

```

*****
*
*      WORD RADAR
*      BY      : JOHN PHILLIPS
*      DATE    : 4/25/83
*      SYS1 FILE NAME: PCD1.JMPO89.SRC.WRROM
*
*****

```

```

      IDT 'WRDRDR'
      TITL 'WRDRDR'
      RORG >6000
      EVEN
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
*

```

```

      RORG >6010
      DATA @WRDRDR          START OF MAINLINE          XML >70

```

```

*****
* MEMORY MAPPED EQUATES
*****

```

```

CPURAM EQU >8300
SCAN   EQU >000E          SCAN ADDRESS
PDT    EQU >0000          PATTERN DESCRIPTOR TABLE
SIT    EQU >1800          SCREEN IMAGE TABLE
SAL    EQU >1D00          SPRITE ATTRIDUTE TABLE
SDL    EQU >1C00          SPRITE DESCRIPTOR LIST
SVT    EQU >1F00          SPRITE VELOCITY TABLE (128 BYTES)
RSMOT  EQU SVT           USED BY MY AUTO MOTION
QSAML  EQU RSMOT-SAL     USED BY MY AUTO MOTION
FREVDP EQU >1F80         FREE VDP MEMORY(128 BYTES)
CT     EQU >2000         COLOR TABLE
HIVDP  EQU >3800         HIGH VDP (FREE) (2048 BYTES)
SNDPNT EQU >83CC         VDPRAM SOUND TABLE POINTER
STRTSD EQU >83CE         FLAG TO START SOUND
RAMFLG EQU >83FD         SOUND LIST IN VDPRAM FLAG
VDPRD  EQU >8800         READ WINDOW VDP
VDPWD  EQU >8C00         WRITE WINDOW VDP
VDPWA  EQU >8C02         HARDWARE READ/WRITE WINDOW VDP
GRMWA  EQU >9C02         HARDWARE READ/WRITE WINDOW GROM
GRMRD  EQU >9800         READ WINDOW GROM
GRMRA  EQU >9802         READ ADDRESS WINDOW
GRMWD  EQU >9C00         WRITE WINDOW GROM

```

```

*****
* CPURAM EQUATES
*****
MYWS EQU CPURAM
FASTWS EQU CPURAM

```

MYWS2 EQU CPURAM+>20	FOR LINE DRAWING ALGORITHM
PTSTAT EQU CPURAM+>20	SAME
* >20, 21 RESERVED FOR BLWP @VSBW OR @VSBR	
Y1 EQU CPURAM+>24	STARTING ROW
Y1LB EQU CPURAM+>25	
X1 EQU CPURAM+>26	STARTING COL
X1LB EQU CPURAM+>27	
Y2 EQU CPURAM+>28	ENDING ROW
Y2LB EQU CPURAM+>29	
X2 EQU CPURAM+>2A	ENDING COL
X2LB EQU CPURAM+>2B	
BLKPNT EQU CPURAM+>2C	2 POINTER TO WHICH BLOCK TABLE
RUNTIM EQU CPURAM+>2E	1 RUN TIME IN MINUTES
JOYST EQU CPURAM+>2F	1 JOY/KEYBRD OPTION 0-KEYBRD
NOWINS EQU CPURAM+>30	1 - HOW MANY INCOMPLETE ROUNDS
SAVLST EQU CPURAM+>31	1 - SAVED WORD LIST
LANGNM EQU CPURAM+>43	WHICH LANGUAGE IS BEING USED
*	0 ENG, 1 FR, 2 GER, 3 DUT, 4 IT
LANG EQU CPURAM+>44	2 LANGUAGE POINTER FOR TEXT
CNTFMT EQU CPURAM+>46	2 TEMPORARY WORKSPACE FOR WORD POINTER
MOVTAB EQU CPURAM+>48	2 GROM ADDRESS OF DATA TABLE
TAB1 EQU CPURAM+>4A	2 GROM ADDRESS OF PATTERN
TAB2 EQU CPURAM+>4C	2 VDPRAM ADDRESS OF PATTERN
BYTE EQU CPURAM+>4E	2 COUNT OF # OF BYTES TO MOVE
CB/ADD EQU CPURAM+>50	2 TEXT: CHAR BUFFER ADDRESS
FMT EQU CPURAM+>52	2 TEXT: FORMATTING CODE
FMT1 EQU CPURAM+>54	2 TEXT: FORMATTING CODE
LENGTH EQU CPURAM+>56	2 TEXT: WORD LENGTH
SPEED EQU CPURAM+>58	1 SKILL LEVEL
WRDLST EQU CPURAM+>59	1 PROBLEM RANGE
LIMIT EQU CPURAM+>5A	1 FOR GPL CODE
MAXPRB EQU CPURAM+>5B	1 MAXIMUM # OF PROBLEMS AT LEVEL
FLAG EQU CPURAM+>5C	1 FLAG FOR MENKEY IN GPL
DIFFIC EQU CPURAM+>5D	1 DIFFICULTY LEVEL 1-9
SNDADR EQU CPURAM+>5E	2 SOUND ADDRESS
RADORG EQU CPURAM+>60	RADAR ORIGIN POINTER
RADPNT EQU CPURAM+>62	POINTER INTO RADAR BYTE LIST
RADCNT EQU CPURAM+>64	TOTAL # OF LINES DRAWN
GTRCNT EQU CPURAM+>66	1 - # LINES DRAWN PER QUARTER
WRDNUM EQU CPURAM+>67	1 - WORD NUMBER
WLANG EQU CPURAM+>68	2 - START OF WORD LANGUAGE LIST
ARYPNT EQU CPURAM+>6A	2 - ARRAY POINTER (0-11)
ARRINC EQU CPURAM+>6C	2 - ARROW INCREMENT VALUE
MCHNUM EQU CPURAM+>6E	1 - WORD NUMBER TO MATCH WITH
HITMAX EQU CPURAM+>6F	1 - MAX # OF HITS PER LEVEL
KEYBRD EQU CPURAM+>74	
KEY EQU CPURAM+>75	
JOYY EQU CPURAM+>76	
JOYX EQU CPURAM+>77	
RANDOM EQU CPURAM+>78	
TIMER EQU CPURAM+>79	TIMER BYTE
MOTION EQU CPURAM+>7A	FOR AUTO MOTION
STATUS EQU CPURAM+>7C	STATUS REGISTER
TEMPRT EQU CPURAM+>7E	TEMPORARY RETURN ADDRESS HOLDER
SETHIT EQU CPURAM+>82	1 - HITS SHOWN DURING A ROUND
ACCHIT EQU CPURAM+>83	1 - HITS ACCUMULATED DURING A ROUND
ASETS EQU CPURAM+>84	1 - SETS COMPLETED
SAVEWN EQU CPURAM+>85	1 - SAVED WORD NUMBER
SPEEDV EQU CPURAM+>86	1 - SPEED VARIABLE
TOTMIN EQU CPURAM+>87	1 - TOTAL MONUTES
TOTSEC EQU CPURAM+>88	2 - TOTAL SECONDS ELAPSED
SECLIM EQU CPURAM+>8A	2 - LIMIT OF SECONDS FOR GAME

```

SAVKEY EQU CPURAM+>BC      1 - SAVED KEY/JOY VALUE
WARRAY EQU CPURAM+>AO      WORD ARRAY - 12 BYTES
HITS EQU CPURAM+>AC        2 # OF HITS
MISSES EQU CPURAM+>AE      2 # OF MISSED
LHITS EQU CPURAM+>BO       2 HIGHEST # OF MISSES
LMISS EQU CPURAM+>B2       2 LOWEST # OF HITS
HHITS EQU CPURAM+>B4       2 LOWEST # OF MISSES
HMISS EQU CPURAM+>B6       2 HIGHEST # OF HITS
JOYNUM EQU CPURAM+>B8      1 WHICH JOYSTICK IS BEING SCANNED
*
GPCSAV EQU CPURAM+>BC      GPC SAVE AREA
GPLRA EQU CPURAM+>BE       GPL RETURN ADDRESS
INTWS EQU CPURAM+>CO       INTERRUPT WORKSPACE
SEED EQU CPURAM+>C0        BUILT IN SEED
DISINT EQU CPURAM+>C2      DISABLE SPRITE MOTION AND QUIT FUNCTION
EXINTR EQU CPURAM+>C4      EXTERNAL INTERRUPT ROUTINE WINDOW
SVVDP1 EQU CPURAM+>D4      TO SAVE VDP R1 IF KEY PRESSED
SCNTIM EQU CPURAM+>D6      SCREEN TIME OUT COUNTER
GPLWS EQU CPURAM+>EO       ADDRESS OF GPL WORKSPACE
*****
* OTHER EQUATES *
*****
* REGISTER EQUATES *
*****
TEMPOO EQU R0
VDPADD EQU R0              VDP ADDRESS
RCOUNT EQU R3              NUMBER OF BYTES
WCOUNT EQU R3              NUMBER OF BYTES
RLOC EQU R4                CPU BUFFER
WLOC EQU R4                CPU BUFFER
COUNT EQU R5              TO MOVE RCOUNT OR WCOUNT TO
RXINC EQU R5
M EQU R5                   XINCREMENT
DIGTHB EQU R6
TEMP EQU R6                TO MOVE VDPADD IU
N EQU R6                   YINCREMENT
CARYHB EQU R7
D EQU R7                   XDELTA
E EQU R8                   YDELTA
XPTL EQU R9                CURRENT XPOINT
YPTL EQU R10               CURRENT YPOINT
RVADD EQU R12
SUM EQU R12                SUMMING REGISTER
RAND EQU R15
*****
* WORKSPACE EQUATES *
*****
ROLB EQU MYWS+1
R1LB EQU MYWS+3
R2LB EQU MYWS+5
R3LB EQU MYWS+7
R4LB EQU MYWS+9
R5LB EQU MYWS+11
R6LB EQU MYWS+13
DIGTLB EQU R6LB           FOR SCORE ROUTINE
R7LB EQU MYWS+15
CARYLB EQU R7LB           FOR SCORE ROUTINE
R8LB EQU MYWS+17
R9LB EQU MYWS+19
RALB EQU MYWS+21
RBLB EQU MYWS+23

```

RCLB EQU MYWS+25
RDLB EQU MYWS+27
RELB EQU MYWS+29
RFLB EQU MYWS+31
R1LB2 EQU MYWS+3

FOR VSBW AND VSBR

* SOUND TABLE EQUATES *

RDRSND EQU >3900 RADAR SOUND

MISSND EQU RDRSND+13 MISS SOUND

* BYTE STATEMENTS *

EVEN

RDR1 BYTE >6D, 2

BYTE >71, 2

BYTE >8C, 3

BYTE >91, 3

BYTE >AB, 4

BYTE >B1, 4

BYTE >CB, 4

BYTE >D1, 4

RDR2 BYTE >2B, 4

BYTE >31, 4

BYTE >4B, 4

BYTE >51, 4

BYTE >6C, 3

BYTE >71, 3

BYTE >8D, 2

BYTE >91, 2

RDREND BYTE >FF, 0

EVEN

RADARS EQU \$

RADAR1 EQU \$

ORIGUR BYTE 54, 136 ORIGIN OF UR RADAR

BYTE 23, 136, 24, 139, 25, 142, 26, 144

BYTE 28, 147, 30, 150, 32, 153, 35, 157

BYTE 38, 159, 40, 161, 42, 163, 45, 164

BYTE 48, 166, 51, 167, 54, 167

EVEN

RADAR2 EQU \$

ORIGLR BYTE 72, 136 ORIGIN OF LR RADAR

BYTE 72, 167, 74, 167, 77, 166, 80, 165

BYTE 83, 163, 85, 161, 87, 159, 90, 157

BYTE 93, 153, 95, 150, 97, 147, 99, 144

BYTE 100, 142, 101, 139, 102, 136

EVEN

RADAR3 EQU \$

ORIGLL BYTE 72, 119 ORIGIN OF LL RADAR

BYTE 102, 119, 101, 116, 100, 113, 99, 111

BYTE 97, 108, 95, 105, 93, 102, 90, 97

BYTE 87, 96, 85, 94, 83, 92, 80, 90

BYTE 77, 89, 74, 88, 72, 88

EVEN

RADAR4 EQU \$

ORIGUL BYTE 54, 119 ORIGIN OF UL RADAR

BYTE 54, 88, 51, 88, 48, 89, 45, 90

BYTE 42, 92, 40, 94, 38, 96, 35, 97

BYTE 32, 102, 30, 105, 28, 108, 26, 111

BYTE 25, 113, 24, 116, 23, 119

HEO BYTE >E0

SAVE VDP REGISTER ONE VALUE

HFF BYTE >FF

```

SPEEDS BYTE 60, 56, 52, 48, 44, 40, 36, 32, 28
SNDRDR  BYTE 4, >9F, >BF, >DF, >FA, 1   RADAR MOVING SOUND
        BYTE 2, >E2, >FO, 1
        BYTE 1, >FF, 0
SNDMIS  BYTE 4, >9A, >BF, >DF, >FF, 1
        BYTE 3, >89, >3F, >90, 0
        BYTE 1, >9F, 0

```

```

*
      EVEN
BLOCK1  DATA >0407, >0415, >0D15, >0D07
        DATA >FFFF
BLOCK2  DATA >0605, >0209, >0213, >0617
        DATA >0B17, >0F13, >0F09, >0B05
        DATA >FFFF
BLOCK3  DATA >0605, >0407, >0209, >0213
        DATA >0415, >0617, >0B17, >0D15
        DATA >0F13, >0F09, >0D07, >0B05
        DATA >FFFF

```

```

*****
*      DATA STATEMENTS NEEDED      *
*****

```

```

ZERES  DATA >0000, >0000, >0000, >0000   TO BLANK OUT ANY PATTERNS
RADCOL DATA >D0D0, >D0D0, >D0D0, >D0D0   COLOR FOR RADAR LINES
HOOFF  DATA >00FF                          FOR @HORLIN ROUTINE
H8000  DATA >8000                          FOR @POINT2 ROUTINE
H0000  DATA >0000                          FULL WORD OF ZERES
THRU   DATA >FFFF                          TERMINATOR FOR ROM BASED FORMATTER
LARROW DATA >0004, >02FF, >FF02, >0400   LEFT ARROW
RARROW DATA >0020, >40FF, >FF40, >2000   RIGHT ARROW

```

```

*****
* START OF MAINLINE ROM CODING *
*****

```

```

      EVEN
WRDRDR LIM1 0                               DISABLE INTERRUPTS
        MOV R11, @GPLRA                      SAAE GPL RETURN ADDRESS
        MOVB @GRMRA, @GPCSAV                 SAVE GPL PROGRAM COUNTER
        MOVB @GRMRA, @GPCSAV+1
        DEC @GPCSAV
        LWPI MYWS
        LIM1 2

```

```

*****
* THIS MODULE DOES THE INITS FOR THE START OF THE GAME *
*****

```

```

INITS  BL @CLEAR                            CLEAR THE SCREEN
        BL @WRTVCL                           MOVE RADAR SOUND DATA TO VDP
        DATA >3900, 27, SNDRDR
        BL @GETBLK                            GET DESIRED BLOCK #
        BL @FMTRAD                           FORMAT RADAR CHARACTERS
        BL @MAKSCN                           MAKE THE SCREEN
        BL @SETRAD                           SET ALL THE RADAR POINTERS
        BL @SETARR                           CREATE ARROWS
        BL @CALCWL                           CALCULATE WORD LIST
        BL @CALCSV                           CALCULATE SPEED VARIABLE
        BL @CALCSC                           CALCULATE SECONDS TO RUN

```

```

*
        MOVB @H00, @ACCHIT                   NO WORDS GUESSED THIS ROUND
        MOVB @H00, @SETHIT                   NO WORD GUESSED THIS SET
        MOVB @H00, @NOWINS                   HASN'T LOST YET
        CLR @IUISEC                          NO SECONDS ELAPSED
        CLR @HITS                             NO HITS OVERALL
        CLR @MISSES                          NO MISSES OVERALL

```

```

*
```

```

        MOV @SAVLST, @WRDLST      GET THE WORD LIST #
        BL @SCRNON                TURN ON THE SCREEN
*****
* ENTRY POINT FOR WHEN WORD LIST IS SUCCESSFULLY COMPLETED *
*****
GAMEST BL @PUTWRD                SHOW THE SET OF WORDS
        BL @SHOWSC                GO SHOW WORD LIST # AND SCORES
        LI R1, 13
        SB @SPEED, @R1LB         GIVES DELAY CYCLES
        MOV R1, WCOUNT          GET THE DELAY
        BL @DLOOP1              GO DELAY
        BL @PUTBLK              PUT OUT BLOCKS ON SCREEN
*
        CLR @ARYPNT              START AT POSITION ZERO
ILOOP  MOV @ARYPNT, R1           GET IT INTO A REGISTER
        MOV @WARRAY(R1), R2      SEE WHAT'S IN THE ARRAY AT THAT SPOT
        SRL R2, 8                PUT IT INTO LSB
        CI R2, >0OFF             IS IT EMPTY?
        JNE MAINL1              NO, SOMETHING IS THERE
        INC @ARYPNT             NOTHING THERE. TRY AGAIN.
        JMP ILOOP
*****
* MAIN PLAYING LOOP FOR WORD RADAR *
*****
MAINL1 BL @REVWRD                GO SHOW THAT ONE WORD
MAINL2 BL @MATCHW              GO GET A MATCHING WORD
*
MAINL3 BL @MOVRAD              GO SEE IF IT'S TIME TO MOVE THE RADAR
        BL @INPUT                GET SOME INPUT
        JMP MAINL3              AND KEEP LOOPING
*****
* THIS MODULE SETS ALL THE POINTERS FOR THE RADAR STARTUP *
*****
SETRAD LI RO, RADARS            POINT TO START OF LIST
        CLR @RADCNT              CLEAR RADAR COUNTER
        MOV @H00, @QTRCNT        NONE DRAWN IN THIS QUARTER
        MOV *RO+, @RADORG        NOW WE HAVE THE ORIGIN FOR FIRST QUARTER
        MOV RO, @RADPNT          POINTING TO FIRST Y, X
        MOV @H00, @TIMER        CLEAR THE TIMER BYTE, FIRST
        B *R11
*****
* THIS MODULE CREATES ARROWS AND PLACES THEM OFF SCREEN. *
*****
SETARR MOV R11, R15            SAVE RETURN
        BL @WRTVCL
        DATA SDL, 16, LARROW    PUT OUT THE PATTERNS
        LI R1, >C001            ROW, COL
        LI R2, >B006            CHAR AND COLOR
        BL @WRTVCL
        DATA SAL, 4, MYWS+2    PUT FIRST SPRITE
        C *RO+, *RO+            POINT TO NEXT SPRITE
        LI R2, >B106            CHAR AND COLOR
        BL @WRTCL
        DATA 4, MYWS+2        PUT SECOND SPRITE
        B *R15                RETURN
*****
* THIS MODULE CLEARS THE SCREEN. RETURN IS IN R15. *
*****
CLEAR  MOV R11, R15            SAVE RETURN
        LI R1, >2020            SPACE CHARS
        LI VDPADD, SIT          START AT SIT
CLEARL BL @WRTCL              WRITE OUT TWO SPACES

```

```

DATA 2,MYWS+2          FROM R1
INCT VDPADD            POINT TO NEXT TWO
CI VDPADD,SIT+>200    PAST END OF SIT?
JL CLEARL              NO, SO KEEP WRITING
CLEARR B *R15          RETURN TO CALLER
*****
* THIS MODULE GETS THE NECESSARY BLOCK #. RETURN IS 11 *
*****
GETBLK CB @DIFFIC,@H01    DIFFICULTY 1?
                        NOP
                        YES,
TRY2  CB @DIFFIC,@H02    DIFFICULTY 2?
                        NOP
                        YES
TRY3  LI RO,BLOCK3       DIFFICULTY 3
SETBLK MOV RO,@BLKPNT    SET THE POINTER
GBRTRN B *R11           RETURN TO CALLER
*****
* THIS ROUTINE FORMATS THE SCREEN (TOP 2/3). BECAUSE IT *
* CALLS HCHAR, AND VCHAR, RETURN IS IN R14. *
*****
MAKSCN MOV R11,R14      SAVE RETURN
*====> FORMAT THE TOP 1/3 HERE <====*
BL @HCHAR
DATA >0110,>0001
BL @HCHAR
DATA >0111,>0101
BL @HCHAR
DATA >0803,>0401
BL @HCHAR
DATA >081E,>0601
BL @VCHAR
DATA >0210,>0207
BL @VCHAR
DATA >0211,>0307
BL @HCHAR
DATA >0804,>050C
BL @HCHAR
DATA >0812,>050C
*====> FORMAT THE MID 1/3 HERE <====*
BL @HCHAR
DATA >1010,>0001
BL @HCHAR
DATA >1011,>0101
BL @HCHAR
DATA >0903,>0401
BL @HCHAR
DATA >091E,>0601
BL @VCHAR
DATA >0910,>0207
BL @VCHAR
DATA >0911,>0307
BL @HCHAR
DATA >0904,>050C
BL @HCHAR
DATA >0912,>050C
*
B *R14                  RETURN TO CALLER
*****
* THIS MODULE FORMATS THE CHARACTERS NEEDED FOR THE LINE *

