EQU  CPURAM+>63    LOW BYTE FOR DIGIT
CARYHB  EQU  CPURAM+>64    HIGH BYTE CARRY DIGIT
CARYLB  EQU  CPURAM+>65    LOW BYTE CARRY DIGIT
FLAG    EQU  CPURAM+>66    FLAG FOR TEST BIT
FLAGZZ  EQU  CPURAM+>67    FLAG FOR HIT ON ENERGIZER
ENGCNT  EQU  CPURAM+>68    COUNT DOWN ENERGIZED MMAN
ENGPTN  EQU  CPURAM+>6A    CURRENT SAVED POINTER FOR ENERGIZER PATTERN
CHGCNT  EQU  CPURAM+>6C    COUNT DOWN FOR ENERGIZER PATTERN CHANGE
FLHCNT  EQU  CPURAM+>6E    COUNTER FOR WARNING FLASH OF MONSTERS
KEYBRD  EQU  CPURAM+>74    KEY BOARD TO SCAN FROM
KEY     EQU  CPURAM+>75    KEY CODE RETURNED
JOYY    EQU  CPURAM+>76
JOYX    EQU  CPURAM+>77
RANDOM  EQU  CPURAM+>78
TIMER   EQU  CPURAM+>79
MOTION  EQU  CPURAM+>7A
STATUS  EQU  CPURAM+>7C
RA      EQU  CPURAM+>80    TEN RETURN ADDRESSES
BESTSC  EQU  CPURAM+>90    BEST SCORE POINTER
YOURSC  EQU  CPURAM+>94    YOUR SCORE POINTER
DEMOFG  EQU  CPURAM+>99    DEMO FLAG(0 OR 1)
ESCAPE  EQU  CPURAM+>9D    ESCAPE FROM DEMO FLAG(0 OR 1)
YXPIT0  EQU  CPURAM+>92    PIT LOCATION OF MONSTER0
YXPIT1  EQU  CPURAM+>96    PIT LOCATION OF MONSTER1
SNDFLG  EQU  CPURAM+>98    SOUND CHIMES FLAG COUNTER
YXPIT2  EQU  CPURAM+>9A    PIT LOCATION OF MONSTER2
LIMIT   EQU  CPURAM+>9C    LIMIT FOR TIME OUT FOR SCAN ROUTINE
YXPIT3  EQU  CPURAM+>9E    PIT LOCATION OF MONSTER3
CLRSCN  EQU  CPURAM+>D6    SCREEN TIME-OUT COUNTER
*****************************************************************
* WORKING SPACE REGISTERS ( START AT >83A0 )                    *
*****************************************************************
REG0    EQU  >83A0         RETURN TO GROM FLAG
MYWS    EQU  CPURAM+>A0
VDPADD  EQU  R0            VDP ADDRESS TO READ/WRITE TO
VADDLB  EQU  MYWS+1        VDP ADDRESS LOWER BYTE
Y       EQU  VDPADD
YLB     EQU  VADDLB
RLOC    EQU  R1            POINTER TO BUFFER TO BE READ INTO
RCOUNT  EQU  R2            # OF BYTES IN BUFFER TO READ
WCOUNT  EQU  R2            # OF BYTES IN BUFFER TO WRITE
WLOC    EQU  R3            POINTER TO BUFFER TO BE WRITTEN
TEMPR4  EQU  R4            USED FOR TEMPORARY VARIBLES
REG4HB  EQU  MYWS+8        FOR COINCIDENCE CHECKING
REG4LB  EQU  MYWS+9        FOR COINCIDENCE CHECKING
COUNT   EQU  R5            COUNTER FOR LOOP
TEMP    EQU  R6            TEMPORARY ADDRESS
X       EQU  R7
XLB     EQU  MYWS+15
PTRNNO  EQU  R8            PATTERN NUMBER
```

```
PTNOLB EQU  MYWS+17
XREM   EQU  R9
TEMPRA EQU  R10
REGALB EQU  MYWS+21       FOR COINCIDENCE CHECKING
YXPOS  EQU  R12
MON    EQU  R13
TEMR14 EQU  R14
RAND   EQU  R15
YPOS   EQU  MYWS+24
XPOS   EQU  MYWS+25

*********************
*  CPURAM BUFFERS   *
*********************
BUFF1  BSS  8

*****************
*  TABLE AREA   *
*****************
BITMSK BYTE >08,>04,>02,>01                    U,D,L,R FOR BIT CHECKING
TABLE1 DATA >FF00,>0100,>00FF,>0001            SAVED INCREMENT U,D,L,R
TABLE2 DATA >0082,>009D,>0222,>023D            ENGZS VDP SCREEN LOCATIONS
TABLE3 DATA >006C,>007C,>008C,>009C,>00AC      DIFFERENT ENGZ PATTERNS
TABLE4 BYTE >39,>3A,>3B,>3C                    EXTRA MMAN LOCATIONS
TABLE5 DATA @DNKEY                             KEYBOARD ONE SCAN TABLE
       DATA @CHKJOY
       DATA @LFKEY
       DATA @RTKEY
       DATA @CHKJOY
       DATA @UPKEY
TABLE6 DATA @DNKEY                             JOYSTICK 1,2 SCAN TABLE
       DATA @LFKEY
       DATA @UPKEY
       DATA @RTKEY
TABLE7 DATA 329,340,457,468                    CAPTURE POINT LOCATIONS
WALL03 BYTE >00,>20,>01                        PIECES OF WALL RESTORED
MSGZER BYTE >30,>30,>30,>30,>30,>30            ZERO FOR SCORE
VBLANK BYTE >07,>07,>07,>07,>07,>07,>07        REPAIR BOTTOM OF MAZE
       BYTE >07,>07,>07,>07,>07,>07,>07
XMUNCH BYTE >2A,>2B,>2C,>2D,>2E                "MUNCH"
XMAN   BYTE >2A,>2F,>29,>28                    "MAN"
MSGMEN BYTE >74,>74,>74,>74                    NUMBER OF MUNCH-MEN REMAINING
MSGPAU TEXT 'PAUSE'
MSGBLK TEXT '                '
READYM TEXT 'ARE YOU READY?'
MSGGAM TEXT 'GAME'
MSGOVE TEXT 'OVER'
MSGYOU TEXT 'YOUR SCORE'
MSGDEM TEXT 'DEMO SCORE'
MSGHI  TEXT 'HIGH SCORE'
MSGTST TEXT 'TEST SCORE'
MSGBEG BYTE >20,>5D
       TEXT 'PRESS ANY KEY TO BEGIN'
       BYTE >5E,>20
MSGRND TEXT 'RND'
       BYTE >5D,>30,>5F,>32,>5E
MSGSCN TEXT 'SCN'
       BYTE >5D,>30,>30,>5F,>31,>39,>5E
MSGMM  TEXT 'MM'
       BYTE >5D,>31,>5F,>39,>5E
MSG100 BYTE >E2,>E0,>E1                        100 POINTS
MSG200 BYTE >E3,>E0,>E1                        200 POINTS
```

```
MSG400 BYTE >E4,>E0,>E1                    400 POINTS
MSG800 BYTE >E5,>E0,>E1                    800 POINTS
COLBLA BYTE >1F,>1F,>1F,>1F,>1F,>1F         BLACK PCT COLOR
COLRED BYTE >8F,>8F,>8F,>8F,>8F            MRED MAZE COLOR
COLTB1 BYTE >1F,>3F,>EF,>EF                RESTORE PCT'S
COLLGN BYTE >13,>13,>13,>13,>13,>13        LGREEN COLOR
COLGRN BYTE >3F                            LGREEN ENERGIZERS
COLMAG BYTE >DF                            MONSTER MAGENTA
WINDT1 DATA >A8A9,>2020,>2020,>B3B3        EYE LEFT
WINDT2 DATA >2020,>A8A9,>2020,>B3B3        EYE CENTER
WINDT3 DATA >2020,>2020,>A8A9,>B3B3        EYE RIGHT
WINDT4 DATA >2020,>A8A9,>2020,>B3B3        EYE CENTER
TABMON DATA >0B80,>0B00,>0C00,>0D00        MONSTER CHARACTER
       DATA >0F40,>0F80,>0FC0              VDP DATA LOCATIONS
TABLE8 DATA 2500,64,330,345,60000
       DATA 2000,56,330,345,60000
       DATA 1500,48,330,345,60000
       DATA 2400,72,320,330,55000
       DATA 1900,64,320,330,55000
       DATA 1400,56,320,330,55000
       DATA 2300,80,310,310,50000
       DATA 1800,72,310,310,50000
       DATA 1300,64,310,310,50000
       DATA 2300,88,310,300,45000
       DATA 1800,80,310,300,45000
       DATA 1300,72,310,300,45000
       DATA 2200,96,300,290,40000
       DATA 1700,88,300,290,40000
       DATA 1200,80,300,290,40000
       DATA 2100,104,280,265,35000
       DATA 1600,96,280,265,35000
       DATA 1200,88,280,265,35000
       DATA 2000,112,270,250,30000
       DATA 1600,104,270,250,30000
TABLE9 DATA 2000,275,250
       DATA 1600,275,250
       DATA 1200,275,250
       DATA 1950,270,245
       DATA 1575,270,245
       DATA 1200,270,245
       DATA 1900,265,240
       DATA 1550,265,240
       DATA 1200,265,240
       DATA 1850,260,230
       DATA 1525,260,230
       DATA 1200,260,230
       DATA 1800,255,225
       DATA 1500,255,225
       DATA 1200,255,225
       DATA 1750,250,215
       DATA 1475,250,215
       DATA 1200,250,215
       DATA 1700,245,205
       DATA 1450,240,200
TABLEA DATA 1600,250,210
       DATA 1400,250,210
       DATA 1200,250,210
       DATA 1500,245,205
       DATA 1300,245,205
       DATA 1200,245,205
       DATA 1400,240,200
       DATA 1300,240,200
```

```
        DATA 1200,240,200
        DATA 1300,235,190
        DATA 1300,235,190
        DATA 1200,235,190
        DATA 1300,230,185
        DATA 1200,230,185
        DATA 1200,230,185
        DATA 1200,225,175
        DATA 1200,225,175
        DATA 1200,225,175
        DATA 1200,220,170
        DATA 1200,215,155
***********************************************
*  SPRITE ATTRIBUTE BLOCK INITIALIZATIONS  *
***********************************************
SABMAN BYTE >5F,>80,>80,>03
SABMNO BYTE >4F,>50,>98,>08
SABMN1 BYTE >6F,>50,>98,>04
SABMN2 BYTE >4F,>A8,>98,>0D
SABMN3 BYTE >6F,>A8,>98,>0A,>D0

*********************************
*  SPRITE DESCRIPTOR BLOCKS  *
*********************************
DMONA   BYTE >62,>B9,>3F,>6A,>7E,>E4,>9D,>46
        BYTE >C7,>A1,>BE,>2A,>3C,>7D,>85,>E3
        BYTE >1E,>C2,>BC,>A9,>BD,>BD,>43,>78
        BYTE >78,>5B,>3D,>EB,>FF,>A4,>CA,>0E
        BYTE >62,>B9,>3F,>6A,>7E,>E4,>9D,>46
        BYTE >C7,>A1,>BE,>2A,>3C,>7D,>85,>E3
        BYTE >1E,>C2,>BC,>A9,>BD,>BD,>43,>78
        BYTE >78,>5B,>3D,>EB,>FF,>A4,>CA,>0E
DMONB   BYTE >18,>24,>3C,>18,>7E,>FF,>7E,>24
        BYTE >18,>24,>3C,>18,>7E,>FF,>7E,>24
        BYTE >00,>18,>24,>3C,>18,>7E,>FF,>7E
        BYTE >00,>18,>24,>3C,>18,>7E,>FF,>7E
        BYTE >18,>24,>3C,>18,>7E,>FF,>7E,>24
        BYTE >18,>24,>3C,>18,>7E,>FF,>7E,>24
        BYTE >18,>24,>3C,>7E,>FF,>7E,>24,>24
        BYTE >18,>24,>3C,>7E,>FF,>7E,>24,>24
DMONC   BYTE >AA,>7F,>DA,>5B,>FE,>43,>FE,>55
        BYTE >AA,>7F,>DA,>5B,>FE,>43,>FE,>55
        BYTE >55,>FE,>5B,>DA,>7F,>E6,>7F,>AA
        BYTE >55,>FE,>5B,>DA,>7F,>E6,>7F,>AA
        BYTE >AA,>7F,>DA,>5B,>FE,>67,>FE,>55
        BYTE >AA,>7F,>DA,>5B,>FE,>67,>FE,>55
        BYTE >55,>FE,>5B,>DA,>7F,>FE,>7F,>AA
        BYTE >55,>FE,>5B,>DA,>7F,>FE,>7F,>AA
DMOND   BYTE >7E,>FF,>C3,>C3,>FF,>FF,>FF,>7E
        BYTE >7E,>FF,>C3,>C3,>FF,>FF,>FF,>7E
        BYTE >00,>7E,>7E,>42,>7E,>7E,>7E,>00
        BYTE >00,>7E,>7E,>42,>7E,>7E,>7E,>00
        BYTE >00,>3C,>7E,>66,>7E,>7E,>3C,>00
        BYTE >00,>3C,>7E,>66,>7E,>7E,>3C,>00
        BYTE >00,>00,>3C,>24,>3C,>3C,>00,>00
        BYTE >00,>00,>3C,>24,>3C,>3C,>00,>00
DMONE   BYTE >7E,>DB,>FF,>A5,>81,>81,>A5,>7E
        BYTE >7E,>DB,>FF,>A5,>81,>81,>A5,>7E
        BYTE >7E,>DB,>FF,>A5,>81,>A5,>7E,>00
        BYTE >7E,>DB,>FF,>A5,>81,>A5,>7E,>00
        BYTE >7E,>DB,>FF,>A5,>A5,>7E,>00,>00
        BYTE >7E,>DB,>FF,>A5,>A5,>7E,>00,>00
```

```
        BYTE >7E,>DB,>FF,>A5,>81,>A5,>7E,>00
        BYTE >7E,>DB,>FF,>A5,>81,>A5,>7E,>00
DMONF   BYTE >3C,>7E,>FF,>00,>55,>00,>2A,>00
        BYTE >3C,>7E,>FF,>08,>20,>40,>04,>00
        BYTE >3C,>7E,>FF,>00,>39,>04,>00,>24
        BYTE >3C,>7E,>FF,>00,>52,>01,>2A,>10
        BYTE >3C,>7E,>FF,>28,>40,>00,>42,>00
        BYTE >3C,>7E,>FF,>00,>39,>04,>00,>24
        BYTE >3C,>7E,>FF,>08,>20,>40,>04,>00
        BYTE >3C,>7E,>FF,>00,>00,>00,>00,>00
DMONG   BYTE >10,>06,>88,>21,>98,>02,>24,>18
        BYTE >08,>26,>44,>83,>24,>52,>06,>10
        BYTE >04,>5A,>81,>B4,>21,>4A,>6C,>08
        BYTE >30,>0C,>53,>C8,>0B,>64,>32,>0C
        BYTE >10,>06,>88,>21,>98,>04,>42,>18
        BYTE >08,>24,>44,>83,>24,>52,>02,>10
        BYTE >04,>5A,>81,>B4,>21,>4A,>6C,>08
        BYTE >30,>08,>53,>C8,>0B,>64,>12,>0C
DMONH   BYTE >00,>00,>18,>3C,>7E,>C3,>81,>81
        BYTE >00,>00,>18,>3C,>FF,>C3,>81,>00
        BYTE >00,>00,>00,>18,>7E,>FF,>81,>00
        BYTE >00,>00,>00,>18,>DB,>FF,>24,>00
        BYTE >00,>00,>81,>99,>DB,>7E,>24,>00
        BYTE >00,>00,>18,>DB,>FF,>24,>00,>00
        BYTE >00,>00,>18,>7E,>FF,>81,>00,>00
        BYTE >00,>00,>18,>3C,>FF,>C3,>81,>00
DMONI   BYTE >18,>18,>18,>18,>18,>18,>18,>18
        BYTE >C0,>60,>30,>18,>18,>0C,>06,>03
        BYTE >00,>00,>00,>FF,>FF,>00,>00,>00
        BYTE >0C,>06,>03,>18,>18,>C0,>60,>30
        BYTE >18,>18,>18,>18,>18,>18,>18,>18
        BYTE >C0,>60,>30,>18,>18,>0C,>06,>03
        BYTE >00,>00,>00,>FF,>FF,>00,>00,>00
        BYTE >0C,>06,>03,>18,>18,>C0,>60,>30
DMONJ   BYTE >FC,>78,>38,>1C,>0C,>04,>04,>08
        BYTE >FC,>7C,>38,>1C,>0C,>06,>02,>04
        BYTE >7C,>78,>1C,>18,>0C,>04,>04,>08
        BYTE >7C,>78,>1C,>1C,>18,>10,>08,>04
        BYTE >7E,>78,>1C,>38,>18,>10,>10,>08
        BYTE >7E,>78,>1C,>1C,>18,>10,>08,>04
        BYTE >7E,>78,>1C,>18,>0C,>04,>04,>08
        BYTE >FC,>7C,>38,>1C,>0C,>06,>02,>04
DMONK   BYTE >C3,>7E,>3C,>18,>18,>3C,>7E,>C3
        BYTE >00,>FF,>3C,>18,>18,>3C,>FF,>00
        BYTE >00,>3C,>FF,>18,>18,>FF,>3C,>00
        BYTE >00,>3C,>3C,>FF,>FF,>3C,>3C,>00
        BYTE >00,>3C,>3C,>FF,>FF,>3C,>3C,>00
        BYTE >00,>3C,>FF,>18,>18,>FF,>3C,>00
        BYTE >00,>FF,>3C,>18,>18,>3C,>FF,>00
        BYTE >C3,>7E,>3C,>18,>18,>3C,>7E,>C3
DMONL   BYTE >18,>3C,>7E,>3C,>18,>18,>18,>3C
        BYTE >18,>3C,>7E,>3C,>18,>18,>18,>3C
        BYTE >18,>3C,>7E,>3C,>18,>18,>18,>3C
        BYTE >18,>3C,>7E,>3C,>18,>18,>18,>3C
        BYTE >00,>18,>3C,>7E,>3C,>18,>A5,>42
        BYTE >00,>18,>3C,>7E,>3C,>18,>A5,>42
        BYTE >00,>18,>3C,>7E,>3C,>18,>A5,>42
        BYTE >00,>18,>3C,>7E,>3C,>18,>A5,>42
DMONM   BYTE >FF,>40,>7C,>20,>3C,>10,>18,>00
        BYTE >00,>FF,>40,>7C,>20,>3C,>10,>18
        BYTE >00,>00,>FF,>40,>7C,>20,>3C,>18
        BYTE >00,>00,>00,>FF,>02,>7C,>3C,>3C
```