```

```

* DRAWINT ALGORITHM FOR THE RADAR LINES. RETURN IS IN R15 *
*****
FMTRAD MOV R11,R15      SAVE RETURN
      LI R5,SIT         FORMAT TOP 1/3 TO START WITH
      LI R6,RDR1       POINT TO SIT TABLE AND REPETITIONS
RRDR  MOVB *R6+,R1     GET SIT SPOT TO WRITE
      SRL R1,8         INTO LSB OF R1
      CI R1,>00FF      TERMINATOR?
      JEQ FRENDR      YES, SO WE'RE DONE
      CI R1,>002B      START OF MID 1/3?
      JNE WRTRDR      NO, SO CONTINUE AS USUAL
*
      LI R5,SIT+>100  YES, SO INCREMENT SIT POINTER
*
WRTRDR MOVB *R6+,R7     GET THE REPETITIONS
      SRL R7,8         INTO R7
      A R5,R1          THIS POINTS US TO THE RIGHT PLACE
WRDRL MOV R1,VDPADD     SET VDPADD
      BL @WRTCL       AND WRITE OUT THE PROPER CHARACTER
      DATA 1,R1LB    FROM ITSELF
      LI WCOUNT,SIT  KEEPS SIT OFFSET
      S WCOUNT,VDPADD SIT x 8 = PDT ADDRESS FOR THIS CHAR
      SLA VDPADD,3     BLANK OUT THAT PATTERN
      BL @WRTCL
      DATA 8,ZEROES  + >2000 = COLOR TABLE ADDRESS FOR THIS CHAR
      AI VDPADD,>2000
      BL @WRTCL
      DATA 8,RADCOL  AND WRITE OUT RADAR COLORS
*
      INC R1          POINT OVER ONE CHARACTER AND ADD 1 CHAR
      DEC R7          DECREMENT LOOP COUNTER
      JNE WRDRL      NOT DONE WRITING, YET
*
      JMP RRDR       GET NEW SIT POSITION
*
FRENDR B *R15         RETURN TO CALLER
*****
* THIS MODULE PUTS THE WORD BLOCK COVERS ON THE SCREEN *
* IN THE PROPER PLACES. CALLS HCHAR, SO RETURN IS IN 14. *
*****
PUTBLK MOV R11,R14    SAVE RETURN
      MOV @BLKPNT,R1  GET WHICH BLOCK WE'RE IN
PBLOOP MOVB *R1+,R5   GET THE ROW
      CB R5,@HFF      TERMINATOR?
      JEQ PBRTRN     YES, SO EXIT THIS MODULE
      MOVB *R1+,R6   GOT THE COLUMN
      BL @HCC        GO WRITE OUT SIX CHARACTERS
      DATA >0806
      JMP PBLOOP     AND KEEP GOING
*
PBRTRN B *R14        RETURN TO CALLER
*****
* THIS MODULE CALCULATES TWO VARIABLES . . . @SAVWRD AND *
* @HITMAX. RETURN IS IN R12 *
*****
CALCWL MOV R11,R12   SAVE RETURN
      MOVB @DIFFIC,R2 GET DIFFICULTY SELECTED
      SRL R2,8        PUT INTO LSB
      SLA R2,2        x BY 4 (4,8,12)
      MOVB @R2LB,@HITMAX GOT THE MAXIMUM # OF HITS TO MATCH
*
      CLR R2         PREPARE FOR MULTIPLY

```



```

LI R0,24          MULTIPLICAND
MOVW @SAVLST,R1  WHICH WORD LIST DID THEY SELECT
SRL R1,8         PUT INTO LSB
DEC R1          ADJUST FOR ASSEMBLY
MOVW @R1LB,@ASETS THIS IS THE # OF SETS WE COMPLETED
*
CI R1,8         RANDOM WORD LIST?
JNE CWLCNT
RANDWL BL @RANDNO GET A RANDOM #
ANDI R10,>0F00
SRL R10,8
CI R10,7        GREATER THAN 8?
JH RANDWL      YES, SO TRY AGAIN
CI R10,5        LESS THAN 6?
JL RANDWL      YES, SO TRY AGAIN
MOV R10,R1     NOW HAVE A RANDOM WORD LIST
*
CWLCNT MPY R0,R1 WRD # IS IN R2
MOVW @R2LB,@SAVEWN PUT IT IN CPU
*
B *R12         RETURN TO CALLER
*****
* THIS MODULE CALCULATES THE VARIABLE @SPEEDV. *
* RETURN IS IN R12 *
*****
CALCSV MOV R11,R12 SAVE RETURN
MOVW @SPEED,R2  GET DIFFICULTY SELECTED
SRL R2,8        PUT INTO LSB
DEC R2          STARTS AT 0, NOT 1
MOVW @SPEEDS(R2),@SPEEDV GETS THE VARIABLE
B *R12         RETURN TO CALLER
*****
* THIS MODULE CALCULATES THE VARIABLE @SECLIM. *
* RETURN IS IN R12 *
*****
CALCSC MOV R11,R12 SAVE RETURN
MOVW @RUNTIM,R1 GET DIFFICULTY SELECTED
SRL R1,8        PUT INTO LSB
CLR R2          PREPARE FOR MULTIPLY
LI R0,60        MULTIPLICAND
MPY R0,R1       R2=# OF SECONDS TO GO
*
MOV R2,R1       GET # OF SECONDS
CLR R2          PREPARE FOR MULTIPLY
MPY R0,R1       R2 = # OF INTERRUPT TICKS TO GO
MOV R2,@SECLIM GETS THE LIMIT I NEED
B *R12         RETURN TO CALLER
*****
* THIS MODULE SELECTS WORDS FROM THE WORD LIST AND PLACES *
* THEM ON THE SCREEN. ALSO INITIALIZES THE WORD ARRAY. *
* RETURN IS IN R15, BUT ALSO USES 14 WITH ANOTHER ROUTINE *
*****
PUTWRD MOV R11,R15 SAVE RETURN
MOVW @SETHIT,R1 HOW MANY WORDS HAVE BEEN SHOWN IN THIS SET
SRL R1,8
CI R1,23        SET COMPLETED?
JL PWCONT      NO, SO JUST GET A NEW LIST
*
AB @H01,@ASETS YES, SO ADD 1 MORE SET DONE
CB @ASETS,@H08 DONE WITH REGULAR WORDS?
JNE TRY9       NO, BUT SEE IF WE'RE ALL DONE!
*

```

LI R0,24	MULTIPLICAND
RANDWD BL @RANDNO	GET A RANDOM #
ANDI R10,>0F00	
SRL R10,8	
CI R10,7	
JH RANDWD	
MOV R10,R1	NOW HAVE A RANDOM WORD LIST
*	
MPY R0,R1	WRD # IS IN R2
MOVB @R2LB,@SAVEWN	PUT IT IN CPU
JMP NOTALL	AND CONTINUE ON
*	
TRY9 CB @ASETS,@H09	DONE WITH ALL SETS?
JL NOTALL	NO, SO GET A NEW LIST
JMP GOTALL	YES, SO I NEED TO START OVER
*	
NOTALL AB @H01,@WRDLST	1 MORE WORD LIST DONE
JMP GODOIT	
*	
GOTALL MOVB @H00,@ASETS	START WITH SET 0 AGAIN
MOVB @H01,@WRDLST	FOR PUTTING ON SCREEN
MOVB @H00,@SAVEWN	START WITH WORD ZERO
*	
GODOIT MOVB @H00,@SETHIT	NO WORDS GUESSED IN A NEW SET
MOVB @H00,@ACCHIT	NO WORDS GUESSED, YET
*	
PWCONT LI R1,12	LOOP COUNTER
LI R3,WARRAY	POINT TO THE ARRAY
CLRARY MOVB @HFF,*R3+	CLEAR THE ARRAY
DEC R1	DONE?
JNE CLRARY	NO, SO CONTINUE WRITING
MOV @BLKPNT,R6	GET THE BLOCK LIST I NEED
PWL1 LI R5,BLOCK3	GET MASTER BLOCK LIST
CLR R7	INDEX COUNTER
MOV *R6+,R8	GETS THE FIRST BLOCK IN THE NEEDED LIST
CI R8,>FFFF	TERMINATOR?
JEQ PWRTRN	YES, SO WE'RE DONE HERE
PWL2 C R8,*R5+	ARE BLOCK THE SAME?
JEQ MATCHB	YES, SO WE FOUND THE INDEX
INC R7	NO, SO INCREMENT INDEX
JMP PWL2	AND TRY AGAIN
MATCHB MOVB @SAVEWN,@WARRAY(R7)	FILL THE DESIRED ARRAY SLOT
MOVB @SAVEWN,@WRDNUM	SHWRD USES WRDNUM
BL @SHWRD	AND SHOW THE WORD ON THE SCREEN
AB @H01,@SAVEWN	POINT TO NEXT WORD
AB @H01,@SETHIT	ONE MORE WORD SHOWN
JMP PWL1	AND KEEP GOING
*	
PWRTRN MOVB @H00,@ACCHIT	IF NEW WORDS, NONE GUESSED YET!!
B *R15	RETURN TO CALLER

* THIS MODULE SHOWS A SELECTED WORD ON THE SCREEN. R8 = *	
* ROW/COL(BASIC FORMAT):@WLANG POINTS TO START OF WORD LIST	
* @WRDNUM CONTAINS THE NUMBER OF THE WORD. RETURN IS IN 14*	

SHWRD MOV R11,R14	SAVE RETURN
MOV R8,R9	COPY ROW/COL
SRL R8,8	ROW IN 8
DEC R8	ADJUST FOR BASIC
ANDI R9,>00FF	COL IN 9
DEC R9	ADJUST FOR BASIC

SLA R8,5	x BY 32 = SIT ROW
A R9,R8	R8 = SIT POSITION TO WRITE WORD
MOV @WLANG,RLOC	START OF WORD LIST
MOVB @WRDNUM,R1	WHICH WORD #
SRL R1,8	INTO LSB OF 1
CLR R2	PREPARE FOR MULTIPLY
LI R5,6	MULTIPLICAND
MPY R5,R1	ANSWER IS IN R2
A R2,RLOC	GETS THE OFFSET
LI VDPADD,SIT	START OF SIT
A R8,VDPADD	OFFSET INTO THE SIT
LI RCOUNT,6	6 BYTES PER WORD
BL @GV	WRITE FROM GROM TO VDP
B *R14	RETURN TO CALLER

 * THIS MODULE TURNS ON THE SCREEN. RETURN IS IN R15. *

SCRNON MOV R11,R15	SAVE RETURN
BL @VWTR	WRITE TO THE REGISTER
DATA >EOB1	
MOVB @HEO,@SVVDP1	SAVE THAT VALUE OUT NOW
B *R15	RETURN TO CALLER

 * THIS MODULE CHECKS TO SEE IF IT IS TIME TO MOVE THE RADAR
 * BECAUSE LINALG USES R13, WE WILL USE R14 FOR RETURN *

MOV RAD MOV R11,R14	SAVE RETURN
CB @TIMER,@SPEEDV	TIME UP FOR THIS ROUND?
JL MRRTN	NO, SO DON'T DRAW NEW LINE

*

CLR R1	
MOVB @TIMER,@R1LB	GETS THE TIMER VALUE
A R1,@TOTSEC	GET CURRENT # OF TIMER CLICKS
C @TOTSEC,@SECLIM	ARE WE PAST THE LIMIT FOR THIS GAME?
JL MRCONT	NO, SO CONTINUE ON WITH THE GAME
B @GOGROM	YES, SO GAME IS OVER

*

MRCNT LI R0,>0002	SET PIXELS
MOV R0,@PTSTAT	
CLR @X1	CLEAR STARTING AND ENDING Y,X'S
CLR @Y1	
CLR @X2	
CLR @Y2	

*

MOVB @RADORG,@Y1LB	GETS THE STARTING Y
MOVB @RADORG+1,@X1LB	GETS THE STARTING X

*

MOV @RADPNT,R0	GET OUR POSITION IN THE LINE LIST
MOVB *R0+,@Y2LB	GETS THE ENDING Y
MOVB *R0+,@X2LB	GETS THE ENDING X
MOV R0,@RADPNT	POINTS TO NEXT LINE COORDINATE

*

BL @LINALG	GO DRAW THE LINE
BL @SOUND	EXECUTE RADAR DRAWING SOUND
DATA RDRSND	
MOVB @H00,@TIMER	AND CLEAR TIMER BYTE OUT
INC @RADCNT	THAT'S ONE MORE LINE DRAWN OVERALL
AB @H01,@QTRCNT	AND ONE MORE THIS QUARTER

*

MOV @RADCNT,R0	HOW MANY HAVE WE DRAWN
CI R0,60	HAVE WE DRAWN 60 LINES
JNE MRCNT2	NO, SO CONTINUE AS USUAL

```

*
AB @H01,@NOWINS      RADAR MADE IT'S CIRCLE!
CB @NOWINS,@H03      3 TIMES?
JNE GOERAS           NO, SO JUST ERASE RADAR
B @GGGRDM            YES, SO GAME IS OVER

*
GOERAS JMP ERASER    ERASE RADAR

*
MRCNT2 CB @QTRCNT,@HOF  HAVE WE FILLED THIS QUARTER?
JL MRRTRN           NO, SO RETURN

*
NEWQTR MOV @H00,@QTRCNT NEW QUARTER COUNTER
MOV @RADPNT,RO      GET WHERE WE WERE
MOV *RO+,@RADORG    GETS THE NEW ORIGIN
MOV RO,@RADPNT      AND SAVE THE POINTER

*
MRRTRN B *R14        RETURN TO CALLER
*****
* THIS MODULE ERASES ALL THE LINES DRAWN WHEN THE SECONDS *
* REACHES ONE MINUTE. NO RETURN LINK IS NEEDED.           *
*****
ERASER CLR @PTSTAT    RESET PIXELS
MOV @QTRCNT,@QTRCNT DID WE FINISH ON A QUARTER BOUNDARY?
JEQ ONQRTR          YES, SO SPECIAL LOGIC NEEDED
DECT @RADPNT        POINT TO LAST LINE DRAWN
ERASEL CLR @X1       CLEAR STARTING AND ENDING Y,X'S
CLR @Y1
CLR @X2
CLR @Y2

*
MOV @RADORG,@Y1LB    GETS THE STARTING Y
MOV @RADORG+1,@X1LB  GETS THE STARTING X

*
MOV @RADPNT,RO       GET OUR POSITION IN THE LINE LIST
MOV *RO+,@Y2LB       GETS THE ENDING Y
MOV *RO+,@X2LB       GETS THE ENDING X
DECT RO              POINT BACK TO SAME LINE NOW
MOV RO,@RADPNT       POINTS TO NEXT LINE COORDINATE

*
BL @LINALG           GO DRAW THE LINE
BL @SOUND            EXECUTE RADAR MOVEMENT SOUND
DATA RDRSND
LI R1,>0800          DELAY
EDLY DEC R1
JNE EDLY
SB @H01,@QTRCNT     ONE LESS IN THIS QUARTER
JNE ERCONT          NOT DONE WITH THIS QUARTER, YET

*
ONQRTR DECT @RADPNT  THIS POINTS TO CURRENT ORIGIN
MOV @RADPNT,RO      MOVE TO REGISTER
LI R1,>0020          SUBTRACT VALUE
S R1,RO             POINTS TO NEXT ORIGIN
MOV *RO,@RADORG     AND BACK TO CPU
MOV @HOF,@QTRCNT    15 MORE LINES TO DRAW

*
ERCONT DECT @RADPNT  POINTS TO A LINE COORDINATE AGAIN
DEC @RADCNT         ARE WE DONE WITH ALL LINES?
JNE ERASEL         OBVIOUSLY NOT

*
BL @SETRAD          RESET EVERYTHING
B @GAMEST           AND GET A NEW SET OF WORDS
*****

```

* INPUT ROUTINE. RETURN IS IN R13.

```
INPUT  MOV R11,R13          SAVE RETURN
      CLR @KEYBRD         START WITH KEYBOARD ZERO
      BL @SCANKEY        CHECK FOR SPACE BAR
      MOVB @KEY,R3
      SRL R3,8
      CI R3,>00FF
      JEQ BRDINC         NO KEY HIT
      CI R3,>0020        SPACE BAR?
      JNE BRDINC        NO, SO IGNORE IT
*
GDFRIT MOV R13,@TEMPRT    SAVE RETURN TEMPORARILY
      BL @DELAY4        WAIT UNTIL HE RELEASES KEY
*
      MOV @ARYPNT,R3     GET WHERE WE ARE
      CB @WARRAY(R3),@MCHNUM DO THE WORDS MATCH?
      JNE MISSED        NO, SO HE MISSED ONE
*
HIT    LI R1,>FE00        FLAG
      MOVB R1,@WARRAY(R3) SET THE ARRAY TO SAY THIS ONE MATCHED
      INC @HITS
      AB @H01,@ACCHIT    ONE MORE WITHIN THIS ROUND
*
      LI R1,4
SNDLP1 BL @SOUND
      DATA RDRSND
      BL @WRTVCL
      DATA SAL+3,1,H00   FLASH ARROWS
      C *R0+,*R0+
      BL @WRTCL
      DATA 1,H00
      LI R2,>1800
SNDLP2 DEC R2
      JNE SNDLP2
      BL @WRTVCL
      DATA SAL+3,1,H06   FLASH ARROWS
      C *R0+,*R0+
      BL @WRTCL
      DATA 1,H06
      LI R2,>1800
SNDLP3 DEC R2
      JNE SNDLP3
      DEC R1
      JNE SNDLP1
*
      BL @SHOWSC        SHOW SCORE NOW
      BL @TSTMAX        SEE IF WE'VE GUESSED ALL THE WORDS
      B @MAINL2         NO, SO GO GET ANOTHR MATCHING WORD
*
MISSED INC @MISSES      ONE MORE MISSED
      BL @SOUND
      DATA MISSND
      BL @SHOWSC
      B @INPTRT
      GO SHOW HITS AND MISSES
*
BRDINC AB @H01,@KEYBRD  SCAN NEXT KEYBOARD
      CB @KEYBRD,@H03  DONE?
      JNE BINCCCT      NO, SO CONTINUE
      B @INPTRT        YES, SO EXIT THIS MODULE
BINCCCT BL @SCANKEY    GET SOME INPUT
```

```

CLR R3
MOVB @KEY, R3
CI R3, >FF00
JEQ CHKJOY
ANY KEY PRESSED?
NOT ON THIS KEYBOARD

*
MOVB @KEY, @SAVKEY
MOV R13, @TEMPRT
BL @DELAY4
MOV @TEMPRT, R13
CLR R3
MOVB @SAVKEY, R3
WATKEY CI R3, >0200
JEQ LEFT
CI R3, >0300
JEQ RIGHT
CI R3, >1200
JEQ @GFRIT
CI R3, >0B00
JNE WRNGKY
B @GGGROM
WRNGKY JMP BRDINC
CHKJOY MOV @JOYY, @JOYY
JEQ BRDINC
MOVB @JOYY, @JOYY
JNE BRDINC
MOVB @JOYX, @SAVKEY
MOV R13, @TEMPRT
BL @DELAY5
MOV @TEMPRT, R13
MOVB @SAVKEY, @SAVKEY
JGT RIGHT
JLT LEFT
B @BRDINC
WRONG DIRECTION

*****
* JUMPS MAY ONLY BE WITHIN >100 BYTES, SO THESE NEXT LINES ALLOW *
* THOSE SUBROUTINES TO BE PLACED ANYWHERE IN THE PROGRAM. *
*****
RIGHT LI R0, 1
MOV R0, @ARRINC
B @RIGLFT
LEFT LI R0, -1
MOV R0, @ARRINC
B @RIGLFT
INCREMENT DIRECTION
INCREMENT DIRECTION

*****
* RETURN POINT FOR ALL MOVING SUBROUTINES. *
*****
INPRT B *R13
RETURN TO CALLER

*****
* ROUTINE TO MOVE THE POINTING ARROWS RIGHT AND OR LEFT. *
*****
RIGLFT MOV @ARYPNT, R1
MOVB @WARRAY(R1), R2
SRL R2, 8
CI R2, >00FE
JEQ MOVARR
BL @COVWRD
MOVARR A @ARRINC, @ARYPNT
MOV @ARYPNT, R1
CI R1, >000B
JLT NOTHI
JEQ NOTHI
CLR @ARYPNT
GET WHERE WE ARE
GET VALUE IN THE ARRAY
PUT IT INTO LSB
HAS THIS WORD BEEN ID'D?
YES, SO DON'T COVER THIS WORD UP AGAIN
NO, SO COVER THIS WORD UP
WHICH WAY TO MOVE
NO
YES, SO SET TO 0

```

```

        JMP INLIM
*
NOTHI  MOV @ARYPNT,@ARYPNT      PAST ZERO?
        JGT INLIM                NO
        JEQ INLIM                NO
        LI R1,11                 YES, SO SET TO 11
        MOV R1,@ARYPNT
*
INLIM  MOV @ARYPNT,R1            GET WHERE WE ARE
        MOV @WARRAY(R1),R2      AND SEE WHAT'S IN THE ARRAY
        SRL R2,8
        CI R2,>FF                ANYTHING THERE?
        JEQ MOVARR              NO, SO TRY AGAIN
        BL @REVWRD              YES, SO REVEAL THAT WORD
RLRTRN B @BRDINC                AND SCAN AGAIN
*****
* THIS MODULE PUTS A WORD BLOCK COVER ON THE SCREEN IN *
* IN THE PROPER PLACE. CALLS HCHAR, SO RETURN IS IN 14. *
*****
COVWRD MOV R11,R14              SAVE RETURN
        MOV @ARYPNT,R1          GET THE CURRENT ARRAY POINTER
        SLA R1,1                 x BY 2 FOR DATA STMT
        MOV @BLOCK3(R1),R5      GET THE ROW,COL
        MOV R5,R6                COPY IT
        ANDI R5,>FF00            THERES THE ROW
        ANDI R6,>00FF
        SWPB R6                  THERE'S THE COL
        BL @HCC                  GO WRITE OUT SIX CHARACTERS
        DATA >0B06
*
        B *R14                  RETURN TO CALLER
*****
* ROUTINE TO SHOW ALL 3 SCORES. RETURN IS IN 14 *
* BUT SHWDGT USES 15, TOO. *
*****
SHOWSC MOV R11,R14
        MOV @HITS,R1
        BL @SH3DGT              SHOW HITS
        DATA SIT+>2C6
        MOV @MISSES,R1
        BL @SH3DGT              SHOW MISSES
        DATA SIT+>2D7
        MOV @WRDLST,R1
        SRL R1,8                 MAKE SURE IT'S IN LOW BYTE
        BL @SHWDGT              SHOW WORD LIST #
        DATA SIT+>2AF
        B *R14                  RETURN TO CALLER
*****
* ROUTINE TO SHOW ANY 3 -DIGIT # ON IHE SCREEN. R1 MUST BE LOADED *
* WITH THE VALUE TO WRITE, AND PLACE TO BE WRITTEN IS TO BE PASSED *
* AS DATA : BL @SHWDGT RETURN IS IN REGISTER 15. *
* DATA SIT+>2C7 *
*****
SH3DGT MOV *R11+,VDPADD         SETS VDPADD
        MOV R11,R15             SAVE RETURN
        MOV R1,R2                COPY VALUE
        CLR R1                    PREPARE FOR DIVIDE
        LI R4,100                DIVISOR
        DIV R4,R1                ANS=1,REM=2
        AI R1,>30                 ASCII BIAS
        MOV @R1LB,@R5LB         HUNDREDS
*

```

```

CLR R1                PREPARE FOR DIVIDE
LI R4,10
DIV R4,R1            ANS=1, REM=2
AI R1,>30            ASCII BIAS
AI R2,>30            ASCII BIAS
SWPB R1             GOT THE 10'S DIGIT IN HI
MOV B @R2LB,@R1LB  AND THE 1'S DIGIT IN LOW
MOV R1,R6          GETS THE 10'S AND 1'S IN 6
BL @WRTCL
DATA 3,R5LB        AND WRITE IT OUT
S3RTRN B *R15
*****
* ROUTINE TO SHOW ANY TWO-DIGIT # ON THE SCREEN. R1 MUST BE LOADED *
* WITH THE VALUE TO WRITE, AND PLACE TO BE WRITTEN IS TO BE PASSED *
* AS DATA :      BL @SHWDGT      RETURN IS IN REGISTER 15.      *
*                  DATA SIT+>2C7
*****
SHWDGT MOV *R11+,VDPADD  SETS VDPADD
MOV R11,R15          SAVE RETURN
MOV R1,R2           COPY VALUE
CLR R1             PREPARE FOR DIVIDE
LI R4,10          DIVISOR
DIV R4,R1         ANS=1, REM=2
AI R1,>30        ASCII BIAS
AI R2,>30        ASCII BIAS
SWPB R1         GOT THE 10'S DIGIT IN HI
MOV B @R2LB,@R1LB AND THE 1'S DIGIT IN LOW
BL @WRTCL
DATA 2,MYWS+2    AND WRITE IT OUT
SDRTRN B *R15
*****
* ROUTINE TO DELAY FOR 1/2 SECOND. OPTIONAL REPETITION *
* IS PASSED DOWN AS DATA. RETURN IS IN R11. *
*****
DELAY MOV *R11+,WCOUNT
DLOOP1 LI WLOC,>FFFF      1/2 SECOND
DLOOP2 DEC WLOC
JNE DLOOP2
DEC WCOUNT
JNE DLOOP1
B *R11                RETURN TO CALLER
*****
* ROUTINE TO DEBOUNCE KEYS. RETURN IS IN R15 *
*****
DELAY4 MOV R11,R15      SAVE RETURN
DEBOUN BL @MOVRAD      SEE IF THE RADAR WANTS TO MOVE
BL @SCANKY            GET INPUT
CB @KEY,@HFF         IS THERE A KEY STILL PRESSED?
JNE DEBOUN          YES, SO DON'T RETURN YET
B *R15              RETURN TO CALLER
*****
* ROUTINE TO DEROUNCE KEYS. RETURN IS IN R15 *
*****
DELAYS5 MOV R11,R15    SAVE RETURN
JOYDEB BL @MOVRAD     SEE IF THE RADAR WANTS TO MOVE
BL @SCANKY           GET INPUT
MOV B @JOYX,@JOYX    IS THE JOYSTICK STILL MOVED?
JNE JOYDEB          YES, SO DON'T RETURN YET
DEBAGN BL @SCANKY
MOV @JOYY,@JOYY     HAS JOYSTICK HIT CENTER?
JNE DEBAGN          NO, SO WAIT UNTIL IT DOES
B *R15              RETURN TO CALLER

```



```

*****
* ROUTINE TO REVEAL A WORD AND MOVE SPRITES AROUND IT. *
* RETURN IS IN R15. HOWEVER, SHWRD USES 14, ALSO! *
*****
REVWRD MOV R11,R15          SAVE RETURN
      MOV @ARYPNT,R1        GET ARRAY POINTER
      MOVB @WARRAY(R1),@WRDNUM GOT THE WORD NUMBER
      SLA R1,1              MULTIPLY BY TWO FOR DATA STATEMENTS
      MOV @BLOCK3(R1),R5    GETS BLOCK TO REVEAL
      MOV R5,R8             COPY IT FOR LATER ON
      MOV R5,R6             COPY IT
      SRL R5,8              GOT Y IN 5
      ANDI R6,>00FF         GOT X IN 6
      DEC R5                ADJUST FOR BASIC
      DEC R6                ADJUST FOR BASIC
      SLA R5,3              x BY 8 GIVES SPRITE ROW
      DEC R5                SCREEN STARTS AT >FF
      SLA R6,3              x BY 8 GIVES SPRITE COL
      SB @H08,@R6LB        PUT NEXT TO WORD

*
      SWPB R5               PUT Y IN HI BYTE
      MOVB @R6LB,@R5LB     PUT X IN LO BYTE
      BL @WRTVCL
      DATA SAL,2,MYWS+10  PUT OUT LEFT ARROW SPRITE
      AI VDPADD,4          POINT TO NEXT SPRITE
      AI R5,56             PUT LEFT ARROW
      BL @WRTCL
      DATA 2,MYWS+10      PUT RIGHT ARROW
***** TEST HERE TO SEE IF WORD NEEDS REVEALING
      MOV @ARYPNT,R1
      MOVB @WARRAY(R1),R2
      SRL R2,8
      CI R2,>00FE          DON'T NEED TO REVEAL THIS WORD
      JEQ REVRTN
      BL @SHWRD           SHOW THE WORD NOW
REVRTN B *R15            RETURN TO CALLER
*****
* ROUTINE TO GET A MATCHING WORD FOR THE PLAYER TO MATCH. *
* RETURN IS IN R12. HOWEVER, SHWRD USES 14, ALSO! *
*****
MATCHW MOV R11,R12          SAVE RETURN
GETRND BL @RANDNN         GET A RANDOM #
      ANDI R10,>F000       MASK OFF
      SRL R10,12
      CI R10,11           PAST 11?
      JH GETRND          YES, SO DO IT AGAIN
      MOVB @WARRAY(R10),R2 LET'S FIND OUT WHAT'S THERE
      SRL R2,8           PUT INTO LSB
      CI R2,>00FF        NOTHING?
      JEQ GETRND        YES, SO TRY AGAIN
      CI R2,>00FE        A MATCHED FLAG?
      JEQ GETRND        YES, SO TRY AGAIN
      MOVB @R2LB,@MCHNUM PUT THIS IN THE MATCHING # CPU
      MOVB @R2LB,@WRDNUM AND GET READY FOR SHWRD
      LI R8,>120E        ROW AND COL
      BL @SHWRD         GO SHOW THE MATCHING WORD
      B *R12           RETURN TO CALLER
*****
* DETERMINE IF ALL WORDS HAVE BEEN GUESSED. RETURN IS IN 11*
*****
TSTMAX CB @ACCHIT,@HITMAX FINISHED ALL WORDS?
      JNE TMRTRN       NOT YET

```

```

        B @ERASER                YES, SO ERASE THE RADAR
TMRTRN B *R11                    RETURN TO CALLER
*****
*      RETURN TO GROM ROUTINE    *
*****
GOGROM BL @CLEAR                CLEAR THE SCREEN
        MOV B @SAVLST,@WRDLST    RESTORE ORIGINAL WORD LIST
        LIMI 0
        LWPI @PLWS
        MOV B @GPCSAV,@GRMWA
        MOV B @GPCSAV+1,@GRMWA
        MOV @GPLRA,R11
        B *R11                    RETURN TO GROM
*****
*      DRAW LINE ROUTINE        *
*****
* PIXEL MODE NO.
*
*      0 = RESET PIXEL...DON'T AFFECT COLOR BYTE
*      1 = RESET PIXEL...CHANGE BACKGROUND COLOR
*      2 = SET PIXEL...DON'T AFFECT COLOR BYTE
*      3 = SET PIXEL...CHANGE FOREGROUND COLOR
*      4 = INVERT PIXEL...DON'T AFFECT COLOR BYTE
*      5 = INVERT PIXEL...CHANGE FORGRND/BKGRND COLOR
* NOTE' IN PIXEL MODE 5, THE FOREGRND COLOR IS CHANGED IF
*       THE TARGET PIXEL WAS SET, BUT BACKGROUND COLOR
*       IS CHANGED IF THE TARGET PIXEL WAS RESET
*
*
*              DATA FORMAT FOR @PTSTAT
*
* 0| 1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| 13| 14| 15|
* | | | | | | | | | | | | | | | |
* C3|C2|C1|C0| X| X| X| X| X| X| X| X| X| P2| P1| P0|
* | | | | | | | | | | | | | | | |
* FOREGRND | BACKGRND | UNDEFINED | PIXEL |
* COLOR    | COLOR    |          | MOD. STATUS|
*
*****
POINT2 EQU $                    ENTRY POINT FOR LINE DRAW
        MOV XPTL,R0
        MOV YPTL,R1
POINT EQU $
        MOV R1,R4                MAKE A COPY OF YPOINT
        SLA R4,5                 MPY YPOINT BY 32
        SOC R1,R4                RESTORE LOWER 3 BITS
        ANDI R4,>FF07            PUT ZERO IN MIDDLE BITS
        A R0,R4                 ADD IN THE XPT
        ANDI R0,7                KEEP 3 LSBITS OF XPT
        S R0,R4                 REMOVE 3 LISBITS OF XPT
        MOV @H8000,R3
        SRC R3,0
        MOV R4,R0                MOVE VBYTE ADDR TO R0
        BLWP @VSBR              GET TARGET PATTERN BYTE
        MOV @PTSTAT,R2          GET POINT COLOR/STATUS INFO
        MOV R2,R4                COPY PIXEL MANIPULATION STATUS
        ANDI R2,>0007            STRIP OFF POSSIBLE COLOR INFO
        SLA R2,1                 MAKE THE MODE NO. A WRD INDEX
PTSET X @INSTBL(R2)            EXECUTE PROPER INST. FRM TABLE
        BLWP @VSBW              RE-STORE THE PATTERN BYTE
        AI R0,>2000              GET R0 RDY FOR COLR BYTE
        MOV @COLTBL(R2),R2      GET THE BRANCH ADDRESS

```

```

      B      *R2          AND BRANCH TO PROPER CODE
*
XCOLOR CDC  R3,R1      SEE IF PIXEL WAS SET
      JNE  BCOLOR      NO, SO CHNG BKGRND COLOR
FCOLOR BLWP @VSBR      READ THE COLOR BYTE
      ANDI R4,>FOOO    SAVE FGRND COLR INFO
      ANDI R1,>OFOO    SAVE BKGRND INFO FRM VDP
      JMP  MIXCOL
BCOLOR BLWP @VSBR      READ COLOR INFO FRM VDP
      ANDI R4,>OFOO    SAVE BKGRND COLOR INFO
      ANDI R1,>FOOO    SAVE FOREGRND COLOR INFO
MIXCOL SOC  R4,R1      MIX THE FORE/BACK COLORS
      BLWP @VSBW      RESTORE COLORS IN VDP
NCOLRT RT
*
INSTBL SZC  R3,R1
      SZC  R3,R1
      SOC  R3,R1
      SOC  R3,R1
      XOR  R3,R1
      RT
*
COLTBL DATA NCOLRT
      DATA BCOLOR
      DATA NCOLRT
      DATA FCOLOR
      DATA NCOLRT
      DATA XCOLOR

```

```

*-----*
*
*          LINE DRAWING ALGORITHM FOR CONNECTING TWO POINTS
*
*          THIS IS A SUM-TRACKING ALGORITHM
*-----*

```

```

LINALG MOV  R11,R13    SAVE RETURN ADR - JEDI
      CLR  SUM          CLR THE SUMMING REGISTER
DODRAW LI   M,>0001    INIT THE X INCREMENT
      LI   N,>0001    INIT THE Y INCREMENT
      MOV  @X2,D       COPY THE 2ND X POINT
      MOV  @X1,XPTL    COPY THE 1ST X POINT
      S    XPTL,D      SUBTRACT TO GET XDELTA
      JEQ  SETS        IF XDELTA IS ZERO, JUMP
      JGT  NOSNEG      IF XDELTA IS POS., JUMP
NEGXFG NEG  M          MAKE THE INCREMENT NEG.
      NEG  D           MAKE THE XDELTA POSITIVE
DZERO  MOV  D,D        CHECK XDELTA=0
      JNE  NOSNEG      JUMP IF POSITIVE
SETS   SETO SUM       MAKE SUM REGISTER = -1
NOSNEG MOV  @Y2,E      COPY THE 2ND Y POINT
      MOV  @Y1,YPTL    COPY THE 1ST Y POINT
      S    YPTL,E      CALCULATE THE YDELTA
      JEQ  SUMFUJ      JUMP IF DELTA = 0
      JGT  SUMFUJ      JUMP IF DELTA > 0
YNEG   NEG  N          MAKE THE YINCREMENT NEG.
      NEG  E           MAKE THE YDELTA POSITIVE
SUMFUJ EQU  $
      C    D,E         COMPARE X&Y DELTAS
      JEQ  NUADJ       DUN'T ADJ IF NOT NEEDED
      JLT  YSUMFG      IF DX<DY FUDGE SUM W/ DY DATA
      MOV  D,SUM       COPY XDELTA INTO SUM
      JMP  SHFSUM      JUMP AND SHIFT SUM(DIV BY 2)

```

```

YSUMFG MOV E, SUM          COPY YDELTA
        NEG SUM           MAKE DY IN SUM NEGATIVE
SHFSUM SRA SUM, 1        DIVIDE SUM BY 2
NOADJ EQU $
PTPASS EQU $
        BL @POINT2      OPERATE ON ONE POINT
        C XPTL, @X2     CHECK FOR ENDING XPOINT
        JNE SCHECK     SKIP YCHECK IF NOT EQUAL
        C YPTL, @Y2     CHECK FOR ENDING YPOINT
        JNE SCHECK     DO NEXT POINT IF NOT EQU
        B *R13         RETURN TO CALLER - JEDI
SCHECK MOV SUM, SUM      CHECK SIGN OF SUM REG.
        JLT NEWY       IF LESS THAN, CHANGE Y
NEWX   A M, XPTL        ADD XINC TO XPOINT
        S E, SUM       SUBTR YDELTA FROM SUM
        JMP PTPASS     JMP AND MODIFY THE POINT
NEWY   A N, YPTL        ADD YINC TO YPOINT
        A D, SUM       ADD XDELTA TO THE SUM
        JMP PTPASS     JMP AND MODIFY THE POINT

```

* DRAWS A VERTICAL LINE OF CHARACTERS *

* SIMULATION OF THE BASIC SUBROUTINE *

* 'CALL VCHAR'. THE PARAMETERS ARE *

* PASSED FOLLOWING THE BL @HCHAR SIMI *

* IN DATA STMTS OR REGISTERS MAY BE *

* PRELOADED AND BL TO ONE OF THE ENTRY*

* POINTS WITHIN THE SUBROUTINE. I.E. : *

* BL @VCHAR *

* DATA >0A0A, >2A10 *

* FORMAT IS ROW:COL::ASCII:REPETITION *

* NOTE . . . THE FIRST ROW AND COLUMN *

* ON THE SCREEN IS 1, NOT 0. A VALUE *

* OF ZERO IN THESE POSITIONS CAUSES *

* THE SUBROUTINE TO CRASH AND VDP GO *

* 'CRAZY'. RETURN LINKAGE IS IN R15 *

```

VCHAR  MOVB *R11+, R5      START ROW
VCCC   MOVB *R11+, R6      START COL
VCC    MOVB *R11+, R7      CHARACTER VALUE
VC     MOVB *R11+, R8      # OF REPETITIONS
V      MOV R11, R15       SAVE RETURN LINK
*

```

```

        SRL R5, 8          }
        SRL R6, 8          } RIGHT JUSTIFY
        SRL R8, 8          }
        DEC R5             NOW OFFSET FOR ACTUAL ROW
        DEC R6             SAME FOR COLUMN
        SLA R5, 5          x32 FOR SIT ADDRESS OF ROW
        A R6, R5          ADD COLUMN VALUE FOR SIT ADDRESS
*

```

```

VCHARL LI VDPADD, SIT     GET START OF SIT
        A R5, VDPADD     ADD THE OFFSET INTO IT
        BL @WRTCL       WRITE YOUR CHARACTER
        DATA 1, MYWS+14
        AI R5, 32        POINT TO NEXT VERT POSITION
        CI R5, SIT+>2FF  PAST END OF SIT?
        JLE VLGCONT     NO
*

```

```

        LI R6, >0300     LENGTH OF SIT
        S R6, R5        POINT NOW TO FIRST ROW AGAIN

```

```

        INC R5                AND MOVE OVER 1 COLUMN
*
VLCONT DEC R8                MET OUR REPETITIONS, YET?
        JNE VCHARL           NO, SO KEEP WRITING
*
VCHART D *R15                RETURN TO CALLER
*****
* DRAWS A HORIZONTAL LINE OF CHARACTERS
*****
* SIMULATION OF THE BASIC SUBROUTINE *
* 'CALL HCHAR'. THE PARAMETERS ARE *
* PASSED FOLLOWING THE BL @HCHAR STMT *
* IN DATA STMTS OR REGISTERS MAY BE *
* PRELOADED AND BL TO ONE OF THE ENTRY*
* POINTS WITHIN THE SUBROUTINE. I.E. : *
*   BL @HCHAR                *
*   DATA >OAOA,>2A10        *
* FORMAT IS ROW:COL:ASCII:REPETITION *
* NOTE . . . THE FIRST ROW AND COLUMN *
* ON THE SCREEN IS 1, NOT 0. A VALUE *
* OF ZERO IN THESE POSITIONS CAUSES *
* THE SUBROUTINE TO CRASH AND VDP GO *
* 'CRAZY'. RETURN LINKAGE IS IN R15 *
*****
HCHAR  MOVB *R11+,R5          START ROW
HCCC   MOVB *R11+,R6          START COL
HCC    MOVB *R11+,R7          CHARACTER VALUE
HC     MOVB *R11+,R8          # OF REPETITIONS
H      MOV  R11,R15           SAVE RETURN LINK
*
        SRL R5,8              }
        SRL R6,8              } RIGHT JUSTIFY
        SRL R8,8              }
        DEC R5                NOW OFFSET FOR ACTUAL ROW
        DEC R6                SAME FOR COLUMN
        SLA R5,5              x32 FOR SIT ADDRESS OF ROW
        A R6,R5              ADD COLUMN VALUE FOR SIT ADDRESS
*
HCHARL LI VDPADD,SIT         GET SIT START
        A R5,VDPADD          ADD THE OFFSET
        BL @WRTCL           WRITE YOUR CHARACTER
        DATA 1,MYWS+14     POINT TO NEXT POSITION
        INC R5              PAST END OF SIT?
        CI R5,SIT+>2FF      NO
        JLE HLCONT
*
        LI R6,>0300          LENGTH OF SIT
        S R6,R5             POINT NOW TO FIRST ROW AGAIN
*
HLCONT DEC R8                MET OUR REPETITIONS, YET?
        JNE HCHARL           NO, SO KEEP WRITING
*
HCHART B *R15                RETURN TO CALLER
*****
* STANDARD KEYBOARD SCAN ROUTINE *
*****
SCANKY LIM1 0                DISABLE INTS
        LWPI GPLWS           GPLWS EQU >83E0
        BL @SCAN            SCAN EQU >000E
        LWPI MYWS
        LIM1 2              ENABLE INTS