```
        BYTE >00,>00,>00,>00,>FF,>7C,>7C,>3C
        BYTE >00,>00,>00,>FF,>08,>7C,>7C,>3C
        BYTE >00,>00,>FF,>04,>7C,>00,>3C,>3C
        BYTE >00,>FF,>02,>7C,>04,>3C,>08,>18
DMONN   BYTE >1C,>3C,>37,>31,>52,>89,>85,>45
        BYTE >1C,>3C,>37,>31,>52,>89,>95,>45
        BYTE >1C,>3C,>37,>31,>31,>52,>4A,>4A
        BYTE >1C,>3C,>37,>31,>31,>52,>4A,>4A
        BYTE >1C,>3C,>36,>52,>32,>51,>51,>92
        BYTE >1C,>3C,>36,>52,>32,>51,>51,>92
        BYTE >38,>78,>66,>62,>91,>92,>92,>61
        BYTE >38,>78,>66,>62,>91,>92,>92,>61
DMONO   BYTE >FF,>18,>FF,>18,>FF,>18,>FF,>18
        BYTE >18,>FF,>18,>FF,>18,>FF,>18,>FF
        BYTE >7E,>18,>7E,>18,>7E,>18,>7E,>18
        BYTE >18,>7E,>18,>7E,>18,>7E,>18,>7E
        BYTE >3C,>18,>3C,>18,>3C,>18,>3C,>18
        BYTE >18,>3C,>18,>3C,>18,>3C,>18,>3C
        BYTE >18,>18,>18,>18,>18,>18,>18,>18
        BYTE >18,>18,>18,>18,>18,>18,>18,>18
DMONP   BYTE >3C,>3C,>3C,>3C,>FF,>FF,>00,>00
        BYTE >3C,>3C,>3C,>3F,>3F,>F0,>F0,>00
        BYTE >3C,>3C,>3F,>3F,>30,>30,>F0,>F0
        BYTE >3C,>3C,>3C,>3F,>3F,>F0,>F0,>00
        BYTE >3C,>3C,>3C,>3C,>FF,>FF,>00,>00
        BYTE >3C,>3C,>3C,>FC,>FC,>0F,>0F,>00
        BYTE >3C,>3C,>FC,>FC,>0C,>0C,>0F,>0F
        BYTE >3C,>3C,>3C,>FC,>FC,>0F,>0F,>00
DMONQ   BYTE >81,>81,>81,>81,>81,>81,>7E,>3C
        BYTE >42,>42,>42,>42,>42,>42,>7E,>3C
        BYTE >24,>24,>24,>24,>24,>24,>18,>18
        BYTE >18,>18,>18,>18,>18,>18,>18,>18
        BYTE >10,>10,>10,>10,>10,>10,>10,>10
        BYTE >18,>18,>18,>18,>18,>18,>18,>18
        BYTE >24,>24,>24,>24,>24,>24,>18,>18
        BYTE >42,>42,>42,>42,>42,>42,>7E,>3C
DMONR   BYTE >18,>BD,>5A,>3C,>3C,>18,>18,>3C
        BYTE >18,>3C,>18,>FF,>3C,>18,>18,>3C
        BYTE >00,>18,>3C,>18,>3C,>FF,>18,>3C
        BYTE >00,>00,>18,>3C,>18,>3C,>7E,>BD
        BYTE >00,>18,>3C,>18,>3C,>FF,>18,>3C
        BYTE >18,>3C,>18,>FF,>3C,>18,>18,>3C
        BYTE >18,>BD,>5A,>3C,>3C,>18,>18,>3C
        BYTE >99,>7E,>3C,>3C,>18,>18,>3C,>00
DMONS   BYTE >00,>3C,>FF,>7E,>00,>00,>03,>03
        BYTE >00,>3C,>FF,>7E,>C0,>C0,>00,>00
        BYTE >00,>3C,>FF,>7E,>00,>C0,>C0,>00
        BYTE >00,>3C,>FF,>7E,>30,>30,>00,>00
        BYTE >00,>3C,>FF,>7E,>00,>00,>30,>30
        BYTE >00,>3C,>FF,>7E,>0C,>0C,>00,>00
        BYTE >00,>3C,>FF,>7E,>00,>00,>0C,>0C
        BYTE >00,>3C,>FF,>7E,>03,>03,>00,>00
DMONT   BYTE >08,>38,>FC,>7C,>7E,>38,>20,>00
        BYTE >20,>38,>7E,>7C,>FC,>38,>08,>00
        BYTE >00,>08,>38,>FC,>7C,>7E,>38,>20
        BYTE >00,>20,>38,>7E,>7C,>FE,>38,>08
        BYTE >00,>04,>1C,>7E,>3E,>3F,>1C,>10
        BYTE >00,>10,>1C,>3F,>3E,>7E,>1C,>04
        BYTE >04,>1C,>7E,>3E,>3F,>1C,>10,>00
        BYTE >10,>1C,>3F,>3E,>7E,>1C,>04,>00
DPC2RT  BYTE >3C,>EA,>FE,>60,>60,>E6,>FE,>3C
        BYTE >3C,>6A,>7E,>F0,>F0,>7E,>7E,>3C
```

```
        BYTE >3C,>EA,>FE,>7C,>7C,>FE,>FE,>3C
DPC2LT  BYTE >3C,>57,>7F,>06,>06,>67,>7F,>3C
        BYTE >3C,>56,>7E,>0F,>0F,>7E,>7E,>3C
        BYTE >3C,>57,>7F,>3E,>3E,>7F,>7F,>3C
DPC2UP  BYTE >00,>66,>E5,>C7,>C5,>FF,>7E,>66
        BYTE >00,>66,>E5,>E7,>FD,>FF,>7E,>81
        BYTE >00,>66,>FD,>FF,>FD,>FF,>7E,>66
DPC2DN  BYTE >66,>7E,>FF,>A3,>E3,>A7,>66,>00
        BYTE >18,>7E,>FF,>BF,>E7,>A7,>66,>00
        BYTE >66,>7E,>FF,>BF,>FF,>BF,>66,>00
DSPLAT  BYTE >89,>40,>24,>08,>C3,>10,>42,>89
        BYTE >7E,>5A,>7E,>7E,>7E,>7E,>7E,>3C
        BYTE >18,>3C,>5A,>7E,>7E,>7E,>7E,>7E
        BYTE >00,>3C,>7E,>5A,>FF,>FF,>FF,>7E
        BYTE >00,>00,>00,>18,>3C,>5A,>FF,>FF
        BYTE >00,>00,>00,>00,>3C,>7E,>DB,>FF
        BYTE >00,>00,>00,>00,>00,>18,>3C,>5A
        BYTE >00,>00,>00,>00,>00,>00,>3C,>7E
        BYTE >00,>00,>00,>00,>00,>00,>00,>3C
        BYTE >89,>40,>24,>08,>C3,>10,>42,>89
        BYTE >00,>00,>00,>00,>00,>00,>00,>00
*****************************************
*                                       *
*               BYTE AREA               *
*                                       *
*****************************************

REGLD   BYTE >00,>A0,00,>0E,01,06,00,>FF          VDP REG VALUES
DCOPYR  BYTE >3C,>42,>99,>A1,>A1,>99,>42,>3C
D1F     BYTE >1F,>1F,>1F,>1F,>1F,>1F,>1F
        BYTE >1F,>1F,>1F,>1F,>1F,>1F
D3F     BYTE >3F,>3F,>3F,>3F,>3F,>3F,>3F,>3F,>3F
D5F     BYTE >5F,>5F
D4F     BYTE >4F
D6F     BYTE >6F
D8F     BYTE >8F
DAF     BYTE >AF
D4F1F   BYTE >4F,>1F
DCLEAR  BYTE >00,>00,>00,>00,>00,>00,>00,>00
DMAZE   BYTE >40,>40,>40,>40,>40,>40,>40,>40
        BYTE >0E,>0E,>0E,>0E,>0E,>0E,>0E,>0E
        BYTE >0A,>0A,>0A,>0A,>0A,>0A,>0A,>0A
        BYTE >5F,>5F,>5F,>5F,>5F,>5F,>5F,>5F
        BYTE >00,>FF,>FF,>FF,>00,>00,>00,>00
        BYTE >00,>00,>00,>00,>00,>00,>FF,>00
        BYTE >FF,>FF,>FF,>FF,>FF,>00,>FF,>00
        BYTE >00,>FF,>00,>FF,>00,>00,>00,>00
        BYTE >00,>1F,>3F,>7F,>40,>40,>40,>40
        BYTE >00,>FC,>FA,>F6,>0E,>0E,>0E,>0E
        BYTE >40,>40,>40,>40,>40,>40,>7F,>00
        BYTE >0E,>0E,>0E,>0E,>0E,>0C,>F8,>00
        BYTE >01,>03,>07,>0F,>0F,>08,>0B,>0A
        BYTE >FE,>FD,>FB,>F7,>EF,>1F,>DF,>5F
        BYTE >0A,>0B,>08,>0F,>00,>00,>00,>00
        BYTE >5E,>DC,>18,>F0,>00,>00,>00,>00
        BYTE >00,>FE,>02,>FA,>0A,>0A,>0A,>0A
        BYTE >00,>7F,>40,>5F,>5F,>5F,>5F,>5F
        BYTE >FA,>FA,>FA,>FA,>FA,>02,>FE,>00
        BYTE >5E,>5D,>5B,>57,>5F,>40,>7F,>00
        BYTE >40,>40,>40,>40,>40,>40,>4F,>4E
        BYTE >0E,>0E,>0E,>0E,>0E,>0E,>CE,>4E
        BYTE >4E,>4E,>4E,>4E,>4E,>4C,>78,>00
```

```
        BYTE >00,>00,>00,>00,>00,>00,>C0,>40
        BYTE >00,>FF,>01,>FF,>00,>00,>00,>00
        BYTE >00,>FF,>80,>FF,>00,>00,>00,>00
        BYTE >5F,>5F,>5F,>5F,>5F,>5F,>5F,>5F
        BYTE >00,>1F,>3F,>7F,>40,>40,>7F,>00
        BYTE >00,>FF,>FF,>FF,>00,>00,>FF,>00
        BYTE >00,>FC,>FA,>F6,>0E,>0C,>F8,>00
        BYTE >4E,>4E,>4E,>4E,>4E,>4E,>4E,>4E
        BYTE >00,>1C,>3A,>76,>4E,>4E,>4E,>4E
DNEWCH  BYTE >4E,>4F,>4F,>4F,>40,>40,>40,>40
DINCL1  BYTE >00,>FF,>FF,>FF,>00,>00,>0F,>0E
        BYTE >00,>FF,>FF,>FF,>00,>00,>C0,>40
        BYTE >40,>C0,>C0,>C0,>00,>00,>00,>00
        BYTE >0C,>0B,>07,>0F,>00,>00,>00,>00
        BYTE >00,>00,>00,>00,>00,>00,>0F,>0E
        BYTE >4E,>CE,>CE,>CE,>0E,>0E,>0E,>0E
DEXPL   BYTE >10,>10,>10,>10,>10,>00,>10,>10
DDOTS   BYTE >00,>00,>00,>00,>00,>00,>00,>00
DARRLT  BYTE >00,>0C,>1C,>3C,>3C,>1C,>0C,>00
DARRRT  BYTE >00,>30,>38,>3C,>3C,>38,>30,>00
DENG11  BYTE >38,>38,>3F,>FF,>FF,>7E,>0C,>04    ENERGIZERS BALL
DENG12  BYTE >38,>38,>3F,>FF,>FF,>7E,>0C,>04    ENERGIZERS BALL
DENG13  BYTE >38,>38,>3F,>FF,>FF,>7E,>0C,>04    ENERGIZERS BALL
DENG14  BYTE >F8,>20,>27,>22,>22,>22,>02,>07    ENERGIZERS BALL
DENG15  BYTE >F8,>20,>27,>22,>22,>22,>02,>07    ENERGIZERS BALL
DENG21  BYTE >38,>28,>2F,>E1,>81,>72,>0C,>04    ENERGIZERS CIRCLE
DENG22  BYTE >38,>28,>2F,>E1,>81,>72,>0C,>04    ENERGIZERS CIRCLE
DENG23  BYTE >38,>28,>2F,>E1,>81,>72,>0C,>04    ENERGIZERS CIRCLE
DENG31  BYTE >38,>28,>2F,>E3,>E3,>7E,>0C,>04    ENERGIZERS BALL W/ WHOLE
DENG32  BYTE >38,>28,>2F,>E3,>E3,>7E,>0C,>04    ENERGIZERS BALL W/ WHOLE
DENG33  BYTE >38,>28,>2F,>E3,>E3,>7E,>0C,>04    ENERGIZERS BALL W/ WHOLE
DBORDA  BYTE >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF
DBORDB  BYTE >FF,>7F,>3F,>1F,>0F,>07,>03,>01
DBORDC  BYTE >FF,>FE,>FC,>F8,>F0,>E0,>C0,>80
DBORDD  BYTE >01,>03,>07,>0F,>1F,>3F,>7F,>FF
DBORDE  BYTE >80,>C0,>E0,>F0,>F8,>FC,>FE,>FF
DNUMOA  BYTE >00,>31,>4A,>4A,>4A,>4A,>4A,>31
DNUMOB  BYTE >0E,>8E,>4E,>4E,>4E,>4E,>4E,>8E
DNUMO1  BYTE >40,>44,>4C,>44,>44,>44,>44,>4E
DNUMO2  BYTE >40,>46,>49,>41,>41,>46,>48,>4F
DNUMO4  BYTE >40,>4A,>4A,>4A,>4F,>42,>42,>42
DNUMO8  BYTE >40,>46,>49,>49,>46,>49,>49,>46
CLRDAT  BYTE >4F,>3F,>EF,>7F,>AF,>3F,>DF,>FF
CHADAT  BYTE >8F,>4F,>CF,>4F,>6F,>4F,>EF,>FF
DCHAIN  DATA >0000,>BB44,>44BB,>0000,>FCFE,>FEFF
        DATA >FFFF,>FFFF,>0000,>1B24,>241B
        DATA >2424,>0000,>BB44,>44B8,>2424,>0000,>BB44,>44BB
        DATA >2424,>0000,>0000,>003C,>0000
        DATA >2418,>1B24,>241B,>0000,>2418,>A464,>6498,>0000
        DATA >2418,>BB44,>44BB,>0000,>2418
        DATA >2424,>2418,>2424,>2418,>2724,>241B,>2424,>2418
        DATA >A464,>6498,>2424,>2418,>BB44
        DATA >44BB,>2424
DLOGO   DATA >ABB3,>BBBF,>FEFC,>0000,>BBBB,>BBBF,>FFFB,>BB9B
        DATA >3F7F,>BB93
        DATA >ABBB,>BBBB,>BFFF,>FBBB,>BBBB,>BBC7,>FFFB,>BB9B
        DATA >ABB3,>BBBF,>FFC3,>BFBF,>BFBF
        DATA >C3FF,>FBBB,>BB83,>BBBB,>BEFC,>BBBF,>FFEF,>D7BB
        DATA >BB83,>BB00
ENGOO   *DATA @DENG11,@DENG15,@DENG14,@DENG13,@DENG12
        *DATA @DENG31,@DENG15,@DENG14,@DENG33,@DENG32
        *DATA @DENG21,@DENG15,@DENG14,@DENG23,@DENG22
```

```
MESS1   BYTE >00
        TEXT 'VERY'
        BYTE >01
        BYTE >00
        TEXT 'GOOD'
        BYTE >01
        BYTE >00
        TEXT 'MOVE'
        BYTE >01
        BYTE >00
        TEXT 'YOUR'
        BYTE >01
        BYTE >00
        TEXT 'GOOD'
        BYTE >01
        BYTE >00
        TEXT 'PLAY'
        BYTE >01
        BYTE >00
        TEXT 'LINK'
        BYTE >01
        BYTE >00
        TEXT 'GOOD'
        BYTE >01
        BYTE >00
        TEXT 'NICE'
        BYTE >01
        BYTE >00
        TEXT 'NICE'
        BYTE >01
        BYTE >00
        TEXT 'LINK'
        BYTE >01
        BYTE >00
        TEXT ' GO '
        BYTE >01
MESS2   BYTE >00
        TEXT 'GOOD'
        BYTE >01
        BYTE >00
        TEXT 'PLAY'
        BYTE >01
        BYTE >00
        TEXT 'FAST'
        BYTE >01
        BYTE >00
        TEXT 'TURN'
        BYTE >01
        BYTE >00
        TEXT 'WORK'
        BYTE >01
        BYTE >00
        TEXT ' ON '
        BYTE >01
        BYTE >00,>CF
        TEXT 'IT'
        BYTE >CF,>01
        BYTE >00
        TEXT 'CCCC'
        BYTE >01
        BYTE >00
        TEXT 'SHOW'
```

```
          BYTE >01
          BYTE >00
          TEXT 'GAME'
          BYTE >01
          BYTE >00
          TEXT 'AWAY'
          BYTE >01
          BYTE >00,>CF
          TEXT 'CC'
          BYTE >CF,>01
***************************************
*        SOUND        LISTS           *
***************************************
SODEAT    BYTE 7,>9F,>BF,>DF,>F2,>CC,>01,>E7,1
          BYTE 2,>CC,>03,1
          BYTE 2,>CC,>05,1
          BYTE 1,>FF,0
SNDENG    BYTE 6,>9F,>DF,>FF,>A0,>08,>B2,1
          BYTE 1,>B4,1
          BYTE 1,>B6,1
          BYTE 1,>BF,0
SNDREW    BYTE 6,>9F,>B4,>DF,>FF,>A0,>04,10
          BYTE 1,>B6,8
          BYTE 1,>B8,7
          BYTE 1,>BA,6
          BYTE 1,>BB,5
          BYTE 1,>BD,4
          BYTE 1,>BE,2
          BYTE 1,>BF,0
SNDWI     BYTE 7,>9F,>BF,>DF,>FF,>80,>05,>99,10
          BYTE 2,>80,>06,10
          BYTE 2,>80,>05,6
          BYTE 2,>80,>06,6
          BYTE 2,>80,>05,5
          BYTE 2,>80,>06,5
          BYTE 2,>80,>05,4
          BYTE 2,>80,>06,4
          BYTE 2,>80,>05,3
          BYTE 2,>80,>06,3
          BYTE 2,>80,>05,2
          BYTE 2,>80,>06,2
          BYTE 2,>80,>05,1
          BYTE 2,>80,>06,1
          BYTE 1,>9F,0
SNEND     BYTE 4,>9F,>BF,>DF,>FF,1
          BYTE 6,>81,>2C,>96,>A0,>2C,>B6,2
          BYTE 4,>81,>2E,>A0,>2E,2
          BYTE 4,>81,>30,>A0,>30,2
          BYTE 4,>81,>32,>A0,>32,3
          BYTE 4,>81,>34,>A0,>34,3
          BYTE 4,>81,>35,>A0,>35,3
          BYTE 4,>81,>36,>A0,>36,4
          BYTE 4,>81,>37,>A0,>37,4
          BYTE 4,>81,>38,>A0,>38,5
          BYTE 4,>81,>39,>A0,>39,5
          BYTE 4,>81,>3A,>A0,>3A,6
          BYTE 4,>81,>3B,>A0,>3B,7
          BYTE 4,>81,>3C,>A0,>3C,8
          BYTE 4,>81,>3D,>A0,>3D,9
          BYTE 4,>81,>3E,>A0,>3E,10
          BYTE 4,>81,>3F,>A0,>3F,20
          BYTE 7,>9F,>BF,>DF,>CC,>01,>F2,>E3,5
```