```

```

      B *R11                RETURN TO CALLER
*****
* THIS IS THE MODULE THAT GENERATES A RANDOM NUMBER. IS IS *
* PASSED IN REGISTER 10. RETURN LINKAGE IS R11             *
*****
RANDNO LI R9,28645
      MPY @SEED,R9
      AI R10,31417
      MOV R10,@SEED                MAKE A NEW SEED
      B *R11                        AND RETURN TO CALLER
*****
* CALL SOUND ROUTINE *
*****
SOUND MOV *R11+,SNDPNT            LOAD DATA POINTER
      LIM1 0
      SOCB @HO1,@RAMFLG          SOUND LIST IS IN VDPRAM
      MOVB @HO1,@STRTSD          START THE SOUND PROCESSING
      LIM1 2
      B *R11
*****
* SINGLE BYTE WRITE TO VDP *
* CALLED BY BLWP @VSBW *
* 1. MUST HAVE SECOND WORKSPACE *
* 2. VDPADD MUST BE PRELOADED *
* 3. MSB OF R1 MUST BE PRELOADED WITH *
* VALUE TO BE WRITTEN *
* 4. R13,14,15 OF 2ND WS MUST BE *
* PRESERVED *
* 5. R1 OF NEW WORKSPACE IS DESTROYED *
*****
VSBW DATA MYWS2,VSBWGO NEW WORKSPACE,PC
VSBWGO LIM1 0                DISABLE INTS
      MOV *R13,R1            GETS VDPADD FROM OLD R0
      MOVB @R1LB2,@VDPWA R1LB2 EQU MYWS2+3
      ORI R1,>4000           SET WRITE FLAG
      MOVB R1,@VDPWA        SET UP HIGH BYTE
      NOP                   VDPWA EQU >8C02
      MOVB @2(R13),@VDPWD VALUE FROM OLD R1 MSB
*                               VDPWD EQU >8C00
      LIM1 2                ENABLE INTS
      RTWP                  RETURN AND RESTORE ME
*****
* SINGLE BYTE READ FROM VDP *
* CALLED BY BLWP @VSBW *
* 1. MUST HAVE SECOND WORKSPACE *
* 2. VDPADD MUST BE PRELOADED *
* 3. MSB OF R1 IS WHERE THE BYTE *
* WILL BE WRITTEN *
* 4. R13,14,15 OF 2ND WS MUST BE *
* PRESERVED *
* 5. R1 OF NEW WORKSPACE IS DESTROYED *
*****
VSBW DATA MYWS2,VSBWGO NEW WORKSPACE,PC
VSBWGO LIM1 0                DISABLE INTS
      MOV *R13,R1            GETS VDPADD FROM OLD R0
      MOVB @R1LB2,@VDPWA R1LB2 EQU MYWS2+3
      NOP                   WASTE SOME TIME
      MOVB R1,@VDPWA        SET UP HIGH BYTE
      NOP                   VDPWA EQU >8C02
      MOVB @VDPRD,@2(R13) VALUE TO OLD R1
*                               VDPRD EQU >8800
      LIM1 2                ENABLE INTS

```

RTWP

RETURN AND RESTORE ME

```
*****
* VDP WRITE TO REGISTER ROUTINE. THE*
* PARAMETER IS PASSED DOWN AS DATA *
* USING THIS FORMAT: (LB, THEN HB) *
* VALUE TO WRITE/8/VDP REG. # *
* I.E. BL @VWTR *
* DATA >1187 *
* THIS INSTANCE CHANGES VDP REG. 7 *
* TO >11 (CHANGES BACKGROUND COLOR) *
* RETURN LINKAGE IS R11. *
```

```
*****
VWTR MOV *R11+,R1 GET THE DATA
VWTRGO LIM1 0 DISABLE INTS
MOV B R1,@VDPWA VDPWA EQU >8C02
SWPB R1 NOW SET UP HIGH BYTE
MOV B R1,@VDPWA
SWPB R1 RESTORE PARAMETER
LIM1 2 ENABLE INTERRUPTS
B *R11 AND RETURN TO CALLER
```

```
*****
* W R I T E T O V D P ROUTINE *
* THIS ROUTINE ASSUMES THAT THE *
* PARAMETERS ARE PASSED FOLLOWING *
* THE BL @WRTVCL STATEMENT. I.E.: *
* BL @WRTVCL *
* DATA SAL, 2, MYWS+2 *
```

```
*****
WRTVCL MOV *R11+,VDPADD VDPADD EQU 0 (R3)
WRTCL MOV *R11+,WDCOUNT WDCOUNT EQU 2 (R2)
WRTL MOV *R11+,WLDC WLDC EQU 3 (R3)
WRITE LIM1 0 DISABLE INTS
SWPB VDPADD
MOV B VDPADD,@VDPWA >8C02
SWPB VDPADD WASTE SOME TIME
ORI VDPADD,>4000 SET FLAG TO WRITE
MOV B VDPADD,@VDPWA >8C02
NOP WASTE SOME TIME
W00010 MOV B *WLDC+,@VDPWD VDPWD EQU >8C00
DEC WDCOUNT
JGT W00010
LI WDCOUNT,>4000
S WDCOUNT,VDPADD RESTORE VDPADD
LIM1 2 ENABLE INTS
B *R11 RETURN TO CALLER
```

```
*****
* R E A D F R O M V D P ROUTINE *
* THIS ROUTINE ASSUMES THAT THE *
* PARAMETERS ARE PASSED FOLLOWING *
* THE BL @RDVCL STATEMENT. I.E.: *
* BL @RDVCL *
* DATA SAL, 2, MYWS+2 *
```

```
*****
RDVCL MOV *R11+,VDPADD VDPADD EQU 0 (R0)
RDCL MOV *R11+,RCOUNT RCOUNT EQU 2 (R2)
RDL MOV *R11+,RLDC RLDC EQU 3 (R3)
READ LIM1 0 DISABLE INTS
SWPB VDPADD
MOV B VDPADD,@VDPWA >8C02
SWPB VDPADD WASTE SOME TIME
MOV B VDPADD,@VDPWA >8C02
NOP WASTE SOME TIME
```


TITLE WORDRADAR

GROM 3

ORG 0

**** DEVELOPMENTAL LEARNING MATERIALS ****

GRMHDR DATA >AA, 1, 1, 0, 0, 0, #PROG, 0, 0, 0, 0, 0, 0, 0, 0

PROG DATA #0, #TITLE, 7, : ENGLISH:

** CPU RAM ALLOCATIONS

* TEMPORARY WORKSPACE EQUATES *

R0 EQU >00

R0LB EQU >01

R1 EQU >02

R1LB EQU >03

R2 EQU >04

R2LB EQU >05

R3 EQU >06

R3LB EQU >07

R4 EQU >08

R4LB EQU >09

R5 EQU >0A

R5LB EQU >0B

R6 EQU >0C

R6LB EQU >0D

R7 EQU >0E

R7LB EQU >0F

R8 EQU >10

R8LB EQU >11

R9 EQU >12

R9LB EQU >13

R10 EQU >14

R10LB EQU >15

R11 EQU >16

R11LB EQU >17

R12 EQU >18

R12LB EQU >19

R13 EQU >1A

R13LB EQU >1B

R14 EQU >1C

R14LB EQU >1D

R15 EQU >1E

R15LB EQU >1F

*

RUNTIM EQU >2E 1 RUN TIME IN MINUTES 1-5

JOYST? EQU >2F 1 JOY/KEY OPTION 0-KEYBOARD

SAVLST EQU >31 1 SAVED WORD LIST

LANGNM EQU >43 1 WHICH LANGUAGE IS BEING USED *
0-ENG 1-FR 2-GER 3-DUT 4-IT*

LANG EQU >44 2 LANGUAGE POINTER FOR TEXT *

CNTFMT EQU >46 2 TEMPORARY WORKSPACE FOR WORD *
POINTER *

MOV TAB EQU >48 2 GROM ADDRESS OF DATA TABLE *

TAB1 EQU >4A 2 GROM ADDRESS OF PATTERN *

TAB2 EQU >4C 2 VDP RAM ADDRESS OF PATTERN *

BYTE EQU >4E 2 COUNT OF # OF BYTES TO MOVE *

CB/ADD EQU >50 2 TEXT: CHAR BUFFER ADDRESS *

```

FMT      EQU  >52      1      TEXT:FORMATTING CODE      *
FMT1     EQU  >53      1      TEXT:FORMATTING CODE      *
LENGTH   EQU  >54      2      TEXT:WORD LENGTH          *
SPEED    EQU  >58      1      SPEED OF RADAR              *
WRDLST   EQU  >59      1      WORD LIST                    *
LIMIT    EQU  >5A      1      X - PIXEL LIMIT              *
MAXPRB   EQU  >5B      1      MAXIMUM # OF PROBLEMS AT      *
*                                                CURRENT LEVEL          *
FLAG     EQU  >5C      1      FLAG FOR MENKEY              *
DIFFIC   EQU  >5D      1      DIFFICULTY LEVEL            *
SOUND    EQU  >5E      2      SOUND ADDRESS                *
WLANG    EQU  >68      2      WORD LIST POINTER            *
HITS     EQU  >AC      2      # OF HITS                    *
MISSES   EQU  >AE      2      # OF MISSES                  *
LHITS    EQU  >B0      2      LOWEST # OF HITS              *
LMISS    EQU  >B2      2      LOWEST # OF MISSES            *
HHITS    EQU  >B4      2      HIGHEST # OF HITS             *
HMISS    EQU  >B6      2      HIGHEST # OF MISSES           *
NMJY?    EQU  >B8      1      WHICH JOYSTICK IS BEING      *
*                                                SCANNED                *
*****STATUS BLOCK DEFINITIONS*****
VRAM     EQU  >70      TOTAL VIDEOD RAM AVAILABLE *
DATSTK   EQU  >72      DATA STACK POINTER          *
SUBSTK   EQU  >73      SUBROUTINE STACK POINTER        *
KEYBRD   EQU  >74      KEYBOARD NUMBER                *
KEY       EQU  >75      KEYCODE                      *
JOYY     EQU  >75      JOYSTICK Y                      *
JOYX     EQU  >76      JOYSTICK X                      *
RANDOM    EQU  >78      RANDOM NUMBER GENERATED        *
TIMER    EQU  >79      TIMER VALUE                    *
MOTION   EQU  >7A      SPRITE MOTION VALUE            *
VDPST    EQU  >7B      VDP STATUS BYTE                *
STATUS   EQU  >7C      CPU STATUS FLAGS                *
KEEPR1   EQU  >D4      TO KEEP VDP R1                  *
***** MEMORY MAPPED EQUATES *****
PDT      EQU  >0000     PATTERN DESCRIPTOR TABLE      *
SIT      EQU  >1800     SCREEN IMAGE TABLE          *
CT       EQU  >2000     COLOR TABLE                  *
*****MONITOR LOCATIONS*****
CHR2     EQU  >18
FAC      EQU  >4A
*****COLOR DEFINITIONS*****
TRANSP   EQU  >0      TRANSPARENT                      *
BLACK    EQU  >1      BLACK                            *
MDGRN    EQU  >2      MEDIUM GREEN (2)                 *
LTGRN    EQU  >3      LIGHT GREEN (1)                  *
DKBLU    EQU  >4      DARK BLUE (2)                   *
MDBLU    EQU  >5      MEDIUM BLUE (1)                  *
DKRED    EQU  >6      DARK RED (3)                    *
CYAN     EQU  >7      BRIGHT, LIGHT BLUE              *
MDRED    EQU  >8      MEDIUM RED (2)                   *
LTRED    EQU  >9      LIGHT RED (1)                    *
DKYLW    EQU  >A      DARK YELLOW (2)                  *
LTYLW    EQU  >B      MEDIUM YELLOW(1)                 *
DKGRN    EQU  >C      DARK GREEN (3)                   *
MAGNTA   EQU  >D      MAGENTA (PURPLISH RED)           *
GRAY     EQU  >E      GRAY                            *
WHITE    EQU  >F      WHITE                            *
*
AID      EQU  >01      CONTROL: AID                      *
BEGIN    EQU  >0E      BEGIN                          *
*****

```

```

*
*      VDP  EQUATES
*
*****
AIDAD EQU >1000      2      ADDRESS LABEL OF AID FNCTN
*      EQU >1001
BEGAD EQU >1002      2      ADDRESS LABEL OF BEGIN FNCTN
*      EQU >1003
*****TEXT EQUATES*****
***** TITLE SCREEN*****
WARCSK EQU 0          ARCADEMIC SKILLBUILDER
WNAME EQU 1          ALLIGATOR MIX
***** PRESS ANY KEY SCREEN *****
WDUMY1 EQU 2          DUMMY LINE (MULTI-LING)
WPRAKY EQU 3          PRESS ANY KEY
WDUMY2 EQU 4          DUMMY LINE (MULTI-LING)
WTOPLY EQU 5          TO PLAY
WAIDGO EQU 6          AID = GAME OPTIONS
***** GAME CONTROL OPTIONS SCREEN *****
WNAME1 EQU 7          ALLIGATOR MIX
WGCO EQU 8           GAME CONTROL OPTIONS
WSPEED EQU 9          SKILL LEVEL 1-9
WPRBRN EQU 10         PROBLEM RANGE 3,6,9
WRUNTM EQU 11         RUN TIME (MIN) 1-5
WDIFF1 EQU 12         DIFFICULTY? 1-3
WEGCIL EQU 13         EXIT GAME CONTROL
WPNOD EQU 14          PRESS NUMBER OF OPTION
WYWTC EQU 15          YOU WISH TO CHANGE
***** GAME SCREEN *****
WHITS EQU 16          HITS (GAME SCREEN)
WMISS EQU 17          MISSES (GAME SCREEN)
***** RECORD YOUR SCORE SCREEN *****
WNAME2 EQU 18         ALLIGATOR MIX
WRCYSC EQU 19         RECORD YOUR SCORE
WCURNT EQU 20         CURRENT
WLOW EQU 21           LOW
WHIGH EQU 22          HIGH
WHITS1 EQU 23         HITS
WMISS1 EQU 24         MISSES
WPAKTC EQU 25         PRESS ANY KEY TO CONTINUE
WENERG EQU 26         ENERGY
WHITS2 EQU 27         HITS
WMISS2 EQU 28         MISSES
WSCANN EQU 29         SCANNER
THRU EQU >FF          END OF FMT
*****
* GENERAL DATA AREA FOR INITIALIZATIONS
*****
VDPNDR DATA >00,>E0,>00,>0E,>01,>06,>00,>00
VDPBIT DATA >02,>A0,>06,>FF,>03,>36,>03,>04
*****
*** TITLE SCREEN PATTERNS *
*****
PAT00 DATA >00,>00,>00,>00,>00,>00,>00,>00
PAT01 DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF
PAT02 DATA >00,>00,>00,>00,>0F,>0F,>0F,>0F
PAT03 DATA >00,>00,>00,>00,>FF,>FF,>FF,>FF
PAT04 DATA >00,>00,>00,>00,>FB,>FC,>FE,>FF
PAT05 DATA >0F,>0F,>0F,>0F,>0F,>0F,>0F,>0F
PAT06 DATA >FF,>80,>80,>80,>80,>80,>80,>80
PAT07 DATA >FF,>3F,>1F,>1F,>1F,>1F,>1F,>1F
PAT08 DATA >80,>80,>80,>FF,>FF,>FF,>FF,>FF

```

```

PAT09 DATA >1F,>1F,>3F,>FF,>FF,>FE,>FC,>F8
PAT0A DATA >00,>00,>00,>00,>80,>80,>80,>80
PAT0B DATA >80,>80,>80,>80,>80,>80,>80,>80
PAT0C DATA >07,>00,>00,>FF,>FF,>FF,>FF,>FF
PAT0D DATA >FF,>CF,>C7,>C7,>C3,>C3,>C1,>C1
PAT0E DATA >00,>80,>80,>C0,>C0,>E0,>E0,>F0
PAT0F DATA >F0,>F9,>79,>7F,>3F,>3F,>1F,>1F
PAT10 DATA >00,>00,>00,>00,>80,>C0,>E0,>F0
PAT11 DATA >00,>00,>00,>00,>01,>03,>07,>0F
PAT12 DATA >F0,>E0,>C0,>80,>00,>00,>00,>00
PAT13 DATA >0F,>07,>03,>01,>00,>00,>00,>00
PAT14 DATA >0F,>1F,>1E,>3E,>3C,>7C,>78,>F8
PAT15 DATA >F0,>F0,>E0,>E0,>C0,>C0,>80,>80
PAT16 DATA >00,>00,>00,>00,>FE,>FE,>FE,>FE
PAT17 DATA >FE,>3E,>3E,>3E,>3E,>3E,>3E,>3E
PAT18 DATA >3E,>3E,>3E,>3E,>3E,>3E,>3E,>3E
PAT19 DATA >00,>00,>01,>03,>03,>03,>01,>00
PAT1A DATA >00,>F0,>F8,>FC,>FC,>FC,>FB,>F0
PAT1B DATA >00,>FF,>FF,>01,>01,>01,>01,>01
PAT1C DATA >00,>FF,>FF,>F8,>F8,>F8,>F8,>F8
PAT1D DATA >03,>03,>03,>07,>07,>07,>FF,>FF
PAT1E DATA >FC,>FC,>9C,>9E,>0E,>0E,>0F,>0F
PAT1F DATA >00,>00,>00,>00,>00,>00,>FF,>FF
PAT20 DATA >00,>FF,>FF,>00,>00,>00,>00,>00
PAT21 DATA >00,>0F,>1F,>3F,>3F,>3F,>1F,>0F
PAT22 DATA >00,>00,>80,>C0,>C0,>C0,>80,>00
PAT23 DATA >00,>FF,>FF,>1F,>1F,>1F,>1F,>1F
PAT24 DATA >00,>FF,>FF,>80,>80,>80,>80,>80
PAT25 DATA >0F,>0F,>07,>79,>70,>70,>F0,>F0
PAT26 DATA >C0,>C0,>C0,>E0,>E0,>E0,>FF,>FF
PAT27 DATA >18,>3C,>7E,>FF,>FF,>FF,>FF,>FF
PAT28 DATA >FF,>FF,>FF,>FF,>FF,>7E,>3C,>18
PAT29 DATA >1F,>3F,>7F,>FF,>FF,>7F,>3F,>1F
PAT2A DATA >FB,>FC,>FE,>FF,>FF,>FE,>FC,>FB
PAT2B DATA >00,>00,>00,>00,>07,>07,>07,>07
PAT2C DATA >07,>07,>07,>07,>07,>07,>07,>07
PAT2D DATA >C0,>00,>00,>C0,>C0,>C0,>C0,>C0
LNGPT DATA >00,>44,>00,>3B,>44,>7C,>44,>44
        DATA >00,>44,>00,>44,>44,>44,>44,>7C

```

```

*****
* CHARACTERS FOR UPPER 1/3 PLAYING SCREEN DEFINED HERE! *
*****

```

```

TOP1/3 DATA >FF,>D2,>12,>12,>02,>02,>02,>02          CHAR 0
        DATA >FF,>4B,>4B,>4B,>40,>40,>40,>40          CHAR 1
        DATA >00,>02,>02,>02,>02,>02,>02,>00          CHAR 2
        DATA >00,>40,>40,>40,>40,>40,>40,>00          CHAR 3
        DATA >A0,>A0,>E0,>E0,>E0,>E0,>FF,>E0          CHAR 4
        DATA >00,>00,>00,>00,>00,>00,>24,>00          CHAR 5
        DATA >05,>05,>07,>07,>07,>07,>FF,>07          CHAR 6
        DATA >00,>FF,>FF,>FF,>FF,>FF,>FF,>00          CHAR 8

```

```

*=====> THOSE 64 BYTES GO INTO PDT+>0000

```

```

*****
* CHARACTERS FOR MIDDLE THIRD PLAYING SCREEN DEFINED HERE*
*****

```

```

MID1/3 DATA >02,>02,>02,>02,>12,>12,>D2,>FF          CHAR 0
        DATA >40,>40,>40,>40,>4B,>4B,>4B,>FF          CHAR 1
        DATA >00,>02,>02,>02,>02,>02,>02,>00          CHAR 2
        DATA >00,>40,>40,>40,>40,>40,>40,>00          CHAR 3
        DATA >E0,>FF,>E0,>E0,>E0,>E0,>A0,>A0          CHAR 4
        DATA >00,>24,>00,>00,>00,>00,>00,>00          CHAR 5
        DATA >07,>FF,>07,>07,>07,>07,>05,>05          CHAR 6
        DATA >00,>FF,>FF,>FF,>FF,>FF,>FF,>00          CHAR 8

```

*=====> THOSE 64 BYTES GO INTO PDT+>0800

* CHARACTERS FOR BOTTOM THIRD PLAYING SCREEN DEFINED HERE*

BOT1/3 DATA >FF,>00,>00,>00,>00,>00,>00,>00 CHAR 0
DATA >01,>02,>04,>08,>10,>20,>40,>80 CHAR 1
DATA >80,>40,>20,>10,>08,>04,>02,>01 CHAR 2
DATA >80,>80,>80,>80,>80,>80,>80,>80 CHAR 3
DATA >01,>01,>01,>01,>01,>01,>01,>01 CHAR 4
DATA >FF,>80,>80,>80,>80,>80,>80,>80 CHAR 5
DATA >FF,>01,>01,>01,>01,>01,>01,>01 CHAR 6

*=====> THOSE 56 BYTES GO INTO PDT+>1000

**** THESE NEXT 4 CHARACTERS GO INTO PDT @>ADB (CHAR SET 11)*

AUML DATA >00,>88,>00,>20,>50,>FB,>88,>88 CHAR >5B A UMLAUT
UUML DATA >00,>88,>00,>90,>90,>90,>90,>68 CHAR >5D O UMLAUT
OUML DATA >00,>88,>00,>70,>88,>88,>88,>70 CHAR >5C O UMLAUT
AUMLS DATA >00,>20,>00,>20,>50,>FB,>88,>88 CHAR >5E A UMLAUT SWEDE

* FINISH UP BOTTOM 1/3 PLAYING CHARACTERS HERE *

FIN1/3 DATA >FF,>FF,>FF,>FF,>7F,>7F,>3F,>1F CHAR 8
DATA >FB,>FB,>FB,>FB,>FB,>FB,>FB,>FB CHAR 9
DATA >1F,>1F,>1F,>1F,>1F,>1F,>1F,>1F CHAR 10
DATA >FB,>FC,>FE,>FE,>FF,>FF,>FF,>FF CHAR 11
DATA >00,>00,>00,>00,>00,>C0,>E0,>FF CHAR 12
DATA >00,>00,>00,>00,>00,>03,>07,>FF CHAR 13
DATA >1F,>3F,>7F,>7F,>FF,>FF,>FF,>FF CHAR 14
DATA >FF,>FF,>FF,>FF,>FE,>FE,>FC,>FB CHAR 15
DATA >FF,>E0,>C0,>00,>00,>00,>00,>00 CHAR 16
DATA >FF,>07,>03,>00,>00,>00,>00,>00 CHAR 17
DATA >FF,>FC,>FB,>E0,>E0,>FB,>FC,>FF CHAR 18
DATA >FF,>3F,>1F,>07,>07,>1F,>3F,>FF CHAR 19
DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF CHAR 20
DATA >01,>03,>07,>0F,>1F,>3F,>7F,>FF CHAR 21
DATA >80,>C0,>E0,>F0,>FB,>FC,>FE,>FF CHAR 22
DATA >FF,0,0,0,0,0,0,0 CHAR 23 TOP CIRCLE
DATA 0,0,0,0,0,0,0,>FF CHAR 24 BOT CIRCLE

*=====> THOSE 120 BYTES GO INTO PDT+>1040

ZER0ES DATA 0,0,0,0,0,0,0,0 ALL ZEROES
DATA 0,0,0,0,0,0,0,0 TO BLANK OUT
DATA 0,0,0,0,0,0,0,0 CHARS INTERFERING
DATA 0,0,0,0,0,0,0,0 W/ LINE DRAWER

* DEFINE COLORS NEEDED FOR PLAYING AREA *

WT DATA >F0,>F0,>F0,>F0,>F0,>F0,>F0,>F0 WHT ON TRANS
GW DATA >CF,>CF,>CF,>CF,>CF,>CF,>CF,>CF GRN ON WHT
GT DATA >C0,>C0,>C0,>C0,>C0,>C0,>C0,>C0 GRN ON TRANS

* BOTTOM 1/3 PLAYING AREA SCREEN IMAGE TABLE FORMAT *

BOTSIT DATA >20,>20,>20,>20,>20,>20,>20,>20 * LINE 0
DATA >20,>20,>20,>20,>15,>00,>00,>00
DATA >00,>00,>00,>16,>20,>20,>20,>20
DATA >20,>20,>20,>20,>20,>20,>20,>20

*
DATA >20,>20,>20,>20,>20,>20,>20,>20 * LINE 1
DATA >20,>20,>20,>15,>12,>14,>14,>14
DATA >14,>14,>14,>13,>16,>20,>20,>20
DATA >20,>20,>20,>20,>20,>20,>20,>20

```

*
DATA >20,>20,>20,>20,>20,>20,>15,>00 * LINE 2
DATA >00,>00,>00,>02,>14,>14,>14,>14
DATA >14,>14,>14,>14,>01,>00,>00,>00
DATA >00,>16,>20,>20,>20,>20,>20,>20
*
DATA >20,>20,>20,>20,>20,>15,>14,>14 * LINE 3
DATA >14,>14,>14,>14,>02,>14,>14,>14
DATA >14,>14,>14,>01,>14,>14,>14,>14
DATA >14,>14,>16,>20,>20,>20,>20,>20
*
DATA >20,>20,>20,>15,>14,>14,>14,>14 * LINE 4
DATA >14,>14,>14,>14,>14,>02,>0F,>10
DATA >11,>08,>01,>14,>14,>14,>14,>14
DATA >14,>14,>14,>00,>16,>20,>20,>20
*
DATA >20,>20,>05,>14,>14,>0F,>10,>17 * LINE 5
DATA >11,>08,>14,>14,>14,>01,>09,>20
DATA >20,>0A,>02,>14,>14,>14,>0F,>10
DATA >17,>11,>08,>14,>14,>06,>20,>20
*
DATA >20,>20,>03,>14,>14,>09,>20,>20 * LINE 6
DATA >20,>0A,>14,>14,>01,>14,>0B,>0C
DATA >0D,>0E,>14,>02,>14,>14,>09,>20
DATA >20,>20,>0A,>14,>14,>04,>20,>20
*
DATA >20,>20,>03,>14,>14,>0B,>0C,>1B * LINE 7
DATA >0D,>0E,>14,>01,>14,>14,>14,>14
DATA >14,>14,>14,>14,>02,>14,>0B,>0C
DATA >1B,>0D,>0E,>14,>14,>04,>20,>20
*
*====> THAT WAS 256 BYTES
*****
*****
*****
*****          DEVELOPMENT LEARNING MATERIALS          *****
*****
*****
*****
TITLE   DST  ENGLISH,@LANG
        DST  ENGWRD,@WLANG          FOR ROM TEXT FORMATTER
        ST   0,@LANGNM
MAIN    DST  >900,@FAC          LOAD IN SMALL CHARACTER SET
        CALL CHR2
*
                MOVE COPYRIGHT SYMBOL
MOVE 32 FROM ROM(#AURL) TO RAM(>0ADB)
MOVE 8 FROM RAM(>B50) TO RAM(>930)
ST 9,@WRDLST          START WITH WORD LIST 9
ST 9,@SAVLST          AND SAVE IT OUT FOR LATER
ST 2,@RUNTIM          INITIALLY 2 MINUTES
ST 7,@SPEED           START W/ 7 ON SPEED
ST 2,@DIFFIC          START WITH 2 BLOCK/QUADRANT
ST 0,@JOYST?         SET TO NO JOYSTICK
DCLR @HITS
DCLR @MISSES
DCLR @HHITS
DST >7FFF,@LHITS
DCLR @HMISS
DST >7FFF,@LMISS
*
TITL1
CALL MOVSCR

```

```

DATA #MOVTTL
ST >FO, TABLE(1)          LETTERS/NUMBERS
MOVE 10 FROM TABLE(1) TO TABLE(1)+1
*                             PATTERNS
MOVE 10 FROM ROM(TTLCOL) TO TABLE(22)
DCLR RAM(BEGAD)
DCLR RAM(AIDAD)

```

```

*****
***** FIRST SCREEN - DLM WITH GAME NAME *****
*****

```

```

ALL : :
BACK BLACK
CALL FORDLM          FORMAT DLM, SQUARES AND MEN
*                   FORMAT TRIANGLES
FMT XPT=14, YPT=4, '>CD, >CB, >CC', 29<, '>CA, >BB, >CB';
29<, '>CF, >C9, >CE', 26<, 3('>C5, >C0, >C4'), 23<;
3('>C2, >B9, >C3'), 23<, 3('>C7, >C1, >C6'), 20<;
5('>FD, >FB, >FC'), 17<, 5('>FA, >B0, >FB'), 17<;
5('>FF, >F9, >FE'), 20<, 4^;
'>C5, >C0, >C4, >CD, >CB, >CC, >C5, >C0, >C4';
23<; '>C2, >B9, >C3, >CA, >BB, >CB, >C2, >B9, >C3', 23<;
'>C7, >C1, >C6, >CF, >C9, >CE, >C7, >C1, >C6', 21<, 1^
CALL TEXT
DATA WARCSK
DATA WNAME, THRU

```

```

*****
***** NEXT SCREEN - COPYRIGHT SECTION *****
*****

```

```

CLR @TIMER
$REPEAT
SCAN1  CALL CTLO
        BR FAST1
        BR CPYSC
FAST1  $UNTIL @TIMER .HE. 180
CPYSC  ALL : :
        BACK BLACK
        CALL FORDLM  FORMAT DLM, SQUARES AND MEN
        FMT YPT=6, XPT=13, '>26', ': 1982: ';
        23<, 1^, ': DLM, INC. AND: ';
        17<, 1^, ': TEXAS INSTRUMENTS: ';
        17<, 1^, ': INCORPORATED: ';
        23<, 3^, ': AUTHORS: ';
        16<, 1^, ': J. CHAFFIN, B. THOMPSON: ';
        14<, 1^, ': AND B. MAXWELL: ';
        15<, 1^, ': UNIVERSITY OF KANSAS: '

```

```

CLR @TIMER
$REPEAT
SCAN2  CALL CTLO
        BR FAST2
        BR PRAKY
FAST2  $UNTIL @TIMER .HE. 180

```

```

*****
***** NEXT SCREEN - PRESS ANY KEY TO PLAY *****
*****

```

```

PRAKY  ALL : :
        BACK BLACK
        DST #GAMOPT, RAM(AIDAD)
        DST #TITL1, RAM(BEGAD)
        CALL FORDLM

```

```

*****
***** FORMAT THE BOXES AROUND SCREEN 3 *****
FMT YPT=7, XPT=3, 15">BB", 9^, 24">BB", 14">BB";
8<, 23^, 25">B0", 7<, 11^, 10">B0", 20<, 14^, 10">B0";

```

```
13^, 13<, 20^>B0', 11<, 9^, 21^>B0';
15^, 13<, 17^>B9', 5^, 15<, 17^>B9', 18^, 15<, 6">B9";
17^, 16<, 6">B9"
```

```
CALL TEXT
DATA WDUMY1
DATA WPRAKY
DATA WDUMY2
DATA WTOPLY
DATA WAIDOO, THRU
```

```
SCAN3 CALL CTLO
BR SCAN3
BR WRDRDR
```

```
*****
* TITLE COLORS *
```

```
*****
TTLCOL DATA >70, >82, >80, >20, >F0, >F0, >F0, >F0, >F0, >70
```

```
***** FORMAT DLM, SQUARES AND MEN *****
FORDLM
```

```
FMT XPT=8, YPT=0, '^>D0, >D1, >D2, >D0, >D8, >E5, >D2, >20';
'^>DE, >E2, >20', 2('^>EB, >E9, >20, >F0, >F1'), 11<;
'^>D3, >D4, >D5, >D3, >D9, >E6, >DB, >DC, >E0, >E3';
1<, 2('^>EA, >EB, >EF, >F2, >F3'), 11<, '^>D3, >D6, >D7';
'^>D3, >D6, >DA, >E7, >DD, >E1, >E4, >20';
2('^>EC, >ED, >EE, >F4, >F5')
```

```
* SQUARES
FMT XPT=0, YPT=2, 11(">B9, >20"), 1<, 1^, 11(">B8, >20");
">B8", 1^, 2<, 2("^>B8, >20'), '^>B8', 26<, 3("^>B9, >20'), 21<;
'^>B9'
FMT XPT=30, YPT=1, 11(">B8, >20"), ">B8", 2^;
1<, 11(">B9, >20")
RTN
```

```
*****
* CHANGE GAME OPTION SCREEN *
*****
```

```
GAMOPT ALL :
```

```
BACK BLACK
DCLR RAM(AIDAD)
ST @SPEED, @R1
ST @WRDLST, @R1+1
ST @RUNTIM, @R2
ST @DIFFIC, @R2+1
```

```
DISPLY FMT XPT=29, YPT=7, BIAS=>30, '@R1', 31<, 1^;
'^@R1+1', 31<, 1^, '@R2', 31<, 1^, '@R2+1'
```

```
CALL TEXT
DATA WNAME1
DATA WGCO
DATA WSPEED
DATA WPRBRN
DATA WRUNTM
DATA WDIFFI
DATA WEGCTL
DATA WPNOD
DATA WYWTC
DATA THRU
```

```
LOPB DST #>1402, YPT
CALL MENKEY
DATA : 5:
DST >071D, YPT
DEC @KEY
CASE @KEY
```



```

BR SKLV0
BR PBRNG1
BR RNTM2
BR JYSTCL
BR EXCN3
SKLV0 CALL MENKEY
DATA : 9:
ST @KEY, @SPEED
BR LOPB
PBRNG1 A 2, @YPT
* ST >FF, @FLAG
PBR1A CALL MENKEY
DATA : 9:
ST @KEY, @WRDLST
ST @KEY, @SAVLST
BR LOPB
RNTM2 A 4, @YPT
CALL MENKEY
DATA : 5:
ST @KEY, @RUNTIM
BR LOPB
JYSTCL A 6, @YPT
CALL MENKEY
DATA : 3:
ST @KEY, @DIFFIC
BR LOPB
EXCN3 CALL MOVSCR
DATA #MOVTTL
ST >FO, TABLE(1)
MOVE 10 FROM TABLE(1) TO TABLE(1)+1
MOVE 10 FROM ROM(TTLCOL) TO TABLE(22)
BR PRAKY

```

```

*
*           W O R D   R A D A R
*
*

```

```

WRDRDR MOVE 8 FROM ROM(#VDPBIT) TO VDP(0)          PUT US IN BIT MAP MODE
BACK >1          BLACK BACKGROUND
ST >A0, @KEEPR1
DST >1100, @FAC          GETS ALL THE CHARACTER
CALL >18          SETS FOR ALL THREE PDT'S
DST >0900, @FAC
CALL >18
DST >0100, @FAC
CALL >18

```

```

*           L O A D   P A T T E R N S   N E E D E D

```

```

MOVE 56 FROM ROM(#TOP1/3) TO RAM(>0000)          *
MOVE 8 FROM ROM(#TOP1/3+56) TO RAM(>0040)        *
MOVE 56 FROM ROM(#MID1/3) TO RAM(>0800)          *
MOVE 8 FROM ROM(#MID1/3+56) TO RAM(>0840)        *
MOVF 56 FROM ROM(#BOT1/3) TO RAM(>1000)          *
MOVE 136 FROM ROM(#FIN1/3) TO RAM(>1040)         *

```

```

*

```

```

MOVE 32 FROM RAM(>158) TO RAM(>6C0)          +, -.
MOVE 32 FROM ROM(#ZEROES) TO RAM(>158)
MOVE 32 FROM RAM(>188) TO RAM(>6E0)          1234
MOVE 32 FROM ROM(#ZEROES) TO RAM(>188)
MOVE 32 FROM RAM(>258) TO RAM(>700)          KLMN
MOVE 32 FROM ROM(#ZEROES) TO RAM(>258)
MOVE 32 FROM RAM(>288) TO RAM(>720)          QRST
MOVE 32 FROM ROM(#ZEROES) TO RAM(>288)

```

```

*
MOVE 32 FROM RAM(>958) TO RAM(>EC0)      +, -.
MOVE 32 FROM ROM(#ZERDES) TO RAM(>958)
MOVE 32 FROM RAM(>988) TO RAM(>EE0)      1234
MOVE 32 FROM ROM(#ZERDES) TO RAM(>988)
MOVE 32 FROM RAM(>A58) TO RAM(>F00)      KLMN
MOVE 32 FROM ROM(#ZERDES) TO RAM(>A58)
MOVE 32 FROM RAM(>A88) TO RAM(>F20)      QRST
MOVE 32 FROM ROM(#ZERDES) TO RAM(>A88)

*
MOVE 32 FROM RAM(>1158) TO RAM(>16C0)    +, -.
MOVE 32 FROM ROM(#ZERDES) TO RAM(>1158)
*
MOVE 32 FROM RAM(>1188) TO RAM(>16E0)    1234
*
MOVE 32 FROM ROM(#ZERDES) TO RAM(>1188)
MOVE 32 FROM RAM(>1258) TO RAM(>1700)    KLMN
MOVE 32 FROM ROM(#ZERDES) TO RAM(>1258)
MOVE 32 FROM RAM(>1288) TO RAM(>1720)    QRST
MOVE 32 FROM ROM(#ZERDES) TO RAM(>1288)

*
MOVE 32 FROM ROM(#AURL) TO RAM(>02DB)    FOR FOREIGN WORDS
MOVE 32 FROM ROM(#AURL) TO RAM(>0ADB)
MOVE 32 FROM ROM(#AURL) TO RAM(>12DB)

*
MOVE 256 FROM ROM(#BOTSIT) TO RAM(>1A00) FORMAT BOTTOM OF SCREEN
CALL TEXT
DATA WHITS2, WMISS2, THRU

*
      MOVE THE COLORS NEEDED
DST 7, @R0          COUNTER
DCLR @R1           AND INCREMENT
COL1 MOVE 8 FROM ROM(#GT) TO RAM(>2000(R1))
DADD 8, @R1        POINT TO NEXT CHAR
DDEC @R0
BR COL1
MOVE 8 FROM ROM(#WT) TO RAM(>2040)

*
DST 7, @R0          COUNTER
DCLR @R1
COL2 MOVE 8 FROM ROM(#GT) TO RAM(>2800(R1))
DADD 8, @R1
DDEC @R0
BR COL2
MOVE 8 FROM ROM(#WT) TO RAM(>2840)

*
DST 7, @R0
DCLR @R1
COL3 MOVE 8 FROM ROM(#GW) TO RAM(>3000(R1))
DADD 0, @R1
DDEC @R0
BR COL3

*
DST 17, @R0
DCLR @R1
COLF MOVE 8 FROM ROM(#WT) TO RAM(>3040(R1))
DADD 8, @R1
DDEC @R0
BR COLF

*
      GET THE COLORS FOR THE CHARACTER SETS
DST 64, @R0
DCLR @R1

```

```

CS1  MOVE 8 FROM ROM(#WT) TO RAM(>2100(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS1
      DST 16,@R0          NEED COLORS FOR MOVED CHAR SET
      DCLR @R1
CS11  MOVE 8 FROM ROM(#WT) TO RAM(>26C0(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS11
*
      DST 64,@R0
      DCLR @R1
CS2  MOVE 8 FROM ROM(#WT) TO RAM(>2900(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS2
      DST 16,@R0
      DCLR @R1
CS22  MOVE 8 FROM ROM(#WT) TO RAM(>2EC0(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS22
*
      DST 64,@R0
      DCLR @R1
CS3  MOVE 8 FROM ROM(#WT) TO RAM(>3100(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS3
      DST 16,@R0
      DCLR @R1
CS33  MOVE 8 FROM ROM(#WT) TO RAM(>36C0(R1))
      DADD 8,@R1
      DDEC @R0
      BR CS33
*****
*   G O T O   R O M   C O D E   *
*****
GOROM XML >70          GOTO ROM CODE
*****
* RETURN POINT FROM ROM CODE *
*****
***  ST >D0,RAM(>300)          DISABLE ALL SPRITES
*
RECSR ALL : :
      MOVE 8 FROM ROM(#VDPNOR) TO VDP(0)
      DST >700,@FAC
      CALL >1B
      BACK BLACK
      MOVE 32 FROM ROM(#AUML) TO RAM(>0AD8)
*
*           GET THE COLORS FOR THE CHARACTER SETS
      DST 2,@R0
      DCLR @R1
RCS1  MOVE 8 FROM ROM(#WT) TO RAM(>380(R1))
      DADD 8,@R1
      DDEC @R0
      BR RCS1
*
      DST #CAMOPT,RAM(AIDAD)

```

```

DST #TITL1, RAM(BEGAD)
$IF @HITS .DL. @LHITS THEN
  DST @HITS, @LHITS
$END IF-THEN
$IF @HITS .DH. @HHITS THEN
  DST @HITS, @HHITS
$END IF-THEN
$IF @MISSES .DL. @LMISS THEN
  DST @MISSES, @LMISS
$END IF-THEN
$IF @MISSES .DH. @HMISS THEN
  DST @MISSES, @HMISS
$END IF-THEN
CALL TEXT
DATA WNAME2
DATA WRCYSC
DATA WCURNT
DATA WLOW, WHIGH
DATA WHITS1, WMISS1, THRU
CALL SCORE
CLR @KEYBRD
CALL TEXT
DATA WPAKTC, THRU
LOOPF CALL CTLO
BR LOOPF
BR WRDRDR

*=====
RYXDAT DATA >0B, >0A, >0E, >0A, >0B, >12, >0E, >12, >0B, >1A, >0E, >1A
*
*** DISPLAY SCORES FOR RECORD YOUR SCORE SCREEN
*
SCORE
ST >AC, @R3 ADDRESS OF HITS
DCLR @R4
$REPEAT
  MOVE 2 FROM ROM(RYXDAT(R4)) TO YPT
  DINCT @R4
  DCLR @R5
  DST *R0, @R5+2
  DDIV 100, @R5
  DIV 10, @R5+2
  ST >0B, @MOV TAB
  $REPEAT
    $IF @MOV TAB .EQ. >0B
      $IF *MOV TAB .EQ. 0 THEN
        FMT '>20'
      $ELSE
        FMT BIAS=>30, '*MOV TAB'
      $END IF-ELSE
    $ELSE
      FMT BIAS=>30, '*MOV TAB'
    $END
  INC @MOV TAB
  $UNTIL @MOV TAB .EQ. >0E
  INCT @R3
  $UNTIL @R3 .EQ. >B8 >B8=ADDRESS OF NMJOY?
RTN

*****
* SUBROUTINE: MOVSCR *
* ***** *
* * CALL MOVSCR *
* * DATA label *