```
        BYTE  1,>F3,4
        BYTE  1,>F4,4
        BYTE  1,>F5,3
        BYTE  1,>F6,3
        BYTE  1,>F7,2
        BYTE  1,>F8,2
        BYTE  1,>F9,1
        BYTE  1,>FA,1
SNDOF   BYTE  4,>9F,>BF,>DF,>FF,0
SNDMU   BYTE  6,>9F,>BF,>FF,>C2,>1A,>DA,1
        BYTE  2,>C2,>15,1
        BYTE  2,>C2,>10,1
        BYTE  1,>DF,0
SODCHM  BYTE  5,>9F,>BF,>DF,>FF,>E3,1
CHLOO   BYTE  9,>8B,>01,>A4,>02,>C5,>01,>9C,>BE,>DE,15
        BYTE  5,>A7,>04,>9E,>BC,>DE,18
        BYTE  5,>CA,>02,>9E,>BE,>DC,15
        BYTE  5,>85,>03,>9C,>BE,>DE,18
        BYTE  5,>A4,>02,>9E,>BE,>DE,15
        BYTE  5,>C5,>01,>9E,>BE,>DC,18
        BYTE  0,>35,>07
SNDCH   BYTE  3,>9E,>BE,>DC,30
        BYTE  4,>9F,>BF,>DF,>FF,0
SNDTI   BYTE  7,>9F,>BF,>DF,>FA,>C0,>01,>E7,1
        BYTE  2,>C0,>03,1
        BYTE  2,>C0,>06,1
        BYTE  2,>C0,>10,1
        BYTE  2,>C0,>15,1
        BYTE  2,>C0,>20,1
        BYTE  2,>C0,>25,1
        BYTE  2,>C0,>30,1
        BYTE  1,>FF,0
SNDMO   BYTE  3,>9E,>BE,>DF,1
SNDMV   BYTE  6,>87,>03,>A8,>03,>C0,>05,1
        BYTE  2,>C0,>05,10
        BYTE  4,>88,>03,>A7,>03,1
        BYTE  4,>89,>03,>A6,>03,1
        BYTE  4,>8A,>03,>A5,>03,1
        BYTE  4,>8B,>03,>A4,>03,1
        BYTE  4,>8C,>03,>A3,>03,1
        BYTE  4,>8D,>03,>A2,>03,1
        BYTE  0,>35,>70
TITLES  DATA  >B0B0,>B0B5,>2020,>2020,>2020,>2020,>2020,>2020
        DATA  >2020,>2020,>2020,>2020,>2020,>2020,>B4B0,>B0B0
        DATA  >B0B0,>B520,>20A8,>A920,>2020,>20B3,>B320,>2020
        DATA  >2020,>2020,>20A8,>A920,>2020,>20B3,>B3B4,>B0B0
        DATA  >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
        DATA  >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
        DATA  >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
        DATA  >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
        DATA  >B0B0,>2070,>2020,>2070,>2060,>2020,>6020,>8020
        DATA  >2080,>2020,>A0A0,>A020,>E820,>20E8,>2020,>B0B0
        DATA  >B0B0,>2070,>7020,>7070,>2060,>2020,>6020,>8080
        DATA  >2080,>20A0,>2020,>2020,>E820,>20E8,>2020,>B0B0
        DATA  >B0B0,>2070,>2070,>2070,>2060,>2020,>6020,>8020
        DATA  >8080,>20A0,>2020,>2020,>E8E8,>E8E8,>2020,>B0B0
        DATA  >B0B0,>2070,>2020,>2070,>2020,>6060,>2020,>8020
        DATA  >2080,>2020,>A0A0,>A020,>E820,>20E8,>2020,>B0B0
        DATA  >B0B0,>2070,>2020,>2070,>2020,>2020,>2020,>2020
        DATA  >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
        DATA  >B0B0,>2070,>2020,>2070,>2020,>2020,>2070,>2020
        DATA  >2070,>2020,>F0F0,>2020,>F820,>20F8,>2020,>B0B0
```

off

- out for snd

title screen sound

```
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2070,>7020
            DATA >7070,>20F0,>2020,>F020,>F8F8,>20F8,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2070,>2070
            DATA >2070,>20F0,>F0F0,>F020,>F820,>F8F8,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2070,>2020
            DATA >2070,>20F0,>2020,>F020,>F820,>20F8,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2070,>2020
            DATA >2070,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2070,>2020
            DATA >2070,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>2020,>2050,>5245,>5353,>B041,>4E59,>B04B
            DATA >4559,>B054,>4FB0,>4245,>4749,>4E20,>2020,>B0B0
            DATA >B0B0,>2020,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>2020,>B0B0
            DATA >B0B0,>B720,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>20B6,>B0B0
            DATA >B0B0,>B0B7,>2020,>2020,>2020,>2020,>2020,>2020
            DATA >2020,>2020,>2020,>2020,>2020,>2020,>B6B0,>B0B0
            DATA >B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0
            DATA >B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0,>B0B0
            DATA >B0B0,>B0B0,>4031,>3938,>32B0,>B054,>4558,>4153
            DATA >B049,>4E53,>5452,>554D,>454E,>5453,>B0B0,>B0B0
SCREEN 0    DATA >200C,>0606,>0606,>0606,>0606,>0606,>0606,>0606
            DATA >0606,>0606,>0606,>0606,>0606,>0606,>0606,>0D20
1           DATA >2002,>D5D3,>D3D3,>D3D3,>D7D3,>D3D3,>D3D3,>D3D3
            DATA >D3D3,>D3D3,>D3D3,>D3D7,>D3D3,>D3D3,>D3D6,>0320
2           DATA >2002,>6C1B,>1C23,>0409,>DC08,>0422,>1C1C,>1C1C
            DATA >1C1C,>1C1C,>2304,>09DC,>0804,>221C,>1D6C,>0320
3           DATA >2002,>D9D3,>D600,>2001,>DC00,>2001,>D5D3,>D3D3
            DATA >D3D3,>D3D6,>0020,>01DC,>0020,>01D5,>D3DA,>1A20
4           DATA >200E,>0710,>DC00,>2001,>DC00,>2001,>DC08,>0404
            DATA >0404,>09DC,>0020,>01DC,>0020,>01DC,>1107,>0F20
5           DATA >0606,>0612,>DC0A,>050B,>DC0A,>050B,>DC0A,>0505
            DATA >0505,>0BDC,>0A05,>0BDC,>0A05,>0BDC,>1306,>0606
6           DATA >D3D3,>D3D3,>DFD3,>D3D3,>DFD3,>D3D3,>DFD3,>D3D3
            DATA >D3D3,>D3DF,>D3D3,>D3DF,>D3D3,>D3DF,>D3D3,>D3D3
7           DATA >0707,>0710,>DC08,>0409,>DC08,>0409,>DC08,>0404
            DATA >0404,>09DC,>0804,>09DC,>0804,>09DC,>1107,>0707
8           DATA >2020,>2002,>DC00,>2001,>DC00,>B1B1,>DC14,>0505
            DATA >0505,>15DC,>B2B2,>01DC,>0020,>01DC,>0320,>2020
9           DATA >2020,>2002,>DC00,>2001,>DC0A,>050B,>DC16,>2020
            DATA >2020,>16DC,>0A05,>0BDC,>0020,>01DC,>0320,>2020
10          DATA >2020,>2002,>DC00,>2001,>DDD3,>D3D3,>DEB2,>B3B3
            DATA >B3B3,>B1DD,>D3D3,>D3DE,>0020,>01DC,>0320,>2020
11          DATA >2020,>2002,>DC00,>2001,>DC08,>0409,>DC1F,>2020
            DATA >2020,>1FDC,>0804,>09DC,>0020,>01DC,>0320,>2020
12          DATA >2020,>2002,>DC00,>2001,>DC00,>B1B1,>DC21,>0404
            DATA >0404,>27DC,>B2B2,>01DC,>0020,>01DC,>0320,>2020
13          DATA >200C,>0612,>DC0A,>050B,>DC0A,>050B,>DC0A,>0505
            DATA >0505,>0BDC,>0A05,>0BDC,>0A05,>0BDC,>1306,>0D20
14          DATA >2002,>D5D3,>DBD3,>D3D3,>DBD7,>D3D3,>DBD3,>D7D3
            DATA >D3D7,>D3DB,>D3D3,>D7DB,>D3D3,>D3DB,>D3D6,>0320
15          DATA >2002,>6C1B,>1C1C,>1C1C,>1DDC,>0804,>0409,>DC08
            DATA >09DC,>0804,>0409,>DC1B,>1C1C,>1C1C,>1D6C,>0320
16          DATA >2002,>D9D3,>D7D3,>D3D3,>D3DE,>0026,>050B,>DC00
            DATA >01DC,>0A05,>1701,>DDD3,>D3D3,>D3D7,>D3DA,>1A20
```

```
17 DATA >200E,>0710,>DC1B,>1C1C,>1DDC,>0001,>D5D3,>DA00
   DATA >01D9,>D3D6,>0001,>DC1B,>1C1C,>1DDC,>1107,>0F20
18 DATA >2020,>2002,>DDD3,>D3D3,>D3DE,>0001,>DC08,>0424
   DATA >2504,>09DC,>0001,>DDD3,>D3D3,>D3DE,>0320,>2020
19 DATA >0606,>0612,>DC11,>0707,>10DC,>0A0B,>DC0A,>0505
   DATA >0505,>0BDC,>0A0B,>DC11,>0707,>10DC,>1306,>0606
20 DATA >D3D3,>D3D3,>DA1A,>2020,>02D9,>D3D3,>DBD3,>D3D3
   DATA >D3D3,>D3DB,>D3D3,>DA1A,>2020,>02D9,>D3D3,>D3D3
21 DATA >0707,>0707,>070F,>2020,>0E41,>5245,>2059,>4F55
   DATA >2052,>4541,>4459,>3F0F,>2020,>0E07,>0707,>0707

*****************************************************************
*                        START OF MAINLINE                      *
*****************************************************************
*****************************************************************
*   THIS PROGRAM IS TO BE USED ON DISK TO SET-UP A GPL "TYPE"   *
*   ENVIRONMENT FOR USE WITH ARCADE GAMES TO BE RUN WITH        *
*   X-BASIC AND/OR EDITOR/ASSM. LOADERS.                        *
*****************************************************************
***     REF  GAME                          THE 9900 CODE PROGRAM MUST HAVE
***                                        A REF OF 'GAME' FOR THIS PROGRAM
***                                        TO LINK TO
START   LI   R1,REGLD                      SET BYTE ADDRESS
        LI   R2,>7F00                      REG WRITE CONSTANT -1
LOOP    SWPB R2                            MOVE HIGH TO LOW/LOW TO HIGH
        INC  R2                            SET REGISTER NUMBER
        MOVB *R1+,R2                       SET VALUE FOR OUTPUT
        MOVB R2,@>8C02                     WRITE VALUE
        SWPB R2                            MOVE HIGH TO LOW/LOW TO HIGH
        MOVB R2,@>8C02                     WRITE REGISTER NUMBER
        CI   R2,>8700                      CHECK FOR LAST REGISTER
        JL   LOOP                          GO BACK FOR NEXT REGISTER & VALUE
***     B    @GAME                         GOTO "9900 CODE" PROGRAM
***  MOVE UP CHARACTERS(>20 - >5F) A FULL >200 BYTES WORTH
***
GAME    LWPI MYWS
        CLR  R10
        CLR  R4
        CLR  R14
        LI   R4,>0901
        LI   R14,>0900
        LI   RLOC,REGALB
        LI   WLOC,REGALB
        LI   RCOUNT,1
MOREBY  MOV  R4,VDPADD
        BL   @READ
        MOV  R14,VDPADD
        BL   @WRITE
        INC  R4
        INC  R14
        CI   R14,>0B01
        JNE  MOREBY
***
***  GET COPYRIGHT CHARACTER
***
        LI   VDPADD,>0850
        LI   RCOUNT,8
        LI   RLOC,YXLOCP
        BL   @READ
        LI   VDPADD,>0A00
        LI   WLOC,YXLOCP
        BL   @WRITE
```

```
***
***   CHAR >28,>29,>2D    ------>     >5D,>5E,>5F ...  CHARS "(,),-"
***
        LI    VDPADD,>0940
        LI    RCOUNT,8
        LI    RLOC,BUFF1
        LI    WLOC,BUFF1
        BL    @READ
        LI    VDPADD,>0AE8
        BL    @WRITE
        LI    VDPADD,>0948
        BL    @READ
        LI    VDPADD,>0AF0
        BL    @WRITE
        LI    VDPADD,>0968
        BL    @READ
        LI    VDPADD,>0AF8
        BL    @WRITE
***
***   INIT MAZE CHARACTERS
***
        LI    VDPADD,>0800
        LI    WCOUNT,256
        LI    WLOC,DMAZE
        BL    @WRITE
***
***   INIT HOR,VERT,CORNER LINES
***
        LI    VDPADD,>0908
        LI    WCOUNT,56
        LI    WLOC,DNEWCH
        BL    @WRITE
***
***   INIT "!"([) AND       ***    DOOR CHARACTERS
***
        LI    VDPADD,>0AD8
        LI    WCOUNT,8
        LI    WLOC,DEXPL
        BL    @WRITE
***
***   INIT MMAN REMAINING
***
        LI    VDPADD,>0BA0
        LI    WCOUNT,8
        LI    WLOC,DPC2RT
        BL    @WRITE
***
***   INIT SPLAT AND EATEN MMEN
***
        LI    VDPADD,>0600
        LI    WCOUNT,88
        LI    WLOC,DSPLAT
        BL    @WRITE
***
***     INIT MUNCH-MAN LOGO
***
        LI    VDPADD,>0940
        LI    WCOUNT,64
        LI    WLOC,DLOGO
        BL    @WRITE
***
***   INIT NUMBERS 0,0,1,2,4,8
```

```
***
        LI    VDPADD,>0F00
        LI    WCOUNT,48
        LI    WLOC,DNUMOA
        BL    @WRITE
***
***     INIT 13 BLANK CHARACTERS
***
        LI    VDPADD,>0E18
        LI    WCOUNT,204
        LI    WLOC,DCHAIN
        BL    @WRITE
***
***   INIT DOT CHARACTERS
***
        LI    VDPADD,>0E98
        LI    WCOUNT,8
        LI    WLOC,DDOTS
MOREDT  BL    @WRITE
        AI    VDPADD,8
        CI    VDPADD,>0F00
        JNE   MOREDT
***
***   INIT RT,LT GRN AND BLU ARROWS FOR SCORE BORDERS
***
        LI    VDPADD,>0BA8
        LI    WCOUNT,16
        LI    WLOC,DARRLT
        BL    @WRITE
        LI    VDPADD,>0CA8
        BL    @WRITE
***
***   INIT MONSTERS(HOONOS) A - F
***
        LI    VDPADD,>0B00
        LI    WCOUNT,64
        LI    WLOC,DMONA
        BL    @WRITE
        LI    VDPADD,>0C00
        LI    WLOC,DMONB
        BL    @WRITE
        LI    VDPADD,>0D00
        LI    WLOC,DMONC
        BL    @WRITE
        LI    VDPADD,>0F40
        LI    WLOC,DMOND
        BL    @WRITE
        LI    VDPADD,>0F80
        LI    WLOC,DMONE
        BL    @WRITE
        LI    VDPADD,>0FC0
        LI    WLOC,DMONF
        BL    @WRITE
***
***   INIT MUNCH-MAN "P"
***
        LI    VDPADD,>0B80
        LI    WCOUNT,24
        LI    WLOC,DPC2RT
        BL    @WRITE
***
***   INIT DRED BORDERS(RT,LT)
```

```
***
        LI    VDPADD,>0D80
        LI    WCOUNT,8
        LI    WLOC,DBORDA
        BL    @WRITE
        LI    VDPADD,>0DA0
        LI    WCOUNT,32
        LI    WLOC,DBORDB
        BL    @WRITE
***
***   LOAD IN SOUND AT HIGH VDP
***
        LI    VDPADD,>3000
        LI    WCOUNT,300
        LI    WLOC,SODEAT
        BL    @WRITE
        LI    VDPADD,>3500
        LI    WCOUNT,354
        LI    WLOC,SODCHM
        BL    @WRITE
***
***   INIT COLORS IN PCT
***
        LI    VDPADD,>0385
        LI    WCOUNT,1
        LI    WLOC,D6F
        BL    @WRITE
        LI    VDPADD,>0380
        LI    WCOUNT,13
        LI    WLOC,D1F
        BL    @WRITE
        LI    VDPADD,>038D
        LI    WCOUNT,9
        LI    WLOC,D3F
        BL    @WRITE
        LI    VDPADD,>0396
        LI    WCOUNT,4
        LI    WLOC,D1F
        BL    @WRITE
        LI    VDPADD,>039A
        LI    WCOUNT,3
        LI    WLOC,D5F
        BL    @WRITE
        LI    VDPADD,>0390
        LI    WCOUNT,1
        LI    WLOC,D1F
        BL    @WRITE
        LI    VDPADD,>0392
        LI    WCOUNT,D4F
        BL    @WRITE
        LI    VDPADD,>0394
        LI    WLOC,D8F
        BL    @WRITE
        LI    VDPADD,>038C
        LI    WLOC,DAF
        BL    @WRITE
        LI    VDPADD,>039E
        LI    WCOUNT,2
        LI    WLOC,D4F1F
        BL    @WRITE
***
***   ZERO OUT HIGH SCORE
```