```

```

*      *   WHERE label IS THE ADDRESS OF THE      *      *
*      *   BEGINNING OF THE DATA TABLE TO USE.  *      *
*      *   *****                               *      *
*      *   * VARIABLES USED IN THE ROUTINE:        *      *
*      *   @MOVTAB--2-BYTE GROM ADDRESS OF DATA TABLE. *      *
*      *   @BYTE--2-BYTE COUNT OF NUMBER OF BYTES TO MOVE. *      *
*      *   @TAB1--2-BYTE GROM ADDRESS OF PATTERN. *      *
*      *   @TAB2--2-BYTE VDP RAM ADDRESS OF PATTERN. *      *
*      *   * NOTE: A ZERO TERMINATES THE SUBROUTINE. *      *
*      *   *****                               *      *
*      *   THIS ROUTINE MOVES DATA FROM GROM INTO VDP RAM *      *
*      *   PATTERN GENERATOR AREA. THE ROUTINE IS DATA TABLE *      *
*      *   DRIVEN WITH FOUR (4) BYTES PER MOVE STATEMENT. THE *      *
*      *   BEGINNING ADDRESS OF THE DATA TABLE SHOULD BE STORED*      *
*      *   IN @MOVDAT BEFORE ENTERING THE ROUTINE AT MOVDTA. *      *
*      *   THE FIRST TWO BYTES OF THE DATA TABLE ARE THE *      *
*      *   GROM ADDRESS OF THE DATA TO BE MOVED. THE SECOND *      *
*      *   BYTE IS THE NUMBER OF BYTES TO MOVE (THIS MEANS THE *      *
*      *   NUMBER OF BYTES MUST BE LESS THAN 256. IF YOU MUST *      *
*      *   MOVE MORE THAN THIS NUMBER OF BYTES, THE ROUTINE CAN*      *
*      *   BE MODIFIED TO ALLOW FOR THIS CHANGE.) THE FINAL *      *
*      *   BYTE IS THE PATTERN NUMBER IN THE PATTERN GENERATOR *      *
*      *   WHERE THE DATA IS TO BE MOVED. THE ROUTINE IS *      *
*      *   TERMINATED WHEN A ZERO (0) IS ENCOUNTERED IN THE *      *
*      *   FIRST BYTE OF A FOUR-BYTE DATA BLOCK. *      *
*      *   *****                               *      *
*      *   *****                               *      *
*      *   DATA TABLES FOR LOADING VDP PATTERN GENERATOR *      *
*      *   *****                               *      *
***   TITLE SEQUENCE DATA
*
MOVITL DATA #PAT01, 8, >B0, #PAT00, 16, >B8, #PAT27, 32, >C0
DATA #PAT10, 32, >C4, #PAT27, 32, >C8, #PAT10, 32, >CC
DATA #PAT02, 112, >D0, #PAT13, 48, >DF, #PAT2B, 24, >E5
DATA #PAT11, 8, >DE
DATA #PAT19, 112, >E8, #PAT27, 32, >F8, #PAT10, 32, >FC
DATA #LNGPT, 16, >5B
DATA >00
*
*
MOVSCR  FETCH @MOVTAB
        FETCH @MOVTAB+1
MOVDTA  DCLR @BYTE          *NUMBER OF BYTES TO BE MOVED
        MOVE 2 FROM ROM(@MOVTAB) TO @TAB1      *GROM ADDRESS
        *IF @TAB1 .EQ. 0 GOTO SUBEXT          *END OF LIST
        DINCT @MOVTAB      *SKIP PAST GROM ADDRESS
        MOVE 2 FROM ROM(@MOVTAB) TO @TAB2      *NUMBER OF BYTES
*      *   ... TO MOVE AND VDP PATTERN GENERATOR NUMBER.
        ST @TAB2, @BYTE+1      *NUMBER OF BYTES TO MOVE
        CLR @TAB2              *CLEAR FIRST BYTE FOR MULTIPLY
        DSSL @TAB2, 3          *MULTIPLY BY 8 BYTES PER PATTERN
        DADD >800, @TAB2      *BIAS TO ADD FOR PATTERN GENERATOR
*      *   MOVE THE DATA FROM GROM TO RAM
MOVBYT  MOVE @BYTE FROM ROM(@TAB1) TO RAM(@TAB2)
        DINCT @MOVTAB      *NEXT GROM ADDRESS
        BR MOVDTA          *UNCONDITIONAL BRANCH
SUBEXT  RTN
*****

```

```

* SUBROUTINE:  MENKEY
*
* *****
* * CALL MENKEY
* * DATA x
* *
* * x - LIMIT
* *
* *****
*
* * VARIABLES SET UP BEFORE ENTERING ROUTINE.
* * YPT - Y POINTER OF INPUT
* * XPT - X POINTER OF INPUT
*
* * VARIABLES RETURNED FROM THIS ROUTINE.
* * @KEY - 1 BYTE DECIMAL VALUE OF INPUT
*
*****
* THIS INPUT ROUTINES WORKS MAINLY WITH MENU AND
* INFORMATION ONLY SCREENS. THE ABOVE MENTIONED DATA
* STATEMENT MUST BE AFTER THE CALL TO THIS ROUTINE.
* THE INPUT VALUE IS RETURNED IN KEY.
*****
MENKEY
    FETCH @LIMIT
    CLR @TIMER
    ST :_,@R1
MN1  $IF @TIMER .HE. 30 THEN      OVER 1/2 SEC
        CLR @TIMER              CLEAR TIMER
        EX @R1,CD
    $END IF-THEN
    CALL CTLO
    BR MN1                      KEEP SCANNING TIL KEY PUSHED
    $IF @LIMIT .EQ. :0: THEN
        CASE @LANGNM
        BR ENGCK                ENGLISH KEY SCAN
        BR FRNCK                FRENCH KEY SCAN
        BR G/DCK                GERMAN KEY SCAN
        BR G/DCK                DUTCH KEY SCAN
        BR ITLCK                ITALIAN KEY SCAN
ENGCK $IF @KEY .NE. :Y: THEN
        $IF @KEY .NE. :N: THEN
            $GOTO MN1
        $END IF-THEN
    $END IF-THEN
    $GOTO LANGOK
FRNCK $IF @KEY .NE. :O: THEN
        $IF @KEY .NE. :N: THEN
            $GOTO MN1
        $END IF-THEN
    $END IF-THEN
    $GOTO LANGOK
G/DCK $IF @KEY .NE. :J: THEN
        $IF @KEY .NE. :N: THEN
            $GOTO MN1
        $END IF-THEN
    $END IF-THEN
    $GOTO LANGOK
ITLCK $IF @KEY .NE. :S: THEN
        $IF @KEY .NE. :N: THEN
            $GOTO MN1
        $END IF-THEN
    $END IF-THEN

```

```

LANGOK  ST @KEY, CB
        RTN
        $END IF-THEN
        $IF @KEY .H. @LIMIT GOTO MN1
        $IF @KEY .L. :1: GOTO MN1
        ST @KEY, CB
        S  :0:, @KEY
MNRTN1 RTN
*****
*CONTRL*          SCAN FOR CONTROL KEYS          *
*****
CONTRL $IF @JOYST? .EQ. 1 THEN
        EX @KEYBRD, @NMJY?
        $END IF-THEN
CTLO   SCAN
        BR   CTLRTN          RETURN IF NO INPUT
        $IF CB .EQ. :_: THEN
        EX  CB, @R1
        $END IF-THEN

*       DO WHAT IS INDICATED BY CONTROL KEYS

CTL1   $IF @KEY .EQ. BEGIN THEN
        $IF RAM(BEGAD) .NOT. .DEQ. 0 THEN
        DST RAM(BEGAD), @MOV TAB
        BR   CTL1A
        $END IF-THEN
        $GOTO CTL1B
        $END IF-THEN
        $IF @KEY .EQ. AID THEN
        $IF RAM(AIDAD) .NOT. .DEQ. 0 THEN
        DST RAM(AIDAD), @MOV TAB
        ST  >B0, @SUBSTK
        DST @MOV TAB, *SUBSTK
        RTN
        $END IF-THEN
        $GOTO CTL1B
        $END IF-THEN
CTL1A  DINCT *SUBSTK          ALTER RETURN
CTL1B  DINCT *SUBSTK          ALTER RETURN
CTLRTN RTN
PAGE
*****
*CBADD *          DETERMINE CHARACTER BUFFER ADDRESS *
*****
*
CBADD  ST   YPT, @CB/ADD      (YPT*32) + XPT
        M   32, @CB/ADD
        A   XPT, @CB/ADD+1
        RTN
*****
*TEXT *          MAIN TEXT FORMATTER SUBROUTINE *
*****
*
TEXT   CLR @FMT              RESET FMT
TEXT10 FETCH @FMT1          GET FORMAT CODE
        $IF @FMT1 .EQ. THRU GOTO TEXTR
        $IF @FMT1 .L. >EO THEN
        DSLL @FMT, 1          WORD # TIMES 2
        DST @LANG, @CNTFMT    BEGINNING OF WORD ADDRESS LIST
        DADD @FMT, @CNTFMT    ADD OFFSET INTO THE LIST
        MOVE 4 FROM ROM(@CNTFMT) TO @FMT *WORD ADDRESSES
        DSUB @FMT, @LENGTH    CALCULATE WORD LENGTH

```

```

MOVE 2 FROM ROM(@FMT) TO @CB/ADD
$IF @CB/ADD .DEQ. 0 GOTO TEXT
DDECT @LENGTH
DINCT @FMT
MOVE @LENGTH FROM ROM(@FMT) TO RAM(@CB/ADD) *GET WORD
$END IF-THEN
$GOTO TEXT

```

```

TEXTR RTN *THRU --RTN

```

```

*****
*WORDS * THESE ARE USED TO DETERMINE WORD LENGTH *
*****

```

```

ENGLSH

```

```

DATA #ENG0 ARCADEMIC SKILLBUILDER
DATA #ENG1 ALLIGATOR MIX
DATA #ENG2 DUMMY LINE (MULTI-LING)
DATA #ENG3 PRESS ANY KEY
DATA #ENG4 DUMMY LINE (MULTI-LING)
DATA #ENG5 TO PLAY
DATA #ENG6 AID = GAME OPTIONS
DATA #ENG7 ALLIGATOR MIX
DATA #ENG8 GAME CONTROL OPTIONS
DATA #ENG9 SKILL LEVEL 1-9
DATA #ENG10 PROBLEM RANGE 3,6,9
DATA #ENG11 RUN TIME (MIN) 1-5
DATA #ENG12 JOYSTICK CONTROL Y/N
DATA #ENG13 EXIT GAME CONTROL
DATA #ENG14 PRESS NUMBER OF OPTION
DATA #ENG15 YOU WISH TO CHANGE
DATA #ENG16 HITS (GAME SCREEN)
DATA #ENG17 MISSES (GAME SCREEN)
DATA #ENG18 ALLIGATOR MIX
DATA #ENG19 RECORD YOUR SCORE
DATA #ENG20 CURRENT
DATA #ENG21 LOW
DATA #ENG22 HIGH
DATA #ENG23 HITS
DATA #ENG24 MISSES
DATA #ENG25 PRESS ANY KEY TO CONTINUE
DATA #ENG26 ENERGY
DATA #ENG27 HITS
DATA #ENG28 MISSES
DATA #ENG29 SCANNER
DATA #ENGEOL * END OF LIST MARKER

```

```

*****
* ENGLISH WORD LIST *
*****

```

```

*
ENG0 DATA #>01E5, : ARCADEMIC SKILLBUILDER:
ENG1 DATA #>02A9, : WORD RADAR :
ENG2 DATA #>0000, :
ENG3 DATA #>01A9, : PRESS ANY KEY:
ENG4 DATA #>0000, :
ENG5 DATA #>01EC, : TO PLAY:
ENG6 DATA #>02E7, : AID = GAME OPTIONS:
ENG7 DATA #>0049, : WORD RADAR :
ENG8 DATA #>0086, : GAME CONTROL OPTIONS:
ENG9 DATA #>00E2, : 1 SPEED 1-9 :
ENG10 DATA #>0122, : 2 WORD LIST 1-9 :
ENG11 DATA #>0162, : 3 RUN TIME (MIN) 1-5:
ENG12 DATA #>01A2, : 4 DIFFICULTY 1-3 :
ENG13 DATA #>01E2, : 5 EXIT GAME CONTROL:

```


ENG14 DATA #>0285, :PRESS NUMBER OF OPTION:
 ENG15 DATA #>02C7, :YOU WISH TO CHANGE:
 ENG16 DATA #>02C2, :HITS:
 ENG17 DATA #>02D5, :MISSES:
 ENG18 DATA #>0049, : WORD RADAR :
 ENG19 DATA #>00A7, :RECORD YOUR SCORE:
 ENG20 DATA #>0108, :CURRENT:
 ENG21 DATA #>0112, :LOW:
 ENG22 DATA #>0119, :HIGH:
 ENG23 DATA #>0161, :HITS:
 ENG24 DATA #>01C1, :MISSES:
 ENG25 DATA #>0284, :PRESS ANY KEY TO CONTINUE:
 ENG26 DATA #>1A6D, :E, :>E3, :E, :>E5, :GY:
 ENG27 DATA #>1A85, :HI, :>E7, >E6
 ENG28 DATA #>1A95, >E2, : I, :>E6, >E6, : E, :>E6
 ENG29 DATA #>1AEC, >E6, : GA, :>E3, >E3, : E, :>E5, : :
 ENGEOL

*
 ENGWRD EQU \$ ENGLISH WORD LIST START

DATA : , ,>E2, : Y : MY
 DATA : I, :>E6, : : IS
 DATA : WE : WE
 DATA : O, :>E3, : : ON
 DATA : I, :>E3, : : IN
 DATA : , ,>E7, : O : TO
 DATA : HE : HE
 DATA : GO : GO
 DATA : DO : DO
 DATA : , ,>E2, : E : MC
 DATA : BE : BE
 DATA : A, :>E7, : : AT
 DATA : I, :>E7, : : IT
 DATA : OF : OF
 DATA : , ,>E3, : O : NO
 DATA : A, :>E2, : : AM
 DATA : FO, :>E5, : : FOR
 DATA : BOY : BOY
 DATA : CA, :>E3, : : CAN
 DATA : UP : UP
 DATA : CA, :>E7, : : CAT
 DATA : , ,>E7, : HE : THE
 DATA : , ,>E3, : O, :>E7, : : NOT
 DATA : O, :>E5, : : OR

^
 !
 *----- S E T O N E

DATA : DOG : 01
 DATA : A, :>E3, : D : 02
 DATA : , ,>E2, : A, :>E3, : : 03
 DATA : A, :>E3, : Y : 04
 DATA : HAD : 05
 DATA : WA, :>E6, : : 06
 DATA : HI, :>E6, : : 07
 DATA : BUY : 08
 DATA : , ,>E6, : EE : 09
 DATA : HOW : 10
 DATA : HI, :>E2, : : 11
 DATA : OU, :>E5, : : 12
 DATA : GE, :>E7, : : 13
 DATA : A, :>E5, : E : 14
 DATA : HE, :>E5, : : 15
 DATA : OU, :>E7, : : 16
 DATA : , ,>E6, : AW : 17
 DATA : DID : 18

DATA : O : >E1 : D : 19
 DATA : BU : >E7 : : 20
 DATA : : >E3 : EW : 21
 DATA : : >E6 : HE : 22
 DATA : WHO : 23
 DATA : : >E3 : A : >E3 : : 24 RAN

*----- S E T T W O RUN

DATA : : >E5 : U : >E3 : : 01
 DATA : O : >E3 : E : 02
 DATA : : >E5 : ED : 03
 DATA : : >E7 : WO : 04
 DATA : PU : >E7 : : 05
 DATA : BIG : 06
 DATA : EA : >E7 : : 07
 DATA : A : >E6 : >E0 : : 08
 DATA : : >E6 : AY : 09
 DATA : HA : >E6 : : 10
 DATA : DAY : 11
 DATA : : >E2 : AY : 12
 DATA : U : >E6 : E : 13
 DATA : : >E1 : E : >E7 : : 14
 DATA : FU : >E3 : : 15
 DATA : WHY : 16
 DATA : BOX : 17
 DATA : FA : >E5 : : 18
 DATA : CA : >E5 : : 19
 DATA : : >E7 : OO : 20
 DATA : : >E7 : >E5 : Y : 21
 DATA : : >E6 : I : >E7 : : 22
 DATA : WAY : 23
 DATA : YOU : 24

*----- S E T T H R E E

DATA : : >E1 : I : >E3 : E : 01
 DATA : WI : >E1 : >E1 : : 02
 DATA : D : >E5 : AW : 03
 DATA : HAVE : 04
 DATA : : >E7 : HI : >E6 : : 05
 DATA : : >E3 : A : >E2 : E : 06
 DATA : BOO : >E0 : : 07
 DATA : GI : >E5 : >E1 : : 08
 DATA : : >E1 : I : >E0 : E : 09
 DATA : HO : >E2 : E : 10
 DATA : : >E7 : HEY : 11
 DATA : F : >E5 : O : >E2 : : 12
 DATA : : >E5 : EAD : 13
 DATA : WHE : >E3 : : 14
 DATA : WO : >E5 : D : 15
 DATA : : >E6 : O : >E2 : E : 16
 DATA : : >E7 : HE : >E2 : : 17
 DATA : YOU : >E5 : : 18
 DATA : WI : >E7 : H : 19
 DATA : GOOD : 20
 DATA : WHA : >E7 : : 21
 DATA : : >E7 : HA : >E7 : : 22
 DATA : BA : >E1 : >E1 : : 23
 DATA : GA : >E2 : E : 24

*----- S E T F O U R

DATA : P : >E1 : AY : 01
 DATA : I : >E3 : >E7 : O : 02
 DATA : CA : >E2 : E : 03
 DATA : WO : >E5 : >E0 : : 04
 DATA : GIVE : 05

DATA :	:	>E7, >E5, : EE :	06	
DATA :	VE, :	>E5, : Y :	07	
DATA :	:	>E7, : A, : >E0, : E :		08
DATA :	DOW, :	>E3, : :	09	
DATA :	:	>E1, : O, : >E3, : G :		10
DATA :	WE, :	>E5, : E :	11	
DATA :	:	>E6, : AID :	12	
DATA :	:	>E7, : OW, : >E3, : :		13
DATA :	:	>E6, : OO, : >E9, : :		14
DATA :	:	>E1, : OO, : >E0, : :		15
DATA :	BAC, :	>E0, : :	16	
DATA :	FI, :	>E3, : D :	17	
DATA :	YEA, :	>E5, : :	18	
DATA :	:	>E1, : EF, : >E7, : :		19
DATA :	FOU, :	>E5, : :	20	
DATA :	AWAY :		21	
DATA :	WA, :	>E3, >E7, : :		22
DATA :	WI, :	>E6, : H :	23	^
DATA :	:	>E1, : A, : >E6, >E7, : :		24

*----- S E T F I V E

DATA :	:	>E6, : HOW :	01	
DATA :	BEE, :	>E3, : :	02	
DATA :	:	>E7, : A, : >E1, >E1, : :		03
DATA :	BE, :	>E6, >E7, : :		04
DATA :	JU, :	>E2, : P :	05	
DATA :	:	>E2, : UCH :	06	
DATA :	EACH :		07	
DATA :	:	>E3, : EX, : >E7, : :		08
DATA :	:	>E2, : O, : >E5, : E :		09
DATA :	BO, :	>E7, : H :	10	
DATA :	FI, :	>E6, : H :	11	
DATA :	HIGH :		12	
DATA :	:	>E0, : I, : >E3, : D :		13
DATA :	A, :	>E1, >E6, : O :		14
DATA :	CA, :	>E1, >E1, : :		15
DATA :	:	>E2, : U, : >E6, >E7, : :		16
DATA :	:	>E3, : EA, : >E5, : :		17
DATA :	FIVE :		18	
DATA :	OVE, :	>E5, : :	19	
DATA :	:	>E0, >E3, : OW :		20
DATA :	O, :	>E3, >E1, : Y :		21
DATA :	DA, :	>E5, >E0, : :		22
DATA :	:	>E2, : O, : >E6, >E7, : :		23
DATA :	HI, :	>E1, >E1, : :		24