```
***
        LI    VDPADD,>2000
        LI    WCOUNT,6
        LI    WLOC,MSGZER
        BL    @WRITE
***
***     TURN ON THE SCREEN
***
        MOVB  HE0,@>8C02          DATA FOR VDP REG#1
        MOVB  H81,@>8C02          WRITE BYTE >E0 TO VDP REG#1
BACK    LI    R10,SNDOFF          TURN OFF ALL
        BL    @SOUND              SOUNDS
        MOVB  H00,DEMOFG          DEFAULT = NO DEMO
***
***     GET TITLE SCREEN(FIRST DISPLAY OF THE GAME)
***
        BL    @TILSCN             PUT OUT TITLE SCREEN
***
***  DELAY AND SCAN W/ TIME-OUT
***
LOOPBK  CLR   R4
        MOVB  H00,KEYBRD          KEYBOARD TYPE = ZERO
        MOV   H0002,R4            SET TO 2 SECONDS
MORETM  MOVB  H00,TIMER           SET DELAY LIMIT(1/60TH SEC) = 60
DELSSS  BL    @SCANKY             SCAN KEYBOARD
        CB    KEY,HFF             ANY KEY PRESSED?
        JNE   CHKHOF              YES, SO START THE GAME
        MOVB  H00,CLRSCN          CLEAR SCREEN TIME-OUT COUNTER
        CB    TIMER,H3C
        JNE   DELSSS
        DEC   R4
        JNE   MORETM
TITLGO  LI    R10,SNDOFF          TURN OFF
        BL    @SOUND              ALL SOUNDS
        LI    R10,SNDMOV          TURN ON
        BL    @SOUND              TITLE SCREEN MOVE SOUND
        B     @XML01              DON'T BLANK OUT SCREEN
CHKHOF  CB    KEY,H0F             STILL "BACK" KEY PRESSED?
        JEQ   LOOPBK              YES, SO LOOP BACK AGAIN
        CB    KEY,H2A             "*"
        JEQ   CHK23
        B     @GOOGAM             START THE GAME
CHK23   BL    @PAUSOA
        BL    @SCANKY
        CB    KEY,HFF
        JEQ   CHK23
        CB    KEY,H2A
        JEQ   CHK23
        CB    KEY,H23             "#"
        JNE   LOOPBK              START TIME DELAY AGAIN
CHK2A   BL    @PAUSOA
        BL    @SCANKY
        CB    KEY,HFF
        JEQ   CHK2A
        CB    KEY,H23
        JEQ   CHK2A
        CB    KEY,H2A             "*"
        JNE   LOOPBK              START TIME DELAY AGAIN
***
***  "***" WAS ENTERED
***
        LI    VDPADD,0
```

```
        LI    WCOUNT,1
SOME20  LI    WLOC,WINDT2       GET BLANKS(>20)
        BL    @WRITE
        INC   VDPADD
        CI    VDPADD,769
        JNE   SOME20
***
***  DELETE ALL SPRITES
***
        LI    VDPADD,>0300
        LI    WCOUNT,1
        LI    WLOC,HD0
        BL    @WRITE
***
***  PUT OUT "CHEAT" MESSAGES
***
        LI    VDPADD,165
        LI    WCOUNT,8
        LI    WLOC,MSGRND       "RND(0-2)" MESSAGE
        BL    @WRITE
CHKRND  BL    @SCANKY
        CB    KEY,H2A
        JEQ   CHKRND
        SB    H30,KEY
        CB    KEY,H02           0 - 2 ??
        JH    CHKRND
        MOVB  KEY,SCRNUB        GOT ROUND NUMBER
        LI    VDPADD,229
        LI    WCOUNT,10
        LI    WLOC,MSGSCN       "SCN(00-19)" MESSAGE
        BL    @WRITE
CHKSCN  BL    @SCANKY
        MOVB  STATUS,STATUS
        JEQ   CHKSCN
        SB    H30,KEY
        CB    KEY,H01           0 - 1 ??
        JH    CHKSCN
        MOVB  KEY,SCRNUM        GOT SCREEN LEVEL
CHKNXT  BL    @SCANKY
        MOVB  STATUS,STATUS
        JEQ   CHKNXT
        SB    H30,KEY
        CB    KEY,H09           0 - 9 ??
        JH    CHKNXT
        MOVB  SCRNUM,SCRNUM
        JEQ   DIGONE
        MOVB  H0A,SCRNUM
DIGONE  AB    KEY,SCRNUM
        LI    VDPADD,293
        LI    WCOUNT,7
        LI    WLOC,MSGMM        "MM(1-9)" MESSAGE
        BL    @WRITE
CHKMM   BL    @SCANKY
        MOVB  STATUS,STATUS
        JEQ   CHKMM
        SB    H30,KEY
        MOVB  KEY,KEY           KEY = ZERO(WAS >30)?
        JEQ   CHKMM             MASK OUT KEY = 00
        CB    KEY,H09           1 - 9 ??
        JH    CHKMM
        MOVB  KEY,COUNT1        GOT MUNCH-MAN COUNT
***
```

```
***  BLANK OUT THE SCREEN
***
        LI    VDPADD,0
        LI    WCOUNT,1
GET020  LI    WLOC,WINDT2           GET THE BLANKS(>20)
        BL    @WRITE
        INC   VDPADD
        CI    VDPADD,769
        JNE   GET020
        LI    VDPADD,37             WRITE "TEST SCORE" MESSAGE
        LI    WCOUNT,10
        LI    WLOC,MSGTST
        BL    @WRITE
        JMP   NOYOUR                SKIP AROUND "YOUR SCORE" MESSAGE
***
***  BLANK OUT THE SCREEN
***
GOOGAM  LI    VDPADD,0
        LI    WCOUNT,1
MORE20  LI    WLOC,WINDT2           GET THE BLANKS(>20)
        BL    @WRITE
        INC   VDPADD
        CI    VDPADD,769
        JNE   MORE20
        LI    VDPADD,37             WRITE "YOUR SCORE" MESSAGE
        LI    WCOUNT,10
        LI    WLOC,MSGYOU
        BL    @WRITE
        MOVB  H00,SCRNUM
        MOVB  H00,SCRNUB
        MOVB  H03,COUNT1
        CB    DEMOFG,H00            IN DEMO MODE?
        JEQ   NOYOUR                NO, SO NO DEMO SCORE MSG
        LI    WLOC,MSGDEM
        BL    @WRITE                WRITE "DEMO SCORE" MESSAGE
***
***     WHITE BACKGROUND COLOR
***
NOYOUR  MOVB  HFF,@>8C02            SEND DATA TO VDP WRITE WINDOW
        MOVB  H87,@>8C02            WRITE >FF TO VDP REG 7(WHITE BACKDROP)
        LI    VDPADD,>0386          RESTORE PCT TO ORIGINAL COLOR
        LI    WCOUNT,6                        "
        LI    WLOC,COLBLA                     "
        BL    @WRITE                          "
        LI    WCOUNT,3
        LI    VDPADD,>0396                    "
        BL    @WRITE                          "
        LI    WCOUNT,1                        "
        LI    WLOC,COLGRN
        LI    VDPADD,>0395                    "
        BL    @WRITE                          "
***
***  INIT RT,LT ARROWS, GET MMAN REMAINING, RESTORE BLANK CHAR
***
        LI    VDPADD,>0BA8
        LI    WCOUNT,16
        LI    WLOC,DARRLT
        BL    @WRITE
        LI    VDPADD,>0BA0
        LI    WCOUNT,8
        LI    WLOC,DPC2RT
        BL    @WRITE
```

```
        LI    VDPADD,>0E18
        LI    WLOC,DCHAIN
        BL    @WRITE
        MOVB  H00,TITLFG
        MOVB  H00,KEYBRD
        CLR   SCRSAV
        CLR   WLOC
        BL    @SHOMAN              SHOW NUMBER OF MUNCH-MAN REMAINING
***
***   DISPLAY "HIGH SCORE" MSG AMD ARROWS
***
        LI    VDPADD,5
        LI    WCOUNT,10
        LI    WLOC,MSGHI
        BL    @WRITE
        LI    VDPADD,16
        LI    WCOUNT,1
        LI    R10,>0075
        LI    WLOC,REGALB
        BL    @WRITE
        LI    VDPADD,23
        LI    R10,>0076
        BL    @WRITE
        LI    VDPADD,48
        LI    R10,>0095
        BL    @WRITE
        LI    VDPADD,55
        LI    R10,>0096
        BL    @WRITE
***
***   ZERO OUT YOUR SCORE NOW
***
        LI    VDPADD,49
        LI    WCOUNT,6
        LI    WLOC,MSGZER
        BL    @WRITE
***
***   GET BEST SCORE FROM HIGH VDP AND DISPLAY
***
        LI    VDPADD,>2000         GET HIGH SCORE SAVED
        LI    RLOC,YXLOCP
        LI    RCOUNT,6
        BL    @READ
        LI    VDPADD,17            MOVE IT TO SCREEN POSTION
        LI    WLOC,YXLOCP
        BL    @WRITE
NEWGAM  BL    @INIT01              " R E D O "
NXTGAM  LI    R10,SNDOFF           TURN OFF
        BL    @SOUND               ALL SOUNDS
***
***   PUT OUT MAZE AND DOTS
***
        LI    VDPADD,64            START AT TOP OF MAZE
        LI    WCOUNT,704           WRITE ONLY MAZE OUT
        LI    WLOC,SCREEN          GET SCREEN MAZE, DOTS DATA
        BL    @WRITE               DISPLAY IT NOW
        MOVB  H02,DOTCNT+1         INIT DOT COUNT
        CB    DEMOFG,H00           GAME DEMOING?
        JEQ   CONT01               NO,  SO CONTINUE
        LI    VDPADD,739           WRITE PRESS ANY . . . MSG AT BOTTOM
        LI    WCOUNT,26
        LI    WLOC,MSGBEG          "PRESS ANY KEY TO BEGIN MSG"
```

```
        BL    @WRITE
***
***     INIT SAB'S AND GET DATA FOR MUNCH-MAN
***
CONTO1  MOVB  H00,KEYBRD
        LI    VDPADD,>0300
        LI    WCOUNT,21
        LI    WLOC,SABMAN
        BL    @WRITE
        LI    VDPADD,>0400
        LI    WCOUNT,24
        LI    WLOC,DPC2RT
        BL    @WRITE
        LI    VDPADD,>0480
        LI    WLOC,DPC2LT
        BL    @WRITE
        LI    VDPADD,>0500
        LI    WLOC,DPC2UP
        BL    @WRITE
        LI    VDPADD,>0580
        LI    WLOC,DPC2DN
        BL    @WRITE
        CB    SCRNUM,H14          PAST 20TH SCREEN?
        JNE   KEEPNM              NOP, KEEP GOING
        MOVB  H00,SCRNUM          RE-INIT OT LEVEL #0 - 19
        AB    H01,SCRNUB          NEXT ROUND
        BL    @INITO1             RE-INIT GAME PARAMETERS
KEEPNM  BL    @INITO3             INIT ALL SCREEN LEVEL INFO
        MOVB  H00,WINFLG          INIT WIN FLAG TO NO WIN YET
***
*** ON 20TH SCREEN(SCRNUM=19) PUT OUT TEXAS MAPS
*** AND LAY DOWN BLANKS INSTEAD OFF CHAINS
***
        LI    WLOC,DDOTS          MAZE INITIALLY BLANKS(DDOTS) DATA
        CB    SCRNUM,H13
        JNE   NOTEX
        LI    WLOC,DENG21         TEXAS MAPS DATA
NOTEX   LI    VDPADD,>0E98
        LI    WCOUNT,8
MORDT1  BL    @WRITE
        AI    VDPADD,8
        CI    VDPADD,>0F00
        JNE   MORDT1
***
***  USE 2 BYTE SAVR11 TO PUT OUT SCREEN LEVEL(#1-20) AT UPPER CORNER
***
        CLR   SAVR11              PUT OUT SCREEN LEVEL(#1-20)
        MOVB  SCRNUM,SAVR11+1                 "
        AB    H01,SAVR11+1                    "
        CB    SAVR11+1,H09                    "
        JH    SECDIG                          "
        AB    H30,SAVR11+1                    "
        MOVB  H30,SAVR11                      "
        JMP   SHOWLV                          "
SECDIG  CB    SAVR11+1,H14                    "
        JNE   NOTTWY                          "
        LI    VDPADD,>0022                    "
        LI    WCOUNT,2                        "
        LI    WLOC,H3230                      "
        BL    @WRITE                          "
        JMP   LEVTWY                          "
NOTTWY  SB    H0A,SAVR11+1                    "
```

```
        AB    H30,SAVR11+1                    "
        MOVB  H31,SAVR11                      "
SHOWLV  LI    VDPADD,>0022                    "
        LI    WCOUNT,2                        "
        LI    WLOC,SAVR11                     "
        BL    @WRITE                          "
LEVTWY  CLR   SAVR11                          "
        CLR   WLOC                    CLEAR WRITE LOCATION
        BL    @SHOMAN                 SHOW NUMBER OF MUNCH-MAN REMAINING
***
***  SAVE ENTIRE SCREEN FOR RESTORATION LATER
***
        CLR   R10
        CLR   R4
        CLR   R14
        LI    R4,0
        LI    R14,>1000
        LI    RLOC,REGALB
        LI    WLOC,REGALB
        LI    RCOUNT,1
READON  MOV   R4,VDPADD
        BL    @READ
        MOV   R14,VDPADD
        BL    @WRITE
        INC   R4
        INC   R14
        CI    R4,769
        JNE   READON
***
***     RANDOMIZE 0 - 11 HERE
***
        MOVB  H00,CLRSCN
        BL    @RANDNO
        ANDI  R14,>00F0               0 - 15 POSSIBLE
        SRL   R14,4
        CI    R14,11
        JH    CHG13
        JMP   NOCH13
CHG13   S     H0005,R14
NOCH13  MOV   R14,POINT
        MOV   POINT,R4
        MPY   H0006,R4
        MOV   R5,POINT                RANDOM NUMBER 0-11 MULT BY 6 FOR INDEXING
        MOVB  H00,SNDFLG              SET SOUND CHIMES COUNTER
        MOVB  H00,INDX                SET TIME-OUT FOR DEMO COUNTER
        CB    DEMOFG,H01              GAME DEMING?
        JNE   MORERY                  NO , SO NO DEMO
        B     @GOOOOO                 YES, SO DEMO
MORERY  MOVB  H00,CLRSCN              RESET SCREEN TIME-OUT COUNTER
        BL    @RANDNO                 RANDOMIZE
        AB    H01,SNDFLG              INCREASE SOUND COUNTER
        CB    SNDFLG,H42              TIME TO TURN OFF CHIMES?
        JNE   NOOFF1                  NOT YET
        LI    R10,SNDCH1              SOUND THE LAST
        BL    @SOUND                  CHIME NOW
        MOVB  H00,SNDFLG              RESET SOUND COUNTER
        AB    H01,INDX                INCREASE CHIME OFF TIME FACTOR
        CB    INDX,H05                FIVE SETS OF CHIME PROMPTS YET?
        JNE   NOOFF1                  NO,  DON'T GO BACK TO TITLE SCREEN
        B     @BACK                   YES, GO BACK TO TITLE SCREEN
NOOFF1  CB    SNDFLG,H3C              TIME FOR CHIMES?
        JNE   NOCHIM                  NOT YET
```

```
        LI    R10,SNDCHM          SOUND THE
        BL    @SOUND              CHIMES NOW
NOCHIM  MOVB  H00,KEYBRD          KEYBOARD = ZERO
        BL    @SCANKY             SCAN IT
        CB    KEY,HFF             NO KEY YET?
        JEQ   CHKKY1              NOP
        CB    KEY,H0F             KEY PRESSED "BACK"?
        JNE   NOBACK              NO
        B     @BACK               YES
NOBACK  CB    KEY,H06             KEY PRESSED "REDO"?
        JNE   JUMP20              NO, ANY KEY WILL START THE GAME
        B     @REDO               "REDO" PRESSED
CHKKY1  AB    H01,KEYBRD          KEYBOARD = ONE
        BL    @SCANKY             SCAN IT
        CB    KEY,HFF             ANY KEY PRESSED?
        JNE   JUMP20              START GAME
        C     JOYY,H0000          JOYSTICK MOVED?
        JNE   JUMP20              YES, SO START GAME
        AB    H01,KEYBRD          KEYBOARD = TWO
        BL    @SCANKY             SCAN IT
        CB    KEY,HFF             ANY PRESSED?
        JNE   JUMP20              START GAME
        C     JOYY,H0000          JOYSTICK MOVED?
        JNE   JUMP20              YES, SO START GAME
        MOVB  H00,TIMER           DELAY
DELY20  CB    TIMER,H30           DELAY
        JNE   DELY20              DELAY
        LI    VDPADD,745          WRITE OUT READY MSG
        LI    WCOUNT,14
        LI    WLOC,READYM
        BL    @WRITE
        LI    VDPADD,326
        LI    WCOUNT,1
        LI    WLOC,H20
MUNOFF  BL    @WRITE              NO MUNCH
        AI    VDPADD,32
        CI    VDPADD,486
        JNE   MUNOFF
        LI    VDPADD,377
MANOFF  BL    @WRITE              NO MAN
        AI    VDPADD,32
        CI    VDPADD,505
        JNE   MANOFF
        LI    VDPADD,>038D
        LI    WCOUNT,1
        LI    WLOC,D6F            ENERGIZER COLORS - ALL DRED
        BL    @WRITE
        LI    VDPADD,>038F
        BL    @WRITE
        LI    VDPADD,>0391
        BL    @WRITE
        LI    VDPADD,>0393
        BL    @WRITE
        LI    VDPADD,>0395
        BL    @WRITE
        MOVB  H00,KEYBRD          KEYBOARD = ZERO
        BL    @SCANKY             SCAN IT
        CB    KEY,HFF             NO KEY YET?
        JEQ   CHKKE1              NOP
        CB    KEY,H0F             KEY PRESSED "BACK"?
        JNE   BACKNO              NO
        B     @BACK               YES
```

```
JUMP20  B     @GOOOOO              START THE GAME
BACKNO  CB    KEY,H06              KEY PRESSED "REDO"?
        JNE   GOOOOO               NO,  ANY KEY WILL START THE GAME
        B     @REDO                YES, "REDO" PRESSED
CHKKE1  AB    H01,KEYBRD           KEYBOARD = ONE
        BL    @SCANKY              SCAN IT
        CB    KEY,HFF              ANY KEY PRESSED?
        JNE   GOOOOO               START GAME
        C     JOYY,HOOOO           JOYSTICK MOVED?
        JNE   GOOOOO               YES, SO START GAME
        AB    H01,KEYBRD           KEYBOARD = TWO
        BL    @SCANKY              SCAN IT
        CB    KEY,HFF              ANY PRESSED?
        JNE   GOOOOO               START GAME
        C     JOYY,HOOOO           JOYSTICK MOVED?
        JNE   GOOOOO               YES, SO START GAME
        MOVB  HOO,TIMER            DELAY
DELO2O  CB    TIMER,H30            DELAY
        JNE   DELO2O               DELAY
        CLR   R4
        LI    VDPADD,326
        LI    WCOUNT,1
MORAI1  AI    R4,XMUNCH            MUNCH
        MOV   R4,WLOC
        BL    @WRITE
        LI    R14,XMUNCH
        S     R14,R4
        INC   R4
        AI    VDPADD,32
        CI    VDPADD,486
        JNE   MORAI1
        LI    VDPADD,377
        CLR   R4
MORAI2  AI    R4,XMAN
        MOV   R4,WLOC              MAN
        BL    @WRITE
        LI    R14,XMAN
        S     R14,R4
        INC   R4
        AI    VDPADD,32
        CI    VDPADD,505
        JNE   MORAI2
        LI    VDPADD,>038D
        LI    WCOUNT,1
        LI    WLOC,D3F             ENERGIZER COLORS - ALL GREEN
        BL    @WRITE
        LI    VDPADD,>038F
        BL    @WRITE
        LI    VDPADD,>0391
        BL    @WRITE
        LI    VDPADD,>0393
        BL    @WRITE
        LI    VDPADD,>0395
        BL    @WRITE
        LI    VDPADD,745
        LI    WCOUNT,14
        LI    WLOC,VBLANK          NO   ARE YOU READY?   MSG
        BL    @WRITE
        B     @MORERY              CONTINUE MSG READY LOOP
GOOOOO  MOVB  HOO,KEYBRD           KEY DEFAULT = ZERO
***
***  RESTORE ENTIRE SCREEN FROM LATER SAVE
```

```
***
        CLR   R10
        CLR   R4
        CLR   R14
        LI    R4,0
        LI    R14,>1000
        LI    RLOC,REGALB
        LI    WLOC,REGALB
        LI    RCOUNT,1
READMR  MOV   R14,VDPADD
        BL    @READ
        MOV   R4,VDPADD
        BL    @WRITE
        INC   R4
        INC   R14
        CI    R4,769
        JNE   READMR
***
***   BLANK OUT "ARE YOU READY?" MSG
***   PUT OUT MUNCH MAN LOGO
***   TURN ENERGIZERS BACK TO GREEN
***   STOP ALL SOUNDS
***
        CLR   R4
        LI    VDPADD,326
        LI    WCOUNT,1
MORAI3  AI    R4,XMUNCH             MUNCH
        MOV   R4,WLOC
        BL    @WRITE
        LI    R14,XMUNCH
        S     R14,R4
        INC   R4
        AI    VDPADD,32
        CI    VDPADD,486
        JNE   MORAI3
        LI    VDPADD,377
        CLR   R4
MORAI4  AI    R4,XMAN
        MOV   R4,WLOC               MAN
        BL    @WRITE
        LI    R14,XMAN
        S     R14,R4
        INC   R4
        AI    VDPADD,32
        CI    VDPADD,505
        JNE   MORAI4
        LI    VDPADD,745
        LI    WCOUNT,14
        LI    WLOC,VBLANK           NO   ARE YOU READY?   MSG
        BL    @WRITE
        LI    VDPADD,>038D
        LI    WCOUNT,1
        LI    WLOC,D3F              ENERGIZER COLORS - ALL GREEN
        BL    @WRITE
        LI    VDPADD,>038F
        BL    @WRITE
        LI    VDPADD,>0391
        BL    @WRITE
        LI    VDPADD,>0393
        BL    @WRITE
        LI    VDPADD,>0395
        BL    @WRITE
```

```
        MOVB  DEMOFG,DEMOFG       DEMOING GAME STILL?
        JEQ   NODEMM
        LI    VDPADD,739          WRITE PRESS ANY ... MSG AT BOTTOM
        LI    WCOUNT,26
        LI    WLOC,MSGBEG         "PRESS ANY KEY TO BEGIN" MSG
        BL    @WRITE
NODEMM  LI     R10,SNDOFF          TURN OFF
        BL     @SOUND              ALL SOUNDS
***
***  ORIGINAL 9900 ASSEMBLY CODE ENTRY FROM GPL PROGRAM
***
XMLO1   MOVB  H00,ESCAPE          SET TO NO ESCAPE FROM DEMO YET
        MOVB  H00,CLRSCN          RESET SCREEN TIME-OUT COUNTER
        MOVB  RANDOM,RAND16+1     RANDOMIZE FOR THE DEMO MODE
        CB    DEMOFG,H01          GAME IN DEMO MODE?
        JEQ   DEMING              YES, SO KEEP RANDOM NUMBER SET ABOVE
        CLR   RAND16              NO,  SO CLEAR RANDOM NUMBER SEED
DEMING  CB    TITLFG,H00          TITLE SCREEN UP?
        JEQ   NOTITL              NO, GO PROCESS THE GAME
        MOVB  H00,KEYBRD
        CLR   R10
KEYON   BL    @SCANKY
        MOVB  STATUS,R10
        JNE   KEYON
        CLR   R4
        CLR   R10
        CLR   R13
        CLR   R15
        MOV   H0090,YXLOC1
        MOV   H0200,WAITMN
        MOV   WAITMN,WAITPC
        LI    RCOUNT,8
MORCHR  MOV   TABMON(R4),R14
        A     R13,R14
        MOV   R14,VDPADD
        LI    RLOC,YXLOCP
        BL    @READ
        MOV   TABMON(R4),VDPADD
        LI    WLOC,YXLOCP
STAL03  DEC   WAITPC
        JNE   STAL03
        MOV   WAITMN,WAITPC
        BL    @WRITE
        DEC   YXLOC1
        JNE   NOWINK
        MOV   H0090,YXLOC1
        LI    VDPADD,>0025
        LI    WLOC,WINDT1
        A     R15,WLOC
        BL    @WRITE
        LI    VDPADD,>0035
        BL    @WRITE
        AI    R15,8
        CI    R15,32
        JNE   NOWINK
        CLR   R15
NOWINK  MOVB  H00,CLRSCN
        BL    @SCANKY
        MOVB  STATUS,R10
        JNE   JUMP10
        INCT  R4
        CLR   R14
```

```
          CI    R4,14
          JNE   MORCHR
          CLR   R4
          AI    R13,8
          CI    R13,64
          JNE   MORCHR
          CLR   R13
          C     WAITMN,H0060
          JH    INIT08
          DEC   WAITMN
          JMP   INIT1
JUMP10    BL    @GOLINK                 CLOSE CHAIN LINK
          B     @GOOGAM                 START THE GAME
INIT08    S     H0010,WAITMN
INIT1     MOV   WAITMN,WAITMN
          JEQ   REINIT
          JMP   GOON1
REINIT    MOVB  H01,DEMOFG
          JMP   JUMP10                  QUIT THE MOVEMENT NOW
GOON1     MOV   WAITMN,WAITPC
          JMP   MORCHR
NOTITL    CB    SCRNUB,H00              1ST ROUND?
          JEQ   ROUND0                  NEXT ROUND IF SCRNUB = 1 --->
          MOV   H00B0,DELTA0            22 CHAR
          MOV   H00A0,DELTA1            20 CHAR
          MOV   H0098,DELTA2            19 CHAR
          MOV   H0090,DELTA3            18 CHAR
          CB    SCRNUB,H01              2ND ROUND?
          JEQ   ROUND1                  YES SCRNUB=1
          MOVB  SCRNUM,R4               3RD ROUND(SCRNUB=2)
          SRL   R4,8
          MPY   H0006,R4
          MOV   TABLEA(R5),ENGSAV
          INCT  R5
          MOV   TABLEA(R5),WAITPC
          INCT  R5
          MOV   TABLEA(R5),WAITMN
          JMP   CONTMN
ROUND1    MOVB  SCRNUM,R4               2ND ROUND(SCRNUB=1)
          SRL   R4,8
          MPY   H0006,R4
          MOV   TABLE9(R5),ENGSAV
          INCT  R5
          MOV   TABLE9(R5),WAITPC
          INCT  R5
          MOV   TABLE9(R5),WAITMN
CONTMN    MOV   WAITMN,SAVWMN
          MOV   HEA60,WAITMN            60,000 MONSTER BOX-DELAY-TIME
          JMP   START1
ROUND0    MOVB  SCRNUM,R4               SCREEN NUMBER(0-19) HB
          SRL   R4,8                    SCREEN NUMBER(0-19) LB
          MPY   H000A,R4                POINT TO DATA
          MOV   TABLE8(R5),ENGSAV        GET ENERGIZER TIME FACTOR
          INCT  R5
          MOV   TABLE8(R5),DELTA0       GET MONSTER RADIUS 0
          MOV   DELTA0,DELTA1
          MOV   DELTA0,DELTA2
          MOV   DELTA0,DELTA3
          S     H0010,DELTA1
          S     H0018,DELTA2
          S     H0020,DELTA3
          INCT  R5
```

```
        MOV   TABLE8(R5),WAITPC
        INCT  R5
        MOV   TABLE8(R5),WAITMN
        MOV   WAITMN,SAVWMN
        INCT  R5
        MOV   TABLE8(R5),WAITMN
START1  MOV   WAITPC,SAVWPC
        MOV   WAITPC,STALPC
        MOV   H0000,SAVINC      INIT MMAN TO SAVE "UP"
        MOV   H0700,CHGCNT      INIT ENERGIZER CHANGE PATTERN COUNTER
        MOVB  H00,FLAGZZ        INIT TO NO ENERGIZER HIT
        CLR   ENGPTN            SET ENERGIZER PATTERN TO 1ST PATTERN
        LI    VDPADD,>0300      INIT VDP ADDRESS
        LI    RLOC,YXLOCP       SET READ LOCATION W/ YPT,XPT
        LI    RCOUNT,20         READ IN FIRST 4 SAB'S
        BL    @READ             GO READ VDP
LOOPO1  DEC   CHGCNT                  DECREMENT ENGZ FLASH COUNTER
        JNE   CHKTM2                  ZERO YET?IF NOT,CHECK MMAN TIMER
        MOVB  H00,CLRSCN              RESET SCREEN TIME-OUT COUNTER
        LI    RCOUNT,1                VDP READ,WRITE COUNT = ONE
        MOV   H0700,CHGCNT            INIT ENGZ CHANGE PATTERN COUNTER
        CLR   TEMPR4                  INIT POINTER TO ENGX VDP POSTION
        MOV   ENGPTN,COUNT            RESTORE CURRENT ENGZ PATTERN POINTER
        CI    COUNT,10                LAST PATTERN(5TH) ALREADY?
        JNE   CHGENG                  NO,  GO GET NEXT ENGZ PATTERN
        CLR   ENGPTN                  YES, SET POINTER TO LAST PATTERN
CHGENG  MOV   TABLE2(TEMPR4),VDPADD   SET VDP BUFFER W/ 1ST ENGZ LOCATION
        CLR   TEMPRA                  CLEAR READ LOCATION REG10
        LI    RLOC,REGALB             ESTABLISH RETURN LOCATION FOR READ
        BL    @READ                   GO READ THE ENGZ CHARACTER PATTERN
        CI    TEMPRA,>00CC            ALREADY A BLANK(EATEN)?
        JEQ   NEXTPT                  YES, GO CHECK OTHER ENGZ
        MOV   ENGPTN,COUNT            RESTORE REG5(POINTER TO ENGZ PATTERN)
        MOV   TABLE2(TEMPR4),VDPADD   LOAD VDP BUFFER W/ 1ST ENERGIZER LOC
        MOV   TABLE3(COUNT),TEMPRA    GET NEW ENGZ PATTERN
        LI    WLOC,REGALB             LOAD IN LOWER BYTE OF PATTERN
        BL    @WRITE                  GO WRITE THE NEW ENGZ PATTERN
NEXTPT  INCT  TEMPR4                  INCREMENT TO NEXT ENGZ POSITION
        CI    TEMPR4,8                PAST LAST ONE(4TH)?
        JEQ   DONEGZ                  YES, SO CONTINUE
        JMP   CHGENG                  NO,  CHANGE THE REST OF THE ENGZS
DONEGZ  INCT  ENGPTN                  SAVE NEXT ENGZ PATTERN POINTER
CHKTM2  DEC   WAITPC            DECREASE MMAN WAITING
        JNE   CHKTM1            NOT YET, GO CHECK MONSTER'S TIMER
        MOV   SAVWPC,WAITPC     RESTORE WAITING FACTOR FOR MMAN
        BL    @MOVMAN           GO MOVE MMAN
        LI    VDPADD,>0300      INIT VDP ADDRESS
        LI    WLOC,YXLOCP       SET WRITE LOCATION W/ YPT,XPT
        LI    WCOUNT,4          WRITE FIRST 4 SAB'S
        BL    @WRITE            GO WRITE TO VDP
CHKTM1  DEC   WAITMN            DECREASE MONSTER WAITING
        JNE   LOOPO1            NOT YET, GO CHECK MMAN'S TIMER
        MOV   SAVWMN,WAITMN     RESTORE WAITING FACTOR FOR MONSTERS
        CLR   MON               INIT TO MONSTER #0
CHKMON  BL    @MOVEYX           GO CHECK FOR MONSTER MOVEMENT
        CLR   R4                CLEAR REG4
        MOVB  YXLOCO(MON)+2,R4  GET MONSTER CHAR FROM SAB(98-9B)
        SWPB  R4
        INC   R4                NEXT CHAR
        ORI   R4,>0018
        ANDI  R4,>009F          IF CHAR=A0, THEN BACK TO 98 CHAR
        SWPB  R4
```

```
        MOVB  R4,YXLOCO(MON)+2 PUT NEW MONSTER CHAR BACK TO SAB
        AI    MON,4            NEXT MONSTER
        CI    MON,16           PAST THE 4TH MONSTER?
        JL    CHKMON           NOT YET, SO KEEP CHECKING
        LI    VDPADD,>0304     INIT VDP ADDRESS
        LI    WLOC,YXLOCO      SET WRITE LOCATION W/ YPT,XPT
        LI    WCOUNT,16        WRITE FIRST 4 SAB'S
        BL    @WRITE           GO WRITE TO VDP
        JMP   LOOP01           PAST 4TH MONSTER, SO LOOP BACK

*********************************
*  SOME GAME INITIALIZATIONS    *
*********************************
INIT01  MOV   H5F70,YXPIT0
        MOV   H5F78,YXPIT1
        MOV   H5F80,YXPIT2
        MOV   H5F88,YXPIT3
        MOV   H000C,PROB00
        MOV   H000C,PROB01
        MOV   H000D,PROB02
        MOV   H000D,PROB03
INIT02  MOV   H0001,INCR0
        MOV   H0001,INCR1
        MOV   H00FF,INCR2
        MOV   H00FF,INCR3
        CLR   INCMAN
        B     *R11

*****************************************************
*  INITS FOR MONSTER PATTERNS, MAZE COLORS,         *
*  AND ENERGIZER TYPES FOR SCREENS #0-19            *
*****************************************************
INIT03  MOV   R11,SAVR11              SAVE RETURN ADDR
        CLR   R4                      CLEAR OUT REG4 FOR MOVE BYTE
        MOVB  SCRNUM,R4               GET SCREEN NUMBER
        SWPB  R4                      SCREEN NUMBER IN LOW END OF R4
        MOV   R4,R14                  SAVE SCREEN NUMBER
        MPY   H0040,R4                R5 CONTAINS THE PRODUCT
        LI    VDPADD,>04C0            LOAD VDP WRITE ADDR TO MONSTER DATA LOC
        LI    WCOUNT,64               64 BYTES OF MONSTER DATA(8X8)
        AI    R5,DMONA                ADD TO OFFSET , THE ADDR OF MONSTER DATA
        MOV   R5,WLOC                 LOAD MONSTER DATA ADDR IN WRITE LOCATION
        BL    @WRITE                  WRITE IN NEW MONSTER DATA
        CLR   R13                     H0003 MUST BE > R13(R14=SCREEN# 0 -19)
        DIV   H0003,R13               R13=QUOT: 0-6, R14=REMAIN:0-2
        CB    SCRNUM,H13              ON SCREEN 19?
        JNE   NONINT                  NO, GO ON
        AI    R13,1                   MAKE REG13 = 7, NOT = 6 TO GET >FF COLOR
NONINT  MOV   R13,R8                  SAVE VALUE OF INDEX FOR CHAIN DATA
        AI    R13,CLRDAT              ADD ADDR FOR PCT COLOR TO TABLE INDX-R13
        AI    R8,CHADAT               CHAIN DATA
        MOV   R13,WLOC                LOAD MAZE COLOR ADDR IN WRITE LOCATION
        LI    WCOUNT,1                WRITE ONE COLOR
        LI    VDPADD,>0380            PCT TABLE(0)
MORTAB  BL    @WRITE                  GO CHANGE COLOR NOW
        INC   VDPADD                  NEXT PCT TABLE
        CI    VDPADD,>0386            JUST PCT'S 0-5 ONLY
        JNE   MORTAB                  LOOP
        LI    VDPADD,>039C            PCT TABLE(28)
        BL    @WRITE                  GO CHANGE COLOR NOW
        LI    VDPADD,>039D            PCT TABLE(29)
        BL    @WRITE                  GO CHANGE COLOR NOW
```

```
        MOV   R8,WLOC              INDEX INTO CHAIN DATA - DCHAIN
        LI    VDPADD,>0398         PCT TABLE(24) FOR CHAINS
        BL    @WRITE               WRITE OUT CHAIN COLOR
        INC   VDPADD               PCT TABLE(25) FOR CHAINS
        BL    @WRITE               WRITE OUT CHAIN COLOR
        MOV   R14,R4               LOAD IN 0 - 2
        MPY   H000A,R4             SET POINTER INTO ENZ ADDR DATA TAB
        MOV   R5,R14               SAVE POINTER SINCE WRITE CALL USES R5
        LI    VDPADD,>0B60         START W/ CHAR >6C
        LI    WCOUNT,8             8 BYTES FOR DATA FOR EACH ENERGIZER
MORENZ  MOV   ENG00(R14),WLOC      GET ADDESSES
        BL    @WRITE               GET DATA FOR ENZ AND WRITE IT
        INCT  R14                  INCREMENT THROUGH >7C - >AC
        AI    VDPADD,>0080         INCREMENT THROUGH >7C - >AC
        CI    VDPADD,>0DE0         PAST >AC CHAR?
        JNE   MORENZ               NOP, KEEP INITING ENZ
        MOV   SAVR11,R11           RESTORE RETURN ADDR
        B     *R11                 RETURN TO CALLER

*****************************
*  GET TITLE SCREEN INFO    *
*****************************
TILSCN  MOV   R11,SAVR11
        LI    R10,SNDOFF
        BL    @SOUND
***
***   BLANK OUT THE SCREEN
***
        LI    VDPADD,0
        LI    WCOUNT,1
MORH20  LI    WLOC,WINDT2          GET THE BLANKS(>20)
        BL    @WRITE
        INC   VDPADD
        CI    VDPADD,769
        JNE   MORH20
        LI    VDPADD,>0386         CHANGE PCT'S NOW
        LI    WCOUNT,6
        LI    WLOC,COLLGN
        BL    @WRITE
        LI    VDPADD,>0395
        LI    WCOUNT,4
        LI    WLOC,COLTB1
        BL    @WRITE
        LI    VDPADD,>039D
        LI    WCOUNT,1
        LI    WLOC,COLMAG
        BL    @WRITE
***
***   INIT MUNCH-MAN "P", TEETH, EYES
***
        LI    VDPADD,>0B98
        LI    WCOUNT,24
        LI    WLOC,DPC2RT
        BL    @WRITE
        LI    VDPADD,>0BB0
        LI    WCOUNT,16
        BL    @WRITE
***     LI    VDPADD,>0DE0
***     LI    WCOUNT,72
***     LI    WLOC,DTEE1
***     BL    @WRITE
***     LI    VDPADD,>0D40
```

```
***     LI    WCOUNT,16
***     LI    WLOC,DEYE1
***     BL    @WRITE
        MOVB  H01,TITLFG          SET TITLE SCREEN FLAG ON(UP)
        MOVB  H33,@>8C02         LGREEN BACKGROUND
        MOVB  H87,@>8C02         LGREEN BACKGROUND
***
***   DISPLAY TITLE SCREEN W/ "Munch-Man" logo
***
        CLR   VDPADD              SET YPT,XPT POINTER
        LI    WCOUNT,768
        LI    WLOC,TITLES
        BL    @WRITE
***
***   DELETE ALL SPRITES IF REMAINING
***
        LI    VDPADD,>0300
        LI    WCOUNT,1
        LI    WLOC,HD0
        BL    @WRITE
        MOV   SAVR11,R11
        B     *R11                RETURN TO CALLER

*****************************************
*  ROUTINE THAT CLOSES THE CHAIN LINK   *
*****************************************
GOLINK  MOV   R11,SAVR11
        LI    R10,SNDOFF          TURN OFF ALL
        BL    @SOUND              SOUNDS
        LI    VDPADD,>0398
        LI    WCOUNT,2
        LI    WLOC,H4F4F
        BL    @WRITE
        LI    VDPADD,548
        LI    WLOC,HC5            DR1
        LI    WCOUNT,1
        BL    @WRITE
        BL    @WAIT01
        LI    WLOC,HC3            LR1
LINK01  INC   VDPADD
        BL    @WRITE
        BL    @WAIT01
        CI    VDPADD,571
        JNE   LINK01
        LI    WLOC,HC6            DL1
        BL    @WRITE
        BL    @WAIT01
        LI    WLOC,HCC            UD1
        AI    VDPADD,32
        BL    @WRITE
        BL    @WAIT01
        LI    WLOC,HCA            UL1
        AI    VDPADD,32
        BL    @WRITE
        BL    @WAIT01
        LI    WLOC,HC3            LR1
LINK02  DEC   VDPADD
        BL    @WRITE
        BL    @WAIT01
        CI    VDPADD,612
        JNE   LINK02
        LI    WLOC,HC9            UR1
```

```
        BL    @WRITE
        BL    @WAIT01
        AI    VDPADD,-32
        LI    WLOC,HCC              UD1
        BL    @WRITE
        BL    @WAIT40
        MOV   SAVR11,R11
        B     *R11

************************
*  DELAY SUBROUTINES   *
************************
WAIT01  MOV   R11,RA+0
        MOVB  H00,TIMER
PAUS01  CB    TIMER,H01
        JNE   PAUS01
        MOV   RA+0,R11
        B     *R11
************************
WAIT40  MOV   R11,RA+0
        MOVB  H00,TIMER
PAUS40  CB    TIMER,H40
        JNE   PAUS40
        MOV   RA+0,R11
        B     *R11

****************************
*  COINCIDENCE DETECTED    *
****************************
COINCI  MOV   R11,RA+4          SAVE REG11
        CB    FLAGZZ,H00        HIT ENERGIZER?
        JEQ   QUIT01            NO,  SO QUIT THIS ROUND
        MOVB  H00,TIMER         SET TIMER
DELAY0  CB    TIMER,H10
        JNE   DELAY0
        MOVB  HC0,YXLOC0(MON)+2     GET SPLAT FOR MONSTER
        LI    VDPADD,>0304          WRITE OUT TO VDP MON SAB'S
        LI    WLOC,YXLOC0
        LI    WCOUNT,16
        BL    @WRITE
        MOVB  H00,TIMER                SET TIMER
DELAYA  CB    TIMER,H05                DELAY 83/60 SECS
        JNE   DELAYA                   DELAY 83/60 SECS
        CLR   R10
        MOVB  CAPPTS,R10        GET NUMBER OF CAPTURE POINTS
        BL    @SCORE            GO INCREASE SCORE
        LI    R10,SNDEAT        GET MONSTER
        BL    @SOUND            EATEN SOUND
        CLR   R10               ZERO REG10
        MOVB  CAPPTS,R10        MOVE 01,02,04, OR 08 TO REG10
        MOV   R10,R0            MOVE 01,02,04, OR 08 TO REG0
        SLA   R0,5
        LI    WCOUNT,3          WRITE THREE BYTES(100,200,400,800)
        JOC   GOT800            4TH CAPTURE
        SLA   R0,1
        JOC   GOT400            3RD CAPTURE
        SLA   R0,1
        JOC   GOT200            2ND CAPTURE
        A     H0064,SCRSAV      1ST CAPTURE, ADD 100 POINTS
        LI    WLOC,MSG100
        LI    VDPADD,329
        JMP   SLAIT
```

```
GOT800  A     H0320,SCRSAV      ADD 800 POINTS
        LI    WLOC,MSG800
        LI    VDPADD,468
        JMP   SLAIT
GOT400  A     H0190,SCRSAV      ADD 400 POINTS
        LI    WLOC,MSG400
        LI    VDPADD,457
        JMP   SLAIT
GOT200  A     H00C8,SCRSAV      ADD 200 POINTS
        LI    WLOC,MSG200
        LI    VDPADD,340
SLAIT   BL    @WRITE            WRITE CAPTURE POINTS IN BLOCKS
        MOV   R10,R0
        SLA   R0,1                    DOUBLE CAPTURE POINTS FOR NEXT TIME
        MOVB  R0,CAPPTS               MOVE BACK 02,04,08,10
        MOVB  H98,YXLOC0(MON)+2       RESTORE MONSTER CHARACTER
        MOV   YXPIT0(MON),YXLOC0(MON)       MONSTER BACK IN PIT
        BL    @CHKPTS                 10,000 POINT EXTRA MMAN CHECK
        BL    @WRITE                  GO WRITE EXTRA MMAN
        MOV   R4+4,R11                RESTORE RETURN ADDRESS
        B     *R11                    RETURN TO CALLER
QUIT01  CB    WINFLG,H01              WIN FLAG ON(MMAN FINISH)?
        JEQ   SPINIT                  YES, SO JUST QUIT
        MOVB  H00,TIMER               NO, MMAN EATEN
DELAY1  CB    TIMER,H05
        JNE   DELAY1
        MOV   YXLOCP,YXLOC0(MON)      MOVE MONSTER OVER MMAN
        MOVB  HC0,YXLOCP              GET RID OF MMAN
        LI    VDPADD,>0300            WRITE OUT TO VDP MON AND MMAN SAB'S
        LI    WLOC,YXLOCP
        LI    WCOUNT,20
        BL    @WRITE
        MOVB  YXLOCP+3,YXLOC0(MON)+3  GET MMAN COLOR
        LI    R10,SNDEND              MELT DOWN
        BL    @SOUND                  SOUND
        LI    R10,>00C1
MOREDY  MOVB  H00,TIMER               SET TIMER
DELAY2  CB    TIMER,H0A               DELAY 83/60 SECS
        JNE   DELAY2                  DELAY 83/60 SECS
        MOVB  REGALB,YXLOC0(MON)+2    CHANGE MONSTER TO EATEN MMAN
        BL    @WRITE                  WRITE NEW PATTERN TO VDP
        INC   R10                     GET NEXT DYING MMAN PATTERN
        CI    R10,>00CB               THAT'S ALL THE PATTERNS?
        JNE   MOREDY                  NOP, KEEP DYING
        JMP   NODLAY                  CONTINUE ON OUT
SPINIT  LI    R10,SNDWIN              GET MUNCH-MAN
        BL    @SOUND                  WIN SOUND
        MOVB  H06,RANDOM
        LI    VDPADD,>0302
        LI    WLOC,YXLOCP+2
        LI    WCOUNT,1
        MOVB  YXLOCP+2,R4
        ANDI  R4,>F000
        MOVB  R4,YXLOCP+2
        MOVB  YXLOCP+2,R10
SPIN    BL    @WRITE
        SRL   R4,8
        AI    R4,>0010
        CI    R4,>00C0
        JEQ   NOSPIN
        SWPB  R4
MORSPI  MOVB  R4,YXLOCP+2
```

```
        CB    RANDOM,HFF
        JEQ   TIMEFF
        MOVB  H00,TIMER
DELAY3  CB    TIMER,RANDOM
        JNE   DELAY3
        JMP   SPIN
NOSPIN  LI    R4,>8000
        SB    H01,RANDOM
        JMP   MORSPI
TIMEFF  MOVB  R10,YXLOCP+2
        BL    @WRITE
NODLAY  BL    @BNKNUM           CLEAR CAPTURE POINTS
        BL    @MAZCOL           RESTORE ORIGINAL MAZE COLOR
        LI    VDPADD,>38D       RESTORE ENERGIZERS BACK TO GREEN
        LI    WLOC,COLGRN
        LI    WCOUNT,1
MORGRE  BL    @WRITE
        INCT  VDPADD
        CI    VDPADD,>397
        JNE   MORGRE
***
***     PREVIOUSLY RETURN TO GPL PROGRAM HERE
***
DEMOUT  MOVB  H00,TIMER
DEL20   CB    TIMER,H20
        JNE   DEL20
***
***  RE-INIT HOONO BOXES IMMEDIATELY
***
        LI    VDPADD,330
        LI    WCOUNT,2
        LI    WLOC,HB1B1
        BL    @WRITE
        AI    VDPADD,128
        BL    @WRITE
        LI    VDPADD,340
        LI    WLOC,HB2B2
        BL    @WRITE
        AI    VDPADD,128
        BL    @WRITE
        LI    WLOC,CENTER
        LI    WCOUNT,6
        LI    VDPADD,397
        BL    @WRITE
        BL    @RANDNO           RANDOMIZE
        MOVB  H00,KEYBRD
        LI    VDPADD,>0300
        LI    WCOUNT,21
        LI    WLOC,SABMAN
        BL    @WRITE
        CB    ESCAPE,H01
        JNE   NOFLSH             NO ESCAPE FROM DEMO YET
        B     @GOFLSH            ESCAPE FROM DEMO AND START REAL GAME
NOFLSH  CB    WINFLG,H01
        JNE   GAMLOS            WIN FLAG NOT SET
        B     @GAMWON           WIN FLAG SET
GAMLOS  BL    @INIT02
        SB    H01,COUNT1        ONE LESS MUNCH-MAN
        CB    COUNT1,H00        NO MORE MEN LEFT?
        JEQ   NOCONT            NO,  SO GAME OVER NOW
        B     @CONT01           YES, "S A M E  S C R E E N"
NOCONT  CLR   WLOC
```

```
        BL    @SHOMAN             SHOW NUMBER OF MUNCH-MAN REMAINING
        LI    VDPADD,334          "G A M E   O V E R "
        LI    WCOUNT,4
        LI    WLOC,MSGGAM
        BL    @WRITE
        LI    VDPADD,462
        LI    WLOC,MSGOVE
        BL    @WRITE
***
***     COMPARE YOUR SCORE WITH BEST SCORE
***
        MOV   H0011,BESTSC
        MOV   H0031,YOURSC
CHKTOP  MOV   BESTSC,VDPADD
        LI    RCOUNT,1
        LI    RLOC,REGALB         REG10 = 1ST DIGIT OF BEST SCORE
        BL    @READ
        MOV   YOURSC,VDPADD
        LI    RLOC,REG4LB         REG4  = 1ST DIGIT OF YOUR SCORE
        BL    @READ
        C     R10,R4              NO HIGH SCORE?
        JH    NOTOP               CORRECT
        C     R4,R10              HIGH SCORE?
        JEQ   NODIFF              NO, DIGITS =
        LI    RCOUNT,6            GOT HIGH SCORE!!
        LI    RLOC,YXLOCP
        LI    VDPADD,49
        BL    @READ               GO READ YOUR SCORE
        LI    WLOC,YXLOCP
        LI    VDPADD,>2000
        BL    @WRITE              GO WRITE NEW HIGH SCORE(YOUR'S) TO HIGH VDP
        BL    @PAUSOA             DELAY
        MOVB  H05,INDX            INIT LOOP COUNTER
MORFLH  LI    VDPADD,49           BLANK HIGH(YOUR'S) SCORE
        LI    WCOUNT,6
        LI    WLOC,MSGBLK
        BL    @WRITE              BLANK IT OUT
        BL    @PAUSOA             DELAY
        LI    VDPADD,>2000        GET YOUR SCORE(HIGH SCORE)
        LI    RLOC,YXLOCP
        BL    @READ               READ IT YOUR SCORE
        LI    VDPADD,49
        LI    WLOC,YXLOCP
        BL    @WRITE              WRITE YOUR SCORE BACK AGAIN
        LI    R10,SNDREW          SOUND THE
        BL    @SOUND              REWARD BELL
        BL    @PAUSOA             DELAY
        SB    H01,INDX            DEC LOOP COUNTER
        CB    INDX,H00            DONE YET?
        JNE   MORFLH              NO, MORE FLASHING TO GO
        LI    VDPADD,49           GET YOUR SCORE
        LI    RCOUNT,6
        LI    RLOC,YXLOCP
        BL    @READ               READ IT
        LI    VDPADD,17           GET SCREEN PLACE OF BEST SCORE
        LI    WLOC,YXLOCP
        BL    @WRITE              WRITE IN NEW HIGH SCORE
        JMP   NOTOP               CONTINUE
NODIFF  INC   BESTSC              NEXT DIGIT
        INC   YOURSC              NEXT DIGIT
        C     BESTSC,H0017        LAST DIGIT?
        JNE   CHKTOP              NO, KEEP CHECKING
```

```
NOTOP   MOVB DEMOFG,DEMOFG    QUIT CHECKING, DEMOING?
        JEQ  LOOP02           NO, SO CONTNUE ON
        MOVB H00,TIMER        DELAY
DEL2    CB   TIMER,H3C        DELAY
        JNE  DEL2             DELAY
        B    @BACK            IN DEMO SO GO BACK TO TITLE SCREEN DEMO
***
***  DELAY AND SCAN W/ TIME-OUT
***
LOOP02  CLR  R4
        MOVB H00,KEYBRD       KEYBOARD TYPE = ZERO
        MOV  H003C,R4         SET TO 60 SECONDS
MORTIM  MOVB H00,TIMER        SET DELAY LIMIT(1/60TH SEC) = 60
DELAYS  BL   @SCANKY          SCAN KEYBOARD
        CB   KEY,HFF          ANY KEY PRESSED?
        JNE  NODEL
        CB   TIMER,H3C
        JNE  DELAYS
        DEC  R4
        JNE  MORTIM
        B    @BACK            1 MINUTE DELAY
NODEL   CB   KEY,H06          "REDO"?
        JEQ  REDO
        CB   KEY,H0F          "BACK"?
        JNE  LOOP02           NO,  KEEPING CHECKING
        B    @BACK            YES, GO "BACK"
***
***  ZERO OUT THE SCORE HERE
***
REDO    LI   R10,SNDOFF       TURN OFF
        BL   @SOUND           ALL SOUNDS
        LI   VDPADD,49        VDP SCREEN YOUR SCORE LOCATION
        LI   WCOUNT,6         SIX ZEROS
        LI   WLOC,MSGZER      GET THE ZEROS
        BL   @WRITE           WRITE SIX ZEROS FOR SCORE
        MOVB H03,COUNT1       INIT TO THREE MEN
        MOVB H00,SCRNUM       INIT TO FIRST SCREEN
        MOVB H00,SCRNUB       INIT TO FIRST ROUND
        CLR  SCRSAV           INIT TO ZERO ACCUM POINTS
        B    @NEWGAM          START NEW GAME NOW
***
***  REWARD MESSAGE WRITTEN OUT NOW ON SCREEN
***
GAMWON  LI   VDPADD,>014D
        LI   WCOUNT,6
        MOV  POINT,R4         GET REWARD MSG POINTER
        AI   R4,MESS1         ADD THE OFFSET
        MOV  R4,WLOC          PUT THAT ADDR IN WRITE VDP LOCATION
        BL   @WRITE           WRITE 1ST REWARD MSG
        LI   VDPADD,>01CD
        MOV  POINT,R4         GET REWARD MSG POINTER
        AI   R4,MESS2         ADD THE OFFSET
        MOV  R4,WLOC          PUT THAT ADDR IN WRITE VDP LOCATION
        BL   @WRITE           WRITE 2ND REWARD MSG
        AB   H01,SCRNUM       SET TO NEXT SCREEN LEVEL
        BL   @INIT02          INIT INCREMENTS
        MOVB H00,TIMER        DELAY
DEL3    CB   TIMER,H3C        DELAY
        JNE  DEL3             DELAY
        B    @NXTGAM          GO TO NEXT SCREEN LEVEL
GOFLSH  BL   @TILSCN          GO GET TITLE SCREEN
        BL   @GOLINK          GO CLOSE CHAIN LINK
```

```
        B     @GOOGAM          START REAL GAME

*************************************
*  SHOW THE TOTAL NUMBER OF MUNCH-  *
*  MEN REMAINING W/ CHAR >74        *
*************************************
SHOMAN  MOV   R11,SAVR11         SAVE RETURN ADDR
        LI    VDPADD,57          POSTION FOR SHOWING REMAINING MUNCH-MEN
        LI    WCOUNT,4           WRITE OUT
        LI    WLOC,MSGBLK        4 BLANKS
        BL    @WRITE             TO INIT NUMBER SHOWING
        CB    COUNT1,H00         ANY MUNCH-MEN LEFT?
        JEQ   NOWRIT             NOP, SO LEAVE BLANK
        LI    WLOC,MSGMEN        SET LOCATION OF CHARACTERS >74
        CB    COUNT1,H05         5 OR MORE MEN LEFT?(4 OF MORE RESERVED)
        JHE   WRITIT             YES,  THEN SIMPLY SHOW 4 IN RESERVE
        MOVB  COUNT1,WCOUNT      ELSE, GET NUMBER REMAINING
        SRL   WCOUNT,8           PUT IN LOW END OF REG
        DEC   WCOUNT             LESS ONE FOR THE TRUE RESERVED NUMBER
        JEQ   NOWRIT             IF RESERVE COUNT=0, THEN NO SHOW
WRITIT  BL    @WRITE             GO WRITE IT TO VDP
NOWRIT  MOV   SAVR11,R11         RESTORE RETURN ADDR
        B     *R11               RETURN TO CALLER

*********************************
*  ROUTINE THAT CHECKS FOR      *
*  POSSIBLE MONSTER MOVEMENT    *
*********************************
MOVEYX  MOV   R11,RA+12           SAVE RETURN ADDRESS
        CB    FLAGZZ,H00          ENERGIZER HIT?
        JEQ   NOBLCK              NO,  DON'T CHANGE COLOR
        DEC   ENGCNT              DECREMENT ENERGIZER COUNTER-TIMER
        JNE   CHKCOL              TIME NOT UP, SO STAY ENERGIZED
        MOVB  H00,FLAGZZ          TIME UP, SO RESET ENERGIZER FLAG
        LI    VDPADD,>38D         RESTORE ENERGIZERS BACK TO GREEN
        LI    WLOC,COLGRN
        LI    WCOUNT,1
MORGRN  BL    @WRITE
        INCT  VDPADD
        CI    VDPADD,>397
        JNE   MORGRN
        BL    @MAZCOL             RESTORE ORIGINAL MAZE COLOR
        BL    @BNKNUM             CLEAR CAPTURE POINTS
        MOV   STALPC,WAITPC       RESET MMAN INIT SPEED
        MOV   STALPC,SAVWPC       RESET MMAN INIT SPEED
        MOVB  H08,YXLOC0+3        ZAP BACK TO ORIGINAL COLOR-MONSTER0
        MOVB  H04,YXLOC1+3        ZAP BACK TO ORIGINAL COLOR-MONSTER1
        MOVB  H0D,YXLOC2+3        ZAP BACK TO ORIGINAL COLOR-MONSTER2
        MOVB  H0A,YXLOC3+3        ZAP BACK TO ORIGINAL COLOR-MONSTER3
NOBLCK  MOV   YXLOC0(MON),YXPOS   SAVE YPT,XPT OF CURRENT MONSTER
        AI    YXPOS,>0100       AJUST SPRITE POSITION TO CENTER
        ANDI  YXPOS,>0707       CLEAR WORD TO CHECK CHAR BOUNDARY
        JNE   JUMP06            "AND" NOT=0, KEEP MOVING(NO CHAR BOUNDARY)
        MOV   YXLOC0(MON),YXPOS   RESTORE YPT,XPT PREVIOUSLY SAVED
        AI    YXPOS,>0100       AJUST SPRITE POSTION TO CENTER
        BL    @FPTTRN           GO GET THE CHARACTER BOUNDARY
        CLR   YMONSV                  CLEAR MONSTER SAVE LOCATION
        CLR   XMONSV                  CLEAR MONSTER SAVE LOCATION
        CLR   YPTSAV                  CLEAR MMAN SAVE LOCATION
        CLR   XPTSAV                  CLEAR MMAN SAVE LOCATION
        MOVB  H00,FLAGPC              RESET PASS FLAG TO NO PASSES
        MOVB  YXLOCP,YPTSAV+1         SAVE MMAN LOCATION
```

```
        MOVB  YXLOCO(MON),YMONSV+1    SAVE MONSTER YPT LOCATION
        S     YMONSV,YPTSAV           GET DIFFERENCE OF YPT'S
        MOV   YPTSAV,TEMPR4           MOVE VALUE TO HIGHER BYTE
        ABS   TEMPR4                  GET POSITIVE DIFFERENCE
        C     TEMPR4,DELTAO(MON)      WITHIN MONSTER CHARACTERS?
        JH    JUMP04                  NO,  GET RANDOM #
        MOV   TEMPR4,R10              SAVE YPT DIFF ABS FOR LATER
        MOVB  YXLOCP+1,XPTSAV+1       SAVE MMAN LOCATION
        MOVB  YXLOCO(MON)+1,XMONSV+1  SAVE MONSTER LOCATION
        S     XMONSV,XPTSAV           GET DIFFERENCE OF XPT'S
        MOV   XPTSAV,TEMPR4           MOVE VALUE TO HIGHER BYTE
        ABS   TEMPR4                  GET POSITIVE DIFFERENCE
        C     TEMPR4,DELTAO(MON)      WITHIN MONSTER CHARACTERS?
        JH    JUMP04                  NO,  GET RANDOM #
        C     R10,H0004               DIFF IN YPT'S LESS THAN 4 PIXELS?
        JH    GOPASS                  NO, SO CONTINUE
        C     R4,H0004                DIFF IN XPT'S LESS THAN 4 PIXELS?
        JH    GOPASS                  NO, SO CONTINUE
        BL    @COINCI                 YPT'S AND XPT'S LESS THAN 4 PIXELS
        B     @DONEMV                 MONSTER CAPTURED, GO ON TO NEXT
JUMP04  B     @GETRAN                 CONTINUE
JUMP06  B     @CHKCOI                 CONTINUE
CHKCOL  C     ENGCNT,FLHCNT           TIME TO FLASH WARNING YET?
        JH    NOWARN                  NO, CONTINUE
        S     H0032,FLHCNT            YES, DECREMENT FLASH COUNTER
        C     FLHCNT,H0064            FLASH COUNT ALMOST UP?
        JNE   NORED                   NO,  CONTINUE
*********************************
*  FLASH MAZE RED FOR WARNING   *
*********************************
        LI    VDPADD,>380             YES, POINT TO COLOR TABLE AREA
        LI    WLOC,COLRED             GET MRED FOR MAZE
        LI    WCOUNT,5                WRITE 5 SETS IN COLOR TABLE
        BL    @WRITE                  GO WRITE IT
        LI    VDPADD,>39C             COLOR SET FOR NUMBERS
        LI    WCOUNT,1                WRITE ONE SET
        BL    @WRITE                  GO WRITE IT
NORED   CB    YXLOCO(MON)+3,H01       IS COLOR OF MONSTER BLACK?
        JEQ   OLDCOL                  YES, GO CHANGE BACK TO OLD COLOR
        MOVB  H01,YXLOCO+3            NO,  CHANGE TO BLACK THEN
        MOVB  H01,YXLOC1+3            NO,  CHANGE TO BLACK THEN
        MOVB  H01,YXLOC2+3            NO,  CHANGE TO BLACK THEN
        MOVB  H01,YXLOC3+3            NO,  CHANGE TO BLACK THEN
        JMP   NOWARN                  CONTINUE
OLDCOL  MOVB  H08,YXLOCO+3            RESTORE ORIGINAL OLD COLOR
        MOVB  H04,YXLOC1+3            RESTORE ORIGINAL OLD COLOR
        MOVB  H0D,YXLOC2+3            RESTORE ORIGINAL OLD COLOR
        MOVB  H0A,YXLOC3+3            RESTORE ORIGINAL OLD COLOR
NOWARN  C     YXLOCO(MON),YXPITO(MON)     MONSTER IN PIT?
        JEQ   DONEMV                  YES, GO ON
        B     @NOBLCK                 NO,  SO GO THROUGH CHECKING
CHKCOI  BL    @COIN                   CHECK FOR COINCIDENCE
        CB    FLAGZZ,H00              MMAN ENERGIZED?
        JEQ   NOCHNG                  NO,  SO INCREMENT(NO COIN DETECTED)
        C     YXLOCO(MON),YXPITO(MON)     YES, CHECK IF COIN OCCURRED
        JEQ   DONEMV                  COIN, SO DON'T INCREMENT
        JMP   NOCHNG                  NO COIN, SO INCREMENT
GOPASS  BL    @RANDNO                 GET RANDOM NUMBER
        ANDI  RAND,>0F00              CLEAR TO 2ND NYBBLE
        SRL   RAND,8                  NOW 0 - 15 PROBABILITY
        C     RAND,PROB00(MON)        CHECK AGAINST MONSTER'S PROB#
        JH    GETRAN                  RAND# HIGHER THAN MONSTER'S PROB#
```

```
        MOVB  H00,FLAGPC              SET PASS FLAG TO NO PASSES
MOREBI  CB    H02,FLAGPC              PASS FLAG SET TO 2 PASSES YET?
        JEQ   GETRAN                  YES, ALREADY 2 PASSES, GET RANDOM#
        AB    H01,FLAGPC              SET PASS FLAG TO 1 OR 2 PASSES
        CB    H01,FLAGPC              FIRST PASS?
        JEQ   CHKYPT                  YES, GO UP OR DOWN IF POSSIBLE
CHKXPT  C     XPTSAV,H0000            MONSTER + OR - XPT(LEFT OR RIGHT)?
        JGT   GORIGT                  MONSTER LEFT, SO GO RIGHT
        CB    FLAGZZ,H01              HIT ENERGIZER?
        JEQ   LIRIGT                  YES, GO MOVE OPPOSITE DIRECTION
LILEFT  LI    RAND,2                  MONSTER RIGHT, SO GO LEFT
        JMP   SAMERD                  GO MOVE MONSTER
GORIGT  CB    FLAGZZ,H01              HIT ENERGIZER?
        JEQ   LILEFT                  YES, GO MOVE OPPOSITE DIRECTION
LIRIGT  LI    RAND,3
        JMP   SAMERD                  GO MOVE MONSTER
CHKYPT  C     YPTSAV,H0000            MONSTER + OR - YPT(ABOVE OR BELOW)?
        JEQ   CHKXPT                  IF NO YPT DELTA, THEN CHECK XPT DELTA
        JGT   GODOWN                  MONSTER ABOVE
        CB    FLAGZZ,H01              HIT ENERGIZER?
        JEQ   LIDOWN                  YES, GO MOVE OPPOSITE DIRECTION
LIUP    LI    RAND,0                  MONSTER BELOW, SO GO UP
        JMP   SAMERD                  GO MOVE MONSTER
GODOWN  CB    FLAGZZ,H01              HIT ENERGIZER?
        JEQ   LIUP                    YES, GO MOVE OPPOSITE DIRECTION
LIDOWN  LI    RAND,1                  GO DOWN
        JMP   SAMERD                  GO MOVE MONSTER
GETRAN  MOVB  H02,FLAGPC              SET PASS FLAG TO ALL DONE
        BL    @RANDNO           GET RANDOM NUMBER
        MOV   RAND,TEMR14       SET RANDOM NUMBER(16 BIT, 1 WORD)
        ANDI  RAND,>0C00        CLEAR TO MIDDLE TWO BITS ONLY
        SRL   RAND,10           NOW 0 - 3 RANDOM NUMBER IN R9
SAMERD  BL    @TSTBIT           GO TEST BIT
        BL    @TSTREV           GO TEST FOR REVERSE POSSIBILITY
        CB    H00,FLAG          WAS FLAG SET(BIT ON- 1000,0100,0010,0001)?
        JEQ   MOREBI            NO,  SO TEST MORE BITS(U,D,L,R)
        MOV   RAND,SAVRMO(MON)  SAVE RANDOM FOR LATER COMPARE IN TSTREV SUB.
        SLA   RAND,1            X2 FOR POINTER INTO TABLE1
        MOV   TABLE1(RAND),INCRO(MON)     SAVE MONSTER INCREMENT(YPT,XPT)
NOCHNG  AB    INCRO(MON),YXLOCO(MON)      INCREMENT MONSTER W/ CURRENT YPT
        AB    INCRO(MON)+1,YXLOCO(MON)+1  INCREMENT MONSTER W/ CURRENT XPT
DONEMV  MOV   RA+12,R11         GET RETURN ADDRESS PREVIOUSLY SAVED
        B     *R11              RETURN TO CALLER

*****************************************************
*  THIS ROUTINE CHANGES MAZE COLOR BACK TO ORIGINAL  *
*****************************************************
MAZCOL  MOV   R11,RA+0
        CLR   R10
        LI    VDPADD,>39D       COLOR SET PERMANENTLY ORIGINAL MAZE COLOR
        LI    RLOC,REGALB
        LI    RCOUNT,1
        BL    @READ
        LI    VDPADD,>380
        LI    WLOC,REGALB
GETCOL  BL    @WRITE            CHANGE 5 COLOR SETS OF MAZE BACK
        INC   VDPADD
        CI    VDPADD,>385
        JNE   GETCOL
        LI    VDPADD,>39C       CHANGE NUMBERS BACK, TOO
        BL    @WRITE
        MOV   RA+0,R11
```

```
        B     *R11

******************
*  DELAY ROUTINE  *
******************
PAUSOA  MOVB  H00,TIMER         DELAY
DEL1    CB    TIMER,H0A         DELAY
        JNE   DEL1              DELAY
        B     *R11              RETURN TO CALLER

*********************************
*  BLANK OUT 100,200,400,800    *
*  CAPTURE POINT DISPLAYS       *
*********************************
BNKNUM  MOV   R11,RA+2          SAVE R11
        CLR   R4                CLEAR REG4
        LI    WCOUNT,3          WRITE 3 BYTES
        LI    WLOC,WALL03       ESTABLISH WRITE ADDRESS
CLRPTS  MOV   TABLE7(R4),VDPADD  VDP SCREEN BLOCK 1, 2, 3, 4
        BL    @WRITE            CLEAR POINTS AND PUT BACK BLOCK
        INCT  R4                NEXT BLOCK
        CI    R4,8              ALL DONE W/ THE FOUR(0,1,2,3)?
        JNE   CLRPTS            NOP, KEEP CLEARING POINTS
        MOV   RA+2,R11          RESTORE R11
        B     *R11              RETURN

*************************************
*  THIS ROUTINE CHECKS COINCIDENCE  *
*  BETWEEN THE MONSTERS AND MMAN    *
*************************************
COIN    MOV   R11,RA+8          SAVE RETURN ADDRESS
        CLR   TEMPRA            CLEAR REG10
        CLR   TEMPR4            CLEAR REG4
        MOVB  YXLOCP,REG4LB     SAVE MMAN LOCATION
        MOVB  YXLOCO(MON),REGALB  SAVE MONSTER YPT LOCATION
        C     R4,R10            YPT'S SAME?
        JEQ   TRYXPT            YES, SO GO CHECK XPT'S
        MOVB  YXLOCP+1,REG4LB   SAVE MMAN LOCATION
        MOVB  YXLOCO(MON)+1,REGALB  SAVE MONSTER LOCATION
        C     R4,R10            XPT'S SAME?
        JNE   GOBACK            NO, SO ON A CORNER TURN
        MOVB  YXLOCP,REG4LB     SAVE MMAN LOCATION
        MOVB  YXLOCO(MON),REGALB  SAVE MONSTER LOCATION
        S     TEMPRA,TEMPR4     GET DIFFERENCE OF YPT'S
        ABS   TEMPR4            GET POSITIVE DIFFERENCE
        C     TEMPR4,H0004      DIFF IN YPT'S LESS THAN 4 PIXELS?
        JH    GOBACK            NO, CONTINUE
        CLR   TEMPRA            CLEAR REG10
        CLR   TEMPR4            CLEAR REG4
TRYXPT  MOVB  YXLOCP+1,REG4LB   SAVE MMAN LOCATION
        MOVB  YXLOCO(MON)+1,REGALB  SAVE MONSTER LOCATION
        S     TEMPRA,TEMPR4     GET DIFFERENCE OF XPT'S
        ABS   TEMPR4            GET POSITIVE DIFFERENCE
        C     TEMPR4,H0004      DIFF IN YPT'S LESS THAN 4 PIXELS?
        JH    GOBACK            NO, SO CONTINUE
        BL    @COINCI           YPT'S AND XPT'S LESS THAN 4 PIXELS
GOBACK  MOV   RA+8,R11          RESTORE RETURN ADDRESS
        B     *R11              RETURN, NO COINCIDENCE

*********************************
*  1% PROPABILITY THE MONSTERS  *
*  CAN REVERSE THEIR DIRECTION  *
```

```
*********************************
TSTREV ANDI  TEMR14,>03F0          CLEAR TO ONLY TWO DIGITS
       SRL   TEMR14,4              PUT THEM IN LOWER PART(0 - 63 RANDOM)
       CI    RAND,3
       JEQ   CHK02
       CI    RAND,2
       JEQ   CHK03
       CI    RAND,1
       JEQ   CHK00
       CI    RAND,0
       JEQ   CHK01
       JMP   DONERV
CHK02  C     SAVRMO(MON),H0002
       JEQ   NOREVS
       JMP   DONERV
CHK03  C     SAVRMO(MON),H0003
       JEQ   NOREVS
       JMP   DONERV
CHK00  C     SAVRMO(MON),H0000
       JEQ   NOREVS
       JMP   DONERV
CHK01  C     SAVRMO(MON),H0001
       JEQ   NOREVS
       JMP   DONERV
NOREVS CI    TEMR14,2
       JL    DONERV
       MOVB  H00,FLAG              NO REVERSING THIS TIME
DONERV B     *R11

************************************
*  ROUTINE THAT CHECKS FOR U,D,L,R *
*  ARROW KEYS TO MOVE THE MMAN     *
************************************
MOVMAN MOV   R11,RA+10
       MOV   YXLOCP,YXPOS
       AI    YXPOS,>0100           AJUST SPRITE POSITION TO CENTER
       ANDI  YXPOS,>0707
       JNE   JUMP02
       MOV   YXLOCP,YXPOS
       AI    YXPOS,>0100           AJUST SPRITE POSTION TO CENTER
       BL    @FPTTRN
       CI    PTRNNO,>AF            ENERGIZER RAN OVER?
       JH    NOENGZ                NO, GO CHECK FOR DOT THEN
       CB    FLAGZZ,H00            MMAN ALREADY ENERGIZED?
       JNE   YESON1                YES, DON'T RE-INIT CAPTURE POINTS
       MOVB  H01,CAPPTS            NO,  INIT CARTURE POINTS
       S     H0070,WAITPC          SPEED UP ENERGIZED MMAN BY 112 COUNTS
YESON1 MOVB  H01,FLAGZZ            GO SET ENERGIZED FLAG
       MOV   WAITPC,SAVWPC         SAVE MMAN SPEED-UP
       MOV   ENGSAV,ENGCNT         INIT ENERGIZED TIME-OUT COUNT
       MOV   H01F4,FLHCNT          INIT MONSTERS' WARNING FLASH 500 COUNTS
       MOVB  H01,YXLOC0+3          CHANGE MONSTER0 TO BLACK
       MOVB  H01,YXLOC1+3          CHANGE MONSTER1 TO BLACK
       MOVB  H01,YXLOC2+3          CHANGE MONSTER2 TO BLACK
       MOVB  H01,YXLOC3+3          CHANGE MONSTER3 TO BLACK
       LI    PTRNNO,>00CC          CHANGE TO A BLANK
       LI    WLOC,PTNOLB           LOAD WRITE BUFFER W/ NEW PATTERN
       BL    @WRITE                GO WRITE TO VDP(SCREEN)
       BL    @MAZCOL               RESTORE ORIGINAL MAZE COLOR

************************
*  TURN ENERGIZERS RED  *
```

```
*************************
        LI    VDPADD,>3AD
        LI    WLOC,COLRED
        LI    WCOUNT,1
MORRED  BL    @WRITE
        INCT  VDPADD
        CI    VDPADD,>397
        JNE   MORRED
        LI    R10,SNDENZ        GET ENERGIZER
        BL    @SOUND            EATEN SOUND
        LI    R10,>0070         INCREASE SCORE BY 70
        A     H0046,SCRSAV      ADD 70 TO SCORE ACCUM
        JMP   WRTSCR            GO WRITE SCORE TO SCREEN
JUMP02  CB    YXPOS,H00
        JNE   COMBYT
        SLA   YXPOS,8
COMBYT  CB    YXPOS,H03
        JNE   CHK06
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>F000      CHANGE SPRITE CHAR(OPEN MOUTH)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
        B     @DONECH           CONTINUE
CHK06   CB    YXPOS,H06
        JNE   JUMP05
        MOVB  YXLOCP+2,TEMPR4
        ANDI  TEMPR4,>F000      HALF OPEN MOUTH
        ORI   TEMPR4,>0100      HALF OPEN MOUTH
        MOVB  TEMPR4,YXLOCP+2
JUMP05  B     @DONECH
NOENGZ  CI    PTRNNO,>00D0      RAN OVER A DOT?
        JL    NODOT1            NO,  KEEP GOING
        AI    PTRNNO,-16        YES, CHANGE TO A BLANK
        LI    WLOC,PTNOLB       LOAD WRITE BUFFER WITH NEW PATTERN
        BL    @WRITE            GO WRITE TO VDP(SCREEN)
        MOVB  @>83CE,R10        ANY SOUND GOING ON?
        JEQ   SNDIT1            >83CE = 0 , SO NO OTHER SOUNDS NOW
        MOV   @>83CC,R10        GET SOUND LIST POINTER VDP LOCATION
        CI    R10,>3025         ENZ OR MON EAT SOUND IN PROGRESS?
        JL    AROUND            YES, SO DON'T EXECUTE MUNCH SOUND
SNDIT1  LI    R10,SNDMUN        GET THE
        BL    @SOUND            MUNCH SOUND
AROUND  LI    R10,>0010         INCREMENT SCORE BY 10'S
        A     H000A,SCRSAV      ADD 10 TO SCORE ACCUM
WRTSCR  BL    @SCORE            GO ADD TO THE SCORE
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>F000      CHANGE SPRITE CHAR(CLOSED MOUTH)
        ORI   TEMPR4,>0200      CHANGE SPRITE CHAR(CLOSED MOUTH)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
        BL    @CHKPTS           CHECK 10,000 POINTS FOR EXTRA MMAN
        BL    @WRITE            GO WRITE EXTRA MMAN
        INC   DOTCNT            ONE MORE DOT EATEN
        C     DOTCNT,H00F8      ALL DOTS GONE YET?(248 DOTS)
        JNE   EATON             NO,  CONTINUE
        MOVB  H01,WINFLG        SET GAME WIN FLAG
        MOVB  H00,FLAGZZ        SET TO NO HIT ENERGIZER
        BL    @COINCI           STOP GAME AND GO BACK TO GPL
NODOT1  MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>F000      CHANGE SPRITE CHAR(OPEN MOUTH)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
EATON   CLR   RLOC              CLEAR VDP READ LOCATION
        BL    @CHKPTS           GO CHECK FOR 10,000 POINTS EXTRA MMAN
        BL    @WRITE            GO WRITE IT ON SCREEN
```

```
        CLR   RLOC              CLEAR READ  LOCATION
        CLR   WLOC              CLEAR WRITE LOCATION
        CB    DEMOFG,H00        IN DEMO MODE?
        JEQ   NODEMO            NOP
        BL    @SCANKY           YEP, SCAN KEYBOARD TO GET OUT OF DEMO
        MOVB  H00,KEYBRD        KEYBOARD ZERO
        MOVB  STATUS,STATUS     CHECK FOR ANY KEY
        JEQ   NOKEYS            NO KEY HIT FO ESCAPE YET
        MOVB  H01,ESCAPE        KEY HIT, SET ESCAPE FLAG
        MOVB  H00,DEMOFG        RE-INIT TO NO DEMO MODE
        BL    @BNKNUM           CLEAR CAPTURE POINTS
        BL    @MAZCOL           RESTORE ORIGINAL MAZE COLOR
        LI    VDPADD,>38D       RESTORE ENERGIZERS BACK TO GREEN
        LI    WLOC,COLGRN                    "
        LI    WCOUNT,1                       "
MOREGN  BL    @WRITE                         "
        INCT  VDPADD                         "
        CI    VDPADD,>397                    "
        JNE   MOREGN                         "
        B     @DEMOUT           GO BACK AND START THE GAME NOW
NOKEYS  BL    @RANDNO           GET RANDOM NUMBER
        ANDI  TEMR14,>03F0      MASK OUT 7 BITS
        SRL   TEMR14,4          GET JUST 0 - 63)
        C     SAVINC,H0001      U,D,L,R(0,1,2,3)? MMAN
        JLE   GOLR
        CI    TEMR14,25         PROB - 50%
        JLE   TAKE01
        MOVB  H05,KEY           GO UP
        JMP   FORCEM
TAKE01  MOVB  H00,KEY           GO DOWN
        JMP   FORCEM
GOLR    CI    TEMR14,30         PROB - 50%
        JHE   TAKE03
        MOVB  H02,KEY           GO LEFT
        JMP   FORCEM
TAKE03  MOVB  H03,KEY           GO RIGHT
FORCEM  BL    @RANDNO
        JMP   FORCEV
NODEMO  MOVB  H01,KEYBRD        SET UP KEYBOARD ONE SCAN
        BL    @SCANKY           SCAN KEYBOARD
        CLR   TEMPR4            ZERO OUT REG4
        CB    KEY,H05           IS KEY RETURNED 0 - 5 ?
        JH    CHKH01            YES, SO GO CHECK JOYSTICK ONE
FORCEV  CLR   R4                CLEAR REG4
        MOVB  KEY,REG4LB        KEY IS 0 - 5
        SLA   TEMPR4,1          ADJUST TABLE POINTER(WORD)
        MOV   TABLE5(TEMPR4),TEMPR4      POINT TO  BRANCH ADDRESS
        BL    *TEMPR4           GO MOVE MMAN
CHKH01  BL    @CHKJOY           GO CHECK JOYSTICK
        MOVB  H02,KEYBRD        TRY KEYBOARD TWO
        BL    @SCANKY           SCAN THE KEYBOARD
        BL    @CHKJOY           GO CHECK JOYSTICK
*****************************************
*  AT THIS POINT, CHECK FOR PAUSE KEY   *
*****************************************
        CB    KEY,H0B           "P" KEY HIT TO STOP GAME?
        JNE   NOTP01            NO,  KEEP ON GOING
KEY0    BL    @SCANKY
        MOVB  STATUS,STATUS
        JNE   KEY0
        CLR   KEYBRD
        LI    VDPADD,25
```

```
        LI    WCOUNT,5
        CLR   R10
MOREMG  LI    WLOC,MSGPAU
        BL    @WRITE
        MOVB  H00,TIMER
        MOVB  H00,CLRSCN
DELYP1  BL    @SCANKY
        MOVB  STATUS,R10
        JNE   BLKMSG
        CB    TIMER,H30
        JNE   DELYP1
        LI    WLOC,MSGBLK
        BL    @WRITE
        MOVB  H00,TIMER
        CLR   R10
DELYP2  BL    @SCANKY
        MOVB  STATUS,R10
        JNE   NOTP01
        CB    TIMER,H10
        JNE   DELYP2
        JMP   MOREMG
BLKMSG  LI    WLOC,MSGBLK
        BL    @WRITE
NOTP01  JMP   CHKSTP            GO CHECK IF MMAN MUST STOP
UPKEY   LI    RAND,0            INIT TO UP BIT
        BL    @TSTBIT
        CB    H00,FLAG
        JEQ   CHKSTP
        MOV   HFF00,INCMAN      MOVE UP NOW
        MOV   H0000,SAVINC
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>0F00      CHANGE SPRITE CHAR(UP)
        ORI   TEMPR4,>A000      CHANGE SPRITE CHAR(UP)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
        JMP   DONECH
DNKEY   LI    RAND,1            INIT TO DOWN BIT
        BL    @TSTBIT
        CB    H00,FLAG
        JEQ   CHKSTP
        MOV   H0100,INCMAN      MOVE DOWN
        MOV   H0001,SAVINC
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>0F00      CHANGE SPRITE CHAR(DOWN)
        ORI   TEMPR4,>B000      CHANGE SPRITE CHAR(DOWN)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
        JMP   DONECH
LFKEY   LI    RAND,2            INIT TO LEFT BIT
        BL    @TSTBIT
        CB    H00,FLAG
        JEQ   CHKSTP
        MOV   H00FF,INCMAN      MOVE LEFT
        MOV   H0002,SAVINC
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>0F00      CHANGE SPRITE CHAR(LEFT)
        ORI   TEMPR4,>9000      CHANGE SPRITE CHAR(LEFT)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
        JMP   DONECH
RTKEY   LI    RAND,3
        BL    @TSTBIT
        CB    H00,FLAG
        JEQ   CHKSTP
        MOV   H0001,INCMAN      MOVE RIGHT
```

```
        MOV   H0003,SAVINC
        MOVB  YXLOCP+2,TEMPR4   MOVE IN SPRITE'S CHAR
        ANDI  TEMPR4,>0F00      CHANGE SPRITE CHAR(RIGHT)
        ORI   TEMPR4,>8000      CHANGE SPRITE CHAR(RIGHT)
        MOVB  TEMPR4,YXLOCP+2   MOVE BACK SPRITE'S NEW CHAR
DONECH  AB    INCMAN,YXLOCP
        AB    INCMAN+1,YXLOCP+1
        JMP   ENDCHK
CHKSTP  MOV   SAVINC,RAND
        BL    @TSTBIT
        CB    H00,FLAG
        JNE   DONECH
ENDCHK  MOV   RA+10,R11
        B     *R11
*****************************************************
*  ROUTINE THAT CHECKS FOR EXTRA MMAN AT 10,000 POINTS  *
*****************************************************
CHKPTS  CLR   WLOC              CLEAR VDP WRITE LOCATION
        C     SCRSAV,H2710      HIT 10000 POINTS YET?
        JL    NOEXTR            NOT YET, CONTINUE
        S     H2710,SCRSAV      YES, ANOTHER 10,000 REACHED AGAIN
        AB    H01,COUNT1        ADD AN EXTRA MMAN
        CB    COUNT1,H06        SIX MMEN NOW?
        JHE   NOEXTR            DON'T SHOW ANY MORE EXTRA MMEN
        CLR   TEMPR4            CLEAR REG4
        MOVB  COUNT1,REG4LB     MOVE TO LOWER BYTE OF REG4
        SB    H02,REG4LB        ADJUST TABLE-SCREEN POINTER
        LI    WCOUNT,1          WRITE ONE BYTE TO VDP
        MOVB  TABLE4(TEMPR4),VDPADD     SET BUFFER SCREEN WRITE LOCATION
        SRA   VDPADD,8                  AND PUT IN LSB OF REG0
        LI    TEMPRA,>0074      LOAD EXTRA MMAN CHARACTER
        LI    WLOC,REGALB       LOAD LOWER BYTE INTO WRITE LOCATION
        B     *R11              RETURN
NOEXTR  A     H0004,R11         GO FOUR WORDS PAST THE WRITE
        B     *R11              RETURN
*************************************
*  ROUTINE THAT CHECKS FOR JOYSTICK  *
*************************************
CHKJOY  MOV   JOYY,TEMPR4       GET JOYY,JOYX
        JEQ   CHKH02            BOTH = ZERO SO NO JOYSTICK
        MOVB  JOYY,REG4HB       GET JOYY
        JEQ   GOODJO            ZERO, SO GOOD DIRECTION
        MOVB  JOYX,REG4LB       GET JOYX
        JEQ   GOODJO            ZERO, SO GOOD DIRECTION
        JMP   CHKH02            NEITHER ZERO SO GOT DIAGONAL
GOODJO  AB    H04,REG4HB        GET RID OF NEG NUMBERS
        AB    H04,REG4LB        GET RID OF NEG NUMBERS
        SZCB  H04,REG4LB        CLEAR OUT ONE COLUMN OF BITS
        AB    REG4HB,REG4LB     ADD UPPER AND LOWER BYTES
        SZCB  HFF,REG4HB        MAKE UPPER BYTE BITS VALUE OF REG4
        SRL   TEMPR4,1          MOVE TO LOW END OF WORD(0,2,4,6 VALUES)
        MOV   TABLE6(TEMPR4),TEMPR4   POINT TO BRANCH ADDRESS
        B     *TEMPR4           GO MOVE MMAN
CHKH02  B     *R11              NO HIT, SO RETURN
***************************
*  SCAN KEYBOARD ROUTINE  *
***************************
SCANKY  MOVB  H00,CLRSCN     ZERO SCREEN TIME-OUT COUNTER
        LIMI  0
        LWPI  GPLWS
        BL    @SCAN          SCAN THE KEYBOARD
        LWPI  MYWS
```

```
        LIMI 2
        B    *R11

*****************************************************************
*  GIVEN ANY YXPOS OF A SPRITE, RETURN THE PATTERN NUMBER       *
*  UNDER THE SPRITE AND RETURN THE PIXEL WITHIN THE             *
*  PATTERN (XREM,YREM)                                          *
*****************************************************************
FPTTRN MOV  R11,RA+14
        MOV  YXPOS,Y       GET Y AND X FROM YXPOS
        SRL  Y,8
        MOV  YXPOS,X
        ANDI X,>00FF
        MOV  Y,CARYHB      YREM
        MOV  X,XREM
        SRL  Y,3           DIVIDE BY 8
        SRL  X,3
        SLA  Y,3           MULTIPLY QUOTIENTS BY 8
        SLA  X,3
        S    Y,CARYHB      YREM,CALCULATE REMAINDER
        S    X,XREM
        SLA  Y,2           (Y*8)*4
        SRL  X,3           RESTORE X TO QUOTIENT
        A    X,Y
        CLR  PTRNNO
        LI   RLOC,PTNOLB
        LI   RCOUNT,1
        BL   @READ         Y IS THE VDP ADDRESS
        MOV  RA+14,R11
        B    *R11

*******************************************
*  THIS ROUTINE GENERATES A 16 BIT       *
*  RANDOM NUMBER AND PLACES IT IN RAND   *
*******************************************
RANDNO LI   TEMR14,28645
        MPY  RAND16,TEMR14
        AI   RAND,31417
        MOV  RAND,RAND16
        B    *R11

***************
*  TEST BIT   *
***************
TSTBIT MOVB PTNOLB,FLAG
        MOVB BITMSK(RAND),SCRTCH
        INV  SCRTCH
        SZCB SCRTCH,FLAG
        B    *R11

***************************
*  ADD TO SCORE ROUTINE   *
***************************
SCORE  MOV  R11,RA+6
        CLR  DIGTHB
        CLR  CARYHB
        LI   VDPADD,>36
        LI   RLOC,DIGTLB
        LI   WLOC,DIGTLB
        LI   RCOUNT,1
RDIGIT BL   @READ
        MOV  R10,SAVR10
```

```
        ANDI R10,>000F          GET RIGHT MOST DIGIT
        A    R10,DIGTHB         ADD DIGITS
        A    CARYHB,DIGTHB      ADD CARRY
        CLR  CARYHB             CLEAR CARRY
        CB   DIGTLB,H39         DIGIT OVER NINE?
        JLE  WDIGIT
        MOVB H01,CARYLB         SET CARRY
        SB   H0A,DIGTLB
WDIGIT  BL   @WRITE
        MOV  SAVR10,R10
        SRL  R10,4              NEXT PAIR OF DIGITS
        DEC  VDPADD
        CI   VDPADD,>31
        JHE  RDIGIT
        MOV  RA+6,R11
        B    *R11

************************
*  CALL SOUND ROUTINE  *
************************
SOUND   MOV  R10,@>83CC         SOUND LIST POINTER(HIGH VDP ADDRESS)
        SOCB H01,@>83FD         SOUND TIMER MSByte(GPL REG14)
        MOVB H01,@>83CE         NUMBER OF SOUND BYTES
        LIMI 2                  ENABLE INTERRUPTS
        B    *R11               RETURN

******************
* READ FROM VDP  *
******************
READ    LIMI 0                  DISABLE INTERRUPTS
        MOVB VADDLB,@>8C02      SET UP LOWER BYTE OF ADDRESS
        MOV  RCOUNT,COUNT
        MOVB VDPADD,@>8C02      SET UP UPPER BYTE OF ADDRESS
        MOV  RLOC,TEMP
R00010  MOVB VDPRD,*TEMP+       READ VDP BYTE INTO BUFFER
        DEC  COUNT              DECREMENT COUNTER
        JGT  R00010             AND LOOP IF NEEDED
        LIMI 2                  ENABLE INTERRUPTS
        RT
******************
* WRITE TO VDP   *
******************
WRITE   LIMI 0                  DISABLE INTERRUPTS
        MOVB VADDLB,@>8C02      SET UP LOWER BYTE OF ADDRESS
        MOV  WCOUNT,COUNT
        MOV  VDPADD,TEMP
        ORI  TEMP,>4000         SET VDP WRITE FLAG
        MOVB TEMP,@>8C02        SET UP UPPER BYTE OF ADDRESS
        MOV  WLOC,TEMP
W00010  MOVB *TEMP+,VDPWD       WRITE BYTE INTO VDP
        DEC  COUNT              DECREMENT COUNTER
        JGT  W00010
        LIMI 2                  ENABLE INTERRUPTS
        RT
SLAST   END
```