*----- S E T S I X

DATA :	HOU, :	>E6, : E :	01	
DATA :	W, :	>E5, : I, : >E7, : E :		02
DATA :	U, :	>E3, : DE, : >E5, : :		03
DATA :	ABOU, :	>E7, : :	04	
DATA :	>E6, >E7, : O, : >E5, : Y :			05
DATA :	WOU, :	>E1, : D :	06	
DATA :	BOO, :	>E0, >E6, : :		07
DATA :	>E7, : HE, : >E5, : E :			08
DATA :	PAPE, :	>E5, : :	09	
DATA :	AF, :	>E7, : E, : >E5, : :		10
DATA :	WO, :	>E5, : D, : >E6, : :		11
DATA :	O, :	>E7, : HE, : >E5, : :		12
DATA :	>E6, >E7, : A, : >E3, : D :			13
DATA :	>E7, : HI, : >E3, >E0, : :			14
DATA :	AGAI, :	>E3, : :	15	
DATA :	WHICH :		16	
DATA :	COU, :	>E1, : D :	17	

DATA :CO,;>E1,;O,;>E5,; :	18
DATA >E7,;HI,;>E3,;G :	19
DATA :FOU,;>E3,;D :	20
DATA >E1,;U,;>E3,;CH :	21
DATA :WHE,;>E5,;E :	22
DATA :WHI,;>E7,;E :	23
DATA :B,;>E5,;I,;>E3,;G :	24

*----- S E T S E V E N

DATA >E1,;EAVE :	01
DATA >E3,;EVE,;>E5,; :	02
DATA :WHI,;>E1,;E :	03
DATA :B,;>E1,;AC,;>E0,; :	04
DATA >E5,;IGH,;>E7,; :	05
DATA >E7,;HE,;>E6,;E :	06
DATA :U,;>E3,;>E7,;I,;>E1,; :	07
DATA :WD,;>E2,;A,;>E3,; :	08
DATA >E2,;O,;>E7,;HE,;>E5	09
DATA :FA,;>E7,;HE,;>E5	10
DATA >E1,;E,;>E7,;>E7,;E,;>E5	11
DATA >E1,;I,;>E7,;>E7,;>E1,;E:	12
DATA :BEFO,;>E5,;E:	13
DATA >E6,;CHOO,;>E1	14
DATA >E6,;I,;>E6,;>E7,;E,;>E5	15
DATA :F,;>E5,;IE,;>E3,;D:	16
DATA :F,;>E1,;OWE,;>E5	17
DATA :P,;>E5,;E,;>E7,;>E7,;Y:	18
DATA :P,;>E1,;EA,;>E6,;E:	19
DATA >E6,;HOU,;>E1,;D:	20
DATA :BE,;>E7,;>E7,;E,;>E5	21
DATA >E6,;ECD,;>E3,;D:	22
DATA :CI,;>E5,;C,;>E1,;E:	23
DATA :FI,;>E5,;>E6,;>E7,; :	24

*----- S E T E I G H T

END

```

*****
*****
*****
*****      DEVELOPMENTAL LEARNING MATERIALS      *****
*****      G R O M 4      *****
*****
*****
*****

```

```

GROM 4
ORG 0

```

```

*
GR4HDR DATA >AA, 1, 1, 0, 0, 0, #PROG4, 0, 0, 0, 0, 0, 0, 0, 0

```

```

PROG4 DATA #PROG3, #START4, 10, : NEDERLANDS:
START4 ST 3, @LANGNM
DST DUTCH, @LANG
DST DUTWRD, @WLANG          FOR ROM TEXT FORMATTER
B MAIN

```

```

PROG3 DATA #PROG2, #START3, 7, : ESPANOL:
START3 ST 4, @LANGNM
DST SPNSH, @LANG
DST SPAWRD, @WLANG          FOR ROM TEXT FORMATTER
B MAIN

```

```

PROG2 DATA #PROG1, #START2, 8, : ITALIANO:
START2 ST 4, @LANGNM
DST ITALN, @LANG
DST ITAWRD, @WLANG          FOR ROM TEXT FORMATTER
B MAIN

```

```

PROG1 DATA #PROG0, #START1, 7, : DEUTSCH:
START1 ST 2, @LANGNM
DST GERMAN, @LANG
DST GERWRD, @WLANG          FOR ROM TEXT FORMATTER
B MAIN

```

```

PROG0 DATA #0, #START0, 8, : FRANCAIS:
START0 ST 1, @LANGNM
DST FRENCH, @LANG
DST FREWRD, @WLANG          FOR ROM TEXT FORMATTER
B MAIN

```

```

*****
* LANGUAGE HARD CODE ADDRESSES *
*****

```

```

DUTWRD EQU >A000
GERWRD EQU >A800

```

```

*****
* GROM 3 EQUATES FOR VARIABLES
*****

```

```

MAIN EQU >644F          ENTRY POINT FOR GROM 3
LANGNM EQU >43          1      WHICH LANGUAGE IS BEING USED *
**                      0-ENG 1-FR 2-GER 3-DUT 4-IT*
LANG EQU >44           2      LANGUAGE POINTER FOR TEXT *
WLANG EQU >68

```

```

*****
*WORDS *              THESE ARE USED TO DETERMINE WORD LENGTH *
*****

```

```

*
FRENCH
DATA #FRN0          ACADEMIC SKILLBUILDER
DATA #FRN1          COGNITIVE SAVANT
DATA #FRN2          APPUIE SUR UNE
DATA #FRN3          TOUCHIE POUR

```

DATA #FRN4	COMMENCER A
DATA #FRN5	T'AMUSER
DATA #FRN6	AID = OPTIONS DE JEU
DATA #FRN7	CROCODILE SAVANT
DATA #FRN8	OPTIONS DE JEU
DATA #FRN9	1 DEGRE DE DIFFICULTE 1-9
DATA #FRN10	2 TYPE D'EXERCISES 3,6,9
DATA #FRN11	3 DUREE (MINUTES) 1-5
DATA #FRN12	4 MANETTES? O/N
DATA #FRN13	5 FIN DE SELECTION
DATA #FRN14	TAPE LE NUMERO DE
DATA #FRN15	L'OPTION A MODIFIER
DATA #FRN16	CORRECT(GAME SCREEN)
DATA #FRN17	FAUX (GAME SCREEN)
DATA #FRN18	CROCODILE SAVANT
DATA #FRN19	NOTE TON SCORE
DATA #FRN20	SCORE
DATA #FRN21	MINI
DATA #FRN22	MAXI
DATA #FRN23	CORRECT
DATA #FRN24	FAUXES
DATA #FRN25	APPUIE SUR UNE TOUCHE
DATA #FRN26	ENERGY
DATA #FRN27	HITS
DATA #FRN28	MISSES
DATA #FRNEOL	* END OF LIST MARKER

* FRENCH WORD LIST *

*

FRN0	DATA #>01E5, : ARCADEMIC SKILLBUILDER:
FRN1	DATA #>02AB, : MOT RADAR:
FRN2	DATA #>0189, : APPUIE SUR UNE:
FRN3	DATA #>01AA, : TOUCHE POUR:
FRN4	DATA #>01CA, : COMMENCER A:
FRN5	DATA #>01EB, : T'AMUSER:
FRN6	DATA #>02E6, : AID = OPTIONS DE JEU:
FRN7	DATA #>004B, : MOT RADAR:
FRN8	DATA #>0007, : OPTIONS DE JEU:
FRN9	DATA #>00E2, : 1 VITESSE 1-9:
FRN10	DATA #>0122, : 2 LISTE MOTS 1-9:
FRN11	DATA #>0162, : 3 DUREE (MINUTES) 1-5:
FRN12	DATA #>01A2, : 4 DEGRE DE DIFFICULTE 1-3:
FRN13	DATA #>01E2, : 5 FIN DE SELECTION:
FRN14	DATA #>0287, : TAPE LE NUMERO DE:
FRN15	DATA #>02C6, : L'OPTION A MODIFIER:
FRN16	DATA #>02C0, : CORRECT:
FRN17	DATA #>02D6, : FAUX:
FRN18	DATA #>004B, : MOT RADAR.
FRN19	DATA #>00A9, : NOTE TON SCORE:
FRN20	DATA #>0108, : SCORE:
FRN21	DATA #>0112, : MINI:
FRN22	DATA #>011A, : MAXI:
FRN23	DATA #>0161, : CORRECT:
FRN24	DATA #>01C1, : FAUX:
FRN25	DATA #>0286, : APPUIE SUR UNE TOUCHE:
FRN26	DATA #>1A6D, : E, >E3, : E, >E5, : GY:
FRN27	DATA #>1A85, : BU, >E7, >E6
FRN28	DATA #>1A95, >E5, : A, >E7, : E, >E6

FRNEOL

* FRENCH WORD LIST START

DATA : : , >E2, : A : MY
 DATA : E : , >E6, : : IS
 DATA : O : , >E3, : : WE
 DATA : : , >E6, : U : , >E5, : : ON
 DATA : E : , >E3, : : IN
 DATA : AU : : TO
 DATA : I : , >E1, : : HE
 DATA : VA : : GO
 DATA : DU : : DO
 DATA : : , >E2, : E : ME
 DATA : : , >E6, : I : BE
 DATA : AUX : : AT
 DATA : CE : : IT
 DATA : DE : : OF
 DATA : : , >E3, : E : NO
 DATA : E : , >E7, : : AM
 DATA : POU : , >E5, : : FOR
 DATA : GA : , >E5, >E6, : : BOY
 DATA : PEU : , >E7, : : CAN
 DATA : HAU : , >E7, : : UP
 DATA : CHA : , >E7, : : CAT
 DATA : : , >E1, : E : , >E6, : : HE
 DATA : PA : , >E6, : : NOT
 DATA : OU : : OR

----- S E T O N E

DATA : DO : , >E7, : : DOG
 DATA : A : , >E3, : E : AND
 DATA : : , >E2, : O : , >E3, : : MAN
 DATA : : , >E7, : E : , >E1, : : ANY
 DATA : EU : , >E7, : : HAD
 DATA : E : , >E7, : E : WAS
 DATA : : , >E6, : O : , >E3, : : HTS
 DATA : BU : , >E7, : : BUY
 DATA : VUE : : SEE
 DATA : HOP : : HOW
 DATA : : , >E1, : UI : : HIM
 DATA : : , >E3, : O : , >E6, : : OUR
 DATA : JE : , >E7, : : GET
 DATA : A : , >E5, : E : ARE
 DATA : : , >E6, : E : , >E6, : : HER
 DATA : HO : , >E5, >E6, : : OUT
 DATA : : , >E6, : CIE : : SAW
 DATA : FU : , >E7, : : DID
 DATA : AGE : : OLD
 DATA : O : , >E5, : : BUT
 DATA : : , >E3, : EUF : : NEW
 DATA : E : , >E1, >E1, : E : SHE
 DATA : : , >E4, : UI : : WHO
 DATA : COU : , >E5, >E7, : : RAN

----- S E T T W O

DATA : COU : , >E5, : : RUN
 DATA : U : , >E3, : E : ONE
 DATA : : , >E5, : A : , >E7, : : RED
 DATA : DEUX : : TWO
 DATA : : , >E2, : I : , >E6, : : PUT
 DATA : G : , >E5, : O : , >E6, : : BIG
 DATA : : , >E1, : I : , >E7, : : EAT
 DATA : A : , >E5, : C : : ASK
 DATA : DI : , >E7, : : SAY
 DATA : EUE : : HAS
 DATA : JOU : , >E5, : : DAY
 DATA : : , >E2, : AI : : MAY

DATA : U: >E6: E : USE
 DATA : : >E7: E: >E1: : TEL
 DATA : FI: >E3: : FUN
 DATA : WA: >E7: >E7: : WHY
 DATA : BOU: >E7: : BOX
 DATA : : >E1: OI: >E3: : FAR
 DATA : CA: >E5: : CAR
 DATA : : >E7: O: >E7: : TOO
 DATA : : >E7: >E5: I : TRY
 DATA : : >E6: IX : SIT
 DATA : VOIE : WAY
 DATA : VOU: >E6: : YOU

*----- S E T T H R E E

DATA : : >E1: U: >E3: E : LINE
 DATA : GOU: >E7: : WILL
 DATA : : >E7: I: >E5: E : DRAW
 DATA : O: >E3: >E7: : HAVE
 DATA : CECI : THIS
 DATA : : >E3: O: >E2: >E6: : NAME
 DATA : BOC: >E0: : BOOK
 DATA : FI: >E1: >E1: E : GIRL
 DATA : AI: >E2: E : LIKE
 DATA : HO: >E7: E : HOME
 DATA : I: >E1: >E6: : THEY
 DATA : VE: >E3: U : FROM
 DATA : : >E1: UE: >E6: : READ
 DATA >E4: UA: >E3: D : WHEN
 DATA : : >E2: O: >E7: >E6: : WORD
 DATA : : >E6: OIE : SOME
 DATA : I: >E1: >E6: : THEM
 DATA : VO: >E7: >E5: E : YOUR
 DATA : AVEC : WITH
 DATA : BO: >E3: >E6: : GOOD
 DATA : : >E4: UE: >E1: : WHAT
 DATA : CE: >E1: A : THAT
 DATA : BA: >E1: >E1: E : BALL
 DATA : JEUX : GAME

*----- S E T F O U R

DATA : JOUE : PLAY
 DATA : DA: >E3: >E6: : INTO
 DATA : VI: >E3: >E7: : CAME
 DATA : : >E2: O: >E5: >E7: : WORK
 DATA : DO: >E3: >E6: : GIVE
 DATA : A: >E5: B: >E5: E : TREE
 DATA : : >E7: >E5: E: >E6: : VERY
 DATA : P: >E5: I: >E6: : TAKE
 DATA : BA: >E6: E : DOWN
 DATA : : >E1: O: >E3: G : LONG
 DATA : VE: >E5: >E7: : WERE
 DATA : DI: >E7: E : SAID
 DATA : VI: >E1: >E1: E : TOWN
 DATA : : >E7: O: >E7: : SOON
 DATA : VOI: >E5: : LOOK
 DATA : DO: >E6: : BACK
 DATA : FI: >E3: E : FIND
 DATA : A: >E3: >E3: EE : YEAR
 DATA : GA: >E3: >E7: : LEFT
 DATA : FOU: >E5: : FOUR
 DATA : AVA: >E1: : AWAY
 DATA : VEUX : WANT
 DATA : VEU: >E7: : WISH
 DATA : DU: >E5: E : LAST

*----- S E T F I V E

DATA : BEAU : SHOW
 DATA : E : >E7 : E : >E6 : : BEEN
 DATA : HAU : >E7 : : TALL
 DATA >E2 : IEUX : BEST
 DATA : : >E6 : AU : >E7 : : JUMP
 DATA : : >E7 >E5 : OP : MUCH
 DATA >E7 : ACHE : EACH
 DATA : VEXE : NEXT
 DATA : P : >E1 : U : >E6 : : MORE
 DATA : BOU : >E7 : : BOTH
 DATA : FI : >E1 : E : FISH
 DATA : CIE : >E1 : : HIGH
 DATA : : >E7 : YPE : KIND
 DATA : : >E2 : E : >E2 : E : ALSO
 DATA : CA : >E1 : E : CALL
 DATA : DOI : >E7 : : MUST
 DATA : P : >E5 : E : >E6 : : NEAR
 DATA : CI : >E3 >E4 : : FIVE
 DATA : PO : >E6 : E : OVER
 DATA : : >E6 : AI : >E7 : : KNOW
 DATA : : >E6 : EU : >E1 : : ONLY
 DATA : : >E3 : OI : >E5 : : DARK
 DATA : : >E7 : OU : >E6 : : MOST
 DATA : HA : >E1 >E1 : E : HILL

*----- S E T S I X

DATA : HO : >E7 : E : >E6 : : HOUSE
 DATA : EC : >E5 : I : >E7 : : WRITE
 DATA : U : >E6 : IC : >E6 : : UNDER
 DATA >E4 : UA : >E3 >E7 : : ABOUT
 DATA : CO : >E3 >E7 : E : STORY
 DATA : POU : >E1 >E6 : : WOULD
 DATA : BOUC : >E6 : : BOOKS
 DATA >E7 : HE : >E6 : E : THERE
 DATA : PAPE : >E6 : : PAPER
 DATA : AP : >E5 : E : >E6 : : AFTER
 DATA >E2 : O : >E7 : U : >E6 : : WORDS
 DATA : AU : >E7 >E5 : E : OTHER
 DATA >E6 >E7 : A : >E3 : D : STAND
 DATA : PE : >E3 >E6 : E : THINK
 DATA : GAI : >E3 >E6 : : AGAIN
 DATA >E4 : UE : >E1 >E6 : : WHICH
 DATA : COU : >E1 : E : COULD
 DATA : CO : >E1 : O : >E3 : : COLOR
 DATA : CHO : >E6 : E : THING
 DATA : FO : >E3 : DU : FOUND
 DATA >E1 : ECHE : LUNCH
 DATA >E1 : IEUX : WHERE
 DATA : B : >E1 : A : >E3 : C : WHITE
 DATA : A : >E2 : E : >E3 : E : BRING

*----- S E T S E V E N

DATA >E1 : A : >E6 >E6 : E :
 DATA >E3 : EVEU :
 DATA : COU : >E5 >E6 : :
 DATA >E3 : OI : >E5 : E :
 DATA : V : >E5 : AI : >E6 : :
 DATA : CE : >E1 >E1 : E :
 DATA : BOU : >E7 >E6 : :
 DATA : FE : >E2 >E2 : E :
 DATA >E2 : OI : >E7 : IE :
 DATA : PA : >E7 >E5 : E : >E6
 DATA >E1 : E : >E7 >E7 >E5 : E :

DATA : PE: , >E7, : I: , >E7, : E:
 DATA : AVA: , >E3, : CE:
 DATA : ECO: , >E1, : E: , >E6
 DATA >E6, : OEU: , >E5, >E6
 DATA : A: , >E2, : ICA: , >E1
 DATA : F: , >E1, : EU: , >E9, >E6
 DATA : JD: , >E1, : IE: , >E6
 DATA : P: , >E1, : AI: , >E5, : E:
 DATA : DEVOI: , >E5
 DATA : BA: , >E7, >E7, >E5, : E:
 DATA >E6, : ECO: , >E3, : D:
 DATA : CE: , >E5, : C: , >E1, : E:
 DATA : CHEF: , >E6, : :

----- S E T E I G H T

 * ITALIAN WORD LIST *

ITALN

DATA #ITL0	ARCADEMIC SKILLBUILDER
DATA #ITL1	L'ALLIGATORE
DATA #ITL2	PREMI
DATA #ITL3	QUALSIASI
DATA #ITL4	TASTO
DATA #ITL5	PER COMINCIARE
DATA #ITL6	AID = OPZIONE DI GIOCO
DATA #ITL7	L'ALLIGATURE
DATA #ITL8	OPZIONE DI GIOCO
DATA #ITL9	1 GRADO DI ABILITA' 1-9
DATA #ITL10	2 TIPO DI ESERCIZIO 3.4.9
DATA #ITL11	3 DURATA (MIN) 1-5
DATA #ITL12	4 CONTROLLO A DISTANZA Y/N
DATA #ITL13	5 FINE SELEZIONE
DATA #ITL14	PREMERE IL NUMERO
DATA #ITL15	DI OPTIONE DA CAMBIARE
DATA #ITL16	GIUSTO
DATA #ITL17	ERRORE
DATA #ITL18	L'ALLIGATORE
DATA #ITL19	MEMORIZZA IL TUO PUNTEGGIO
DATA #ITL20	ATTUALE
DATA #ITL21	MIN
DATA #ITL22	MAX
DATA #ITL23	GIUSTO
DATA #ITL24	ERRORE
DATA #ITL25	PREMI UN TASTO PER CONTINUARE
DATA #ITL26	
DATA #ITL27	
DATA #ITL28	
DATA #ITLEOL	* END OF LIST MARKER

 * ITALIAN WORD LIST *

ITL0	DATA #>01E5, : ARCADEMIC SKILLBUILDER:
ITL1	DATA #>02AA, : RADAR PAROLE:
ITL2	DATA #>01BD, : PREMI:
ITL3	DATA #>01AB, : QUALSIASI:
ITL4	DATA #>01CD, : TASTO:
ITL5	DATA #>01E9, : PER COMINCIARE:
ITL6	DATA #>02E3, : AID = SELEZIONE DEL GIOCO:
ITL7	DATA #>004A, : RADAR PAROLE:
ITL8	DATA #>008B, : OPZIONE DI GIOCO:
ITL9	DATA #>00E1, : 1 VELOCITA 1-9:

ITL10 DATA #>0121,,2 LISTA PAROLE 1-9:
 ITL11 DATA #>0161,,3 DURATA (MIN) 1-5:
 ITL12 DATA #>01A1,,4 GRADO DI DIFFICOLTA 1-3:
 ITL13 DATA #>01E1,,5 FINE SELEZIONE:
 ITL14 DATA #>0285,,PREMERE IL NUMERO:
 ITL15 DATA #>02C5,,DESIDERI CAMBIARE:
 ITL16 DATA #>02C1,,GIUSTO:
 ITL17 DATA #>02D5,,ERRATO:
 ITL18 DATA #>004A,,RADAR PAROLE:
 ITL19 DATA #>00A3,,MEMORIZZA IL TUO PUNTEGGIO:
 ITL20 DATA #>0108,,ATTUALE:
 ITL21 DATA #>0112,,MIN:
 ITL22 DATA #>011A,,MAX:
 ITL23 DATA #>0161,,GIUSTO:
 ITL24 DATA #>01C1,,ERRATO:
 ITL25 DATA #>0281,,PREMI UN TASTO PER CONTINUARE:
 ITL26 DATA #>1A6D,,E:, >E3,,E:, >E5,,GY:
 ITL27 DATA #>1A84,,GIU:, >E6, >E7,,O:
 ITL28 DATA #>1A95,,E:, >E5, >E5,,A:, >E7,,O:

ITLEQL

*

ITAWRD EQU \$ ITALIAN WORD LIST START

DATA : ,, >E2,, E : MY
 DATA : ,, >E6,, I : IS
 DATA : CI : WE
 DATA : ,, >E6,, U : ON
 DATA : I:, >E3,, : IN
 DATA : A:, >E1,, : TO
 DATA : VI : HIC
 DATA : VA : GO
 DATA : DO : DO
 DATA : ,, >E2,, I : ME
 DATA : ,, >E6,, E : BE
 DATA : I:, >E1,, : AT
 DATA : ,, >E1,, I : IT
 DATA : DI : OF
 DATA : ,, >E3,, O : NO
 DATA : FU : AM
 DATA : PE:, >E5,, : FOR
 DATA : ,, >E1,, UI : BOY
 DATA : PUD : CAN
 DATA : ,, >E6,, F : UP
 DATA : ,, >E1,, EI : CAT
 DATA : G:, >E1,, I : HE
 DATA : ,, >E3,, O:, >E3,, : NOT
 DATA : OD : OR

*----- S E T O N E

DATA : AIA : DOG
 DATA : ,, >E6,, IA : AND
 DATA : IVI : MAN
 DATA : ,, >E7, >E5,, E : ANY
 DATA : E:, >E5,, O : HAD
 DATA : EVO : WAS
 DATA : ,, >E6,, UO : HIS
 DATA : CO:, >E3,, : BUY
 DATA : FUI : SEE
 DATA : ,, >E6,, IC : HOW
 DATA : ,, >E1,, UI : HIM
 DATA : ,, >E7,, UO : OUR
 DATA : A:, >E5,, A : GET
 DATA : CO:, >E3,, : ARE
 DATA : E:, >E5,, I : HER

DATA : E, >E5, : A : OUT
 DATA : PE, >E1, : SAW
 DATA : P, >E5, : O : DID
 DATA : , >E7, : E, >E5, : OLD
 DATA : BI, >E6, : BUT
 DATA : DUO : NEW
 DATA : , >E6, >E7, : O : SHE
 DATA : , >E6, >E7, : A : WHO
 DATA : VIA : RAN

*----- S E T T W O
 DATA : , >E2, : IA : RUN
 DATA : , >E7, : UA : ONE
 DATA : , >E6, : UA : RED
 DATA : O, >E5, : A : TWO
 DATA : POI : PUT
 DATA : , >E7, : IC : BIG
 DATA : , >E5, : EO : EAT
 DATA : , >E5, : EA : ASK
 DATA : CO, >E1, : SAY
 DATA : , >E6, : U, >E1, : HAS
 DATA : , >E2, : A, >E1, : DAY
 DATA : BE, >E1, : MAY
 DATA : DE, >E1, : USE
 DATA : , >E7, : A, >E1, : TEL
 DATA : U, >E3, : O : FUN
 DATA : DUE : WHY
 DATA : , >E7, >E5, : E : BOX
 DATA : , >E6, : EI : FAR
 DATA : ZIO : CAR
 DATA : ZIA : TOO
 DATA : PA, >E5, : TRY
 DATA : , >E6, : E, >E7, : SIT
 DATA : , >E6, : O, >E1, : WAY
 DATA : BE, >E3, : YOU

*----- S E T T H R E E
 DATA : , >E5, : IGA : LINE
 DATA : VUO, >E1, : WILL
 DATA : , >E7, : I, >E5, : A : DRAW
 DATA : EBBI : HAVE
 DATA : E, >E1, >E1, : A : THIS
 DATA : , >E3, : O, >E2, : E : NAME
 DATA : , >E1, : I, >E5, : A : BOOK
 DATA : GI, >E7, : A : GIRL
 DATA : CO, >E2, : E : LIKE
 DATA : CA, >E6, : A : HOME
 DATA : E, >E6, >E6, : I : THEY
 DATA : , >E6, : EDE : FROM
 DATA : , >E1, : EGD : READ
 DATA : DOPO : WHEN
 DATA : CI, >E7, : A : WORD
 DATA : E, >E1, : GI : SOME
 DATA : O, >E7, >E7, : O : THEM
 DATA : , >E7, : UOI : YOUR
 DATA : CO, >E6, : I : WITH
 DATA : BE, >E3, : E : GOOD
 DATA : CO, >E6, : A : WHAT
 DATA : , >E7, : A, >E1, : E : THAT
 DATA : , >E6, : A, >E1, : A : BALL
 DATA : , >E7, : I, >E5, : U : GAME

*----- S E T F O U R
 DATA : , >E2, : I, >E5, : O : PLAY
 DATA : PA, >E5, : O : INTO

DATA : CA: , >E3, : E : CAME
 DATA : , , >E7, : IPO : WORK
 DATA : DA: , >E5, : E : GIVE
 DATA : , , >E3, : OVE : TREE
 DATA : , , >E7, : A: , >E5, : A : VERY
 DATA : PE: , >E9, : A : TAKE
 DATA : , , >E2, : E: , >E1, : A : DOWN
 DATA : PA: , >E1, : A : LONG
 DATA : VO: , >E1, : A : WERE
 DATA : CE: , >E1, : A : SAID
 DATA : CA: , >E5, : A : TOWN
 DATA : CA: , >E1, : A : SOON
 DATA : AU: , >E7, : O : LOOK
 DATA : , , >E2, : O: , >E7, : O : BACK
 DATA : , , >E3, : O: , >E7, : A : FIND
 DATA : A: , >E3, >E3, : O : YEAR
 DATA : , , >E2, : E: , >E6, : E : LEFT
 DATA : BEVO : FOUR
 DATA : , , >E6, : UDO : AWAY
 DATA : , , >E1, : UCE : WANT
 DATA : , , >E1, : U: , >E2, : E : WISH
 DATA : , , >E3, : U: , >E2, : E : LAST

*----- S E T F I V E

DATA : , , >E3, : E: , >E5, : O : SHOW
 DATA : GE: , >E1, : O : BEEN
 DATA : VE: , >E1, : O : TALL
 DATA : , , >E7, : E: , >E1, : O : BEST
 DATA : , , >E2, : E: , >E1, : O : JUMP
 DATA : , , >E7, : UFO : MUCH
 DATA : , , >E2, : U: , >E1, : O : EACH
 DATA : POCO : NEXT
 DATA : , , >E5, : OCO : MORE
 DATA : , , >E2, : O: , >E1, : O : BOTH
 DATA : , , >E7, : OPO : FISH
 DATA : , , >E3, : IDO : HIGH
 DATA : , , >E6, : I: , >E1, : O : KIND
 DATA : VI: , >E3, : O : ALSO
 DATA : , , >E7, : I: , >E3, : O : CALL
 DATA : PI: , >E3, : O : MUST
 DATA : CO: , >E3, : O : NEAR
 DATA : , , >E7, : O: , >E3, : O : FIVE
 DATA : , , >E6, : O: , >E3, : O : OVER
 DATA : , , >E1, : O: , >E5, : O : KNOW
 DATA : , , >E2, : A: , >E5, : E : ONLY
 DATA : PA: , >E3, : E : DARK
 DATA : VI: , >E7, : A : MOST
 DATA : , , >E2, : E: , >E7, : A : HILL

*----- S E T S I X

DATA : VE: , >E5, : DE : HOUSE
 DATA : , , >E5, : O: , >E6, >E6, : O : WRITE
 DATA : B: , >E5, : U: , >E3, : O : UNDER
 DATA : CIE: , >E1, : O : ABOUT
 DATA : >E7, : E: , >E5, >E5, : A : STORY
 DATA : >E2, : A: , >E5, : EA : WOULD
 DATA : CO: , >E1, >E1, : E : BOOKS
 DATA : >E2, : O: , >E3, >E7, : E : THERE
 DATA : U: , >E1, : IVO : PAPER
 DATA : ABE: , >E7, : E : AFTER
 DATA : ACE: , >E5, : O : WORDS
 DATA : PAD: , >E5, : E : OTHER
 DATA : >E2, : AD: , >E5, : E : STAND
 DATA : BI: , >E2, : DO : THINK

DATA >E6, : UD, : >E5, : A :	AGAIN
DATA >E3, : O, : >E3, >E3, : O :	WHICH
DATA >E6, : O, : >E3, >E3, : O :	COULD
DATA >E1, : E, : >E7, >E7, : O :	COLOR
DATA >E6, : OG, : >E3, : O :	THING
DATA : CA, : >E3, >E7, : O :	FOUND
DATA >E6, : UD, : >E3, : O :	LUNCH
DATA : BA, : >E1, >E1, : O :	WHERE
DATA : DA, : >E3, : ZO :	WHITE
DATA : P, : >E5, : EGO :	BRING

----- S E T S E V E N

DATA : DO, : >E3, >E3, : A :	LEAVE
DATA : P, : >E5, : E, : >E7, : E :	NEVER
DATA : G, : >E5, : E, : >E7, : O :	WHILE
DATA : C, : >E5, : I, : >E3, : E :	BLACK
DATA >E2, : E, : >E6, >E6, : A :	RIGHT
DATA : G, : >E5, : IDO :	THESE
DATA : P, : >E5, : ODE :	UNTIL
DATA : C, : >E5, : E, : >E7, : A :	WOMAN
DATA : GIA, : >E1, >E1, : O :	MOTHER
DATA : FAGGIO :	FATHER
DATA >E5, : OVE, : >E5, : E :	LETTER
DATA >E7, >E5, : IPPA :	LITTLE
DATA >E7, >E5, : OPPO :	BEFORE
DATA : FIG, : >E1, : IA :	SCHOOL
DATA : FOG, : >E1, : IA :	SISTER
DATA >E2, : AG, : >E1, : IO :	FRIEND
DATA >E2, : IG, : >E1, : IO :	FLOWER
DATA : VAO, : >E1, : IO :	PRETTY
DATA >E6, >E7, : AG, : >E3, : O :	PLEASE
DATA : B, : >E5, : I, : >E1, >E1, : A :	SHOULD
DATA >E7, >E5, : I, : >E1, >E1, : O :	BETTER
DATA : CHIE, : >E6, : A :	SECOND
DATA : GUG, : >E1, : IA :	CIRCLE
DATA : P, : >E5, : I, : >E2, : O :	FIRST

----- S E T E I G H T

 * SPANISH WORD LIST *

SPNSH

DATA #SPN0	ARCADEMIC SKILLBUILDER
DATA #SPN1	COCODRILO SABIO
DATA #SPN2	DUMMY LINE
DATA #SPN3	PULSE
DATA #SPN4	CUALQUIER
DATA #SPN5	TECLA
DATA #SPN6	AID = CONTROL DE OPCIONES
DATA #SPN7	COCODRILO SABIO
DATA #SPN8	CONTROL DE OPCIONES
DATA #SPN9	1 NIVEL DE ABILIDAD 1-9
DATA #SPN10	2 RANGO 3, 6, 9
DATA #SPN11	3 TIEMPO DE JUEGO 1-5
DATA #SPN12	4 CONTROLES REMOTES S/N
DATA #SPN13	5 SALIR DE OPCIONES
DATA #SPN14	PULSE EL NRO. DE OPCION
DATA #SPN15	QUE QUIERE CAMBIAR
DATA #SPN16	CIERTOS
DATA #SPN17	ERRORES
DATA #SPN18	COCODRILO SABIO
DATA #SPN19	GUARDAR SU PUNTAJE
DATA #SPN20	NORMAL
DATA #SPN21	BAJO

DATA #SPN22 ALTO
 DATA #SPN23 CIERTOS
 DATA #SPN24 ERRORES
 DATA #SPN25 PULSE UNA TECLA P/CONT
 DATA #SPN26
 DATA #SPN27
 DATA #SPN28
 DATA #SPNEQL * END OF LIST MARKER

 * SPANISH WORD LIST *

*
 SPN0 DATA #>01E5, : ARCADEMIC SKILLBUILDER:
 SPN1 DATA #>02A7, : RADAR DE PALABRAS:
 SPN2 DATA #>0, : :
 SPN3 DATA #>01AD, : PULSE:
 SPN4 DATA #>01C8, : CUALQUIER TECLA:
 SPN5 DATA #>01EB, : PARA JUGAR:
 SPN6 DATA #>02E3, : AID = CONTROL DE OPCIONES:
 SPN7 DATA #>0047, : RADAR DE PALABRAS:
 SPN8 DATA #>0086, : CONTROL DE OPCIONES:
 SPN9 DATA #>00E1, : 1 VELOCIDAD 1-9:
 SPN10 DATA #>0121, : 2 LISTA DE PALABRAS 1-9:
 SPN11 DATA #>0161, : 3 TIEMPO DE JUEGO 1-5:
 SPN12 DATA #>01A1, : 4 NIVEL DE ABILIDAD 1-3:
 SPN13 DATA #>01E1, : 5 SALIR DE OPCIONES:
 SPN14 DATA #>0284, : PULSE EL NRO. DE OPCION:
 SPN15 DATA #>02C6, : QUE QUIERE CAMBIAR:
 SPN16 DATA #>02C0, : CIERTOS:
 SPN17 DATA #>02D4, : ERRORES:
 SPN18 DATA #>0048, : RADAR DE PALABRAS:
 SPN19 DATA #>00A7, : GUARDAR SU PUNTAJE:
 SPN20 DATA #>0108, : NORMAL:
 SPN21 DATA #>0112, : BAJO:
 SPN22 DATA #>011A, : ALTO:
 SPN23 DATA #>0161, : CIERTOS:
 SPN24 DATA #>01C1, : ERRORES:
 SPN25 DATA #>0285, : PULSE UNA TECLA P/CONT:
 SPN26 DATA #>1A6D, : E, : >E3, : E, : >E5, : >Y:
 SPN27 DATA #>1A84, : CIE, : >E5, >E7, : O, : >E6
 SPN28 DATA #>1A94, : E, : >E5, >E5, : O, : >E5, : E, : >E6
 SPNEQL

*
 SPAWRD EQU \$ SPANISH WORD LIST START
 DATA : , : >E2, : I : MY
 DATA : E, : >E6, : : IS
 DATA : VE : WE
 DATA : E, : >E3, : : ON
 DATA : U, : >E3, : : IN
 DATA : , : >E7, : U : TO
 DATA : HA : HE
 DATA : , : >E6, : I : GO
 DATA : DA : DO
 DATA : , : >E2, : E : ME
 DATA : , : >E5, : E : BE
 DATA : A, : >E1, : : AT
 DATA : , : >E7, : I : IT
 DATA : A, : >E6, : : OF
 DATA : , : >E3, : O : NO
 DATA : AY : AM
 DATA : PD : FOR
 DATA : BA, : >E5, : : BOY

DATA : : , >E7, : A, : >E1, : : : CAN
 DATA : I, : >E5, : : : UP
 DATA : : , >E1, : : A : : : CAT
 DATA : E, : >E1, : : : : HE
 DATA : : , >E3, : : O, : >E6, : : : : NOT
 DATA : : , >E7, : : E : : : OR

*----- S E T O N E

DATA : DA, : >E6, : : : : DOG
 DATA : : , >E6, : : I, : >E3, : : : : AND
 DATA : VE, : >E3, : : : : MAN
 DATA : U, : >E3, : : O : : : ANY
 DATA : HAZ : : : : HAD
 DATA : : , >E6, : : E, : >E5, : : : : WAS
 DATA : HA, : >E3, : : : : HIS
 DATA : FI, : >E3, : : : : BUY
 DATA : VE, : >E5, : : : : SEE
 DATA : : , >E6, : : OY : : : HOW
 DATA : O, : >E1, : : A : : : HIM
 DATA : : , >E6, : : A, : >E1, : : : : OUR
 DATA : DAD : : : : GET
 DATA : : , >E6, : : O, : >E1, : : : : ARE
 DATA : : , >E6, : : O, : >E3, : : : : HER
 DATA : DA, : >E5, : : : : OUT
 DATA : HOZ : : : : SAW
 DATA : : , >E5, : : IO : : : DID
 DATA : PA, : >E5, : : : : ULD
 DATA : U, : >E1, : : A : : : BUT
 DATA : BO, : >E1, : : : : BOWL
 DATA : VOZ : : : : SHE
 DATA : : , >E4, : : UE : : : WHO
 DATA : VOY : : : : RAN

*----- S E T T W O

DATA : : , >E5, : : EO : : : RUN
 DATA : : , >E7, : : U, : >E1, : : : : ONE
 DATA : FEO : : : : RED
 DATA : : , >E2, : : A, : >E5, : : : : TWO
 DATA : DO, : >E6, : : : : PUT
 DATA : PO, : >E3, : : : : BIG
 DATA : U, : >E6, : : O : : : EAT
 DATA : DOY : : : : ASK
 DATA : VA, : >E3, : : : : SAY
 DATA : DIA : : : : HAS
 DATA : : , >E7, : : O, : >E6, : : : : DAY
 DATA : VA, : >E6, : : : : MAY
 DATA : PO, : >E5, : : : : USE
 DATA : : , >E7, : : EZ : : : TEL
 DATA : : , >E7, : : IA : : : FUN
 DATA : : , >E6, : : ED : : : WHY
 DATA : CA, : >E1, : : : : BOX
 DATA : BU, : >E6, : : : : FAR
 DATA : : , >E1, : : IO : : : CAR
 DATA : : , >E1, : : ID : : : TOO
 DATA : : , >E1, : : UZ : : : TRY
 DATA : CO, : >E1, : : : : SIT
 DATA : FEA : : : : WAY
 DATA : : , >E7, : : IO : : : YOU

*----- S E T T H R E E

DATA : I, : >E6, >E1, : : A : : : LINE
 DATA : : , >E5, : : OJO : : : WILL
 DATA : DA, : >E2, : : E : : : DRAW
 DATA : : , >E7, : : O, : >E2, : : A : : : HAVE
 DATA : E, : >E6, >E7, : : O : : : THIS

DATA :	:	>E3, : ADA :			NAME
DATA :	BOCA :		BOOK		
DATA :	:	>E3, : I, : >E3, : A :			GIRL
DATA :	CA, :	>E6, : A :			LIKE
DATA :	PE, :	>E5, : O :			HOME
DATA :	CA, :	>E9, : O :			THEY
DATA :	:	>E6, : O, : >E1, : O :			FROM
DATA :	GA, :	>E7, : O :			READ
DATA :	:	>E7, : UYO :			WHEN
DATA :	A, :	>E1, >E7, : O :			WORD
DATA :	:	>E6, : U, : >E2, : A :			SOME
DATA :	CA, :	>E2, : A :			THEM
DATA :	CU, :	>E3, : A :			YOUR
DATA :	DO, :	>E7, : E :			WITH
DATA :	CA, :	>E1, : A :			GOOD
DATA :	BO, :	>E1, : A :			WHAT
DATA :	CA, :	>E5, : A :			THAT
DATA :	POCO :		BALL		
DATA :	:	>E5, : A, : >E5, : O :			GAME

*----- S E T F O U R

DATA :	BIE, :	>E3, : :			PLAY
DATA :	PU, :	>E5, : O :			INTO
DATA :	PEO, :	>E3, : :			CAME
DATA :	:	>E5, : I, : >E6, : A :			WORK
DATA :	CAJA :		GIVE		
DATA :	BAJO :		TREE		
DATA :	COJO :		VERY		
DATA :	AZU, :	>E1, : :			TAKE
DATA :	A, :	>E4, : UI :			DOWN
DATA :	A, :	>E1, >E1, : I :			LONG
DATA :	PE, :	>E6, : O :			WERE
DATA :	PE, :	>E5, : A :			SAID
DATA :	:	>E1, : U, : >E3, : A :			TOWN
DATA :	:	>E6, : UYO :			SOON
DATA :	:	>E1, : I, : >E6, : O :			LOOK
DATA :	PA, :	>E6, : O :			BACK
DATA :	E, :	>E6, >E7, : E :			FIND
DATA :	:	>E2, : A, : >E6, : A :			YEAR
DATA :	:	>E1, : OCO :			LEFT
DATA :	HIJO :		FOUR		
DATA :	:	>E7, : A, : >E6, : A :			AWAY
DATA :	:	>E2, : U, : >E5, : O :			WANT
DATA :	:	>E1, : I, : >E1, : A :			WISH
DATA :	PA, :	>E5, : A :			LAST

*----- S E T F I V E

DATA :	CO, :	>E2, : O :			SHOW
DATA :	:	>E7, >E5, : E, : >E3, : :			BEEN
DATA :	:	>E1, : IGA :			TALL
DATA :	BECA :		BEST		
DATA :	:	>E2, : A, : >E1, : A :			JUMP
DATA :	PE, :	>E6, : O :			MUCH
DATA :	BA, :	>E1, : A :			EACH
DATA :	F, :	>E1, : O, : >E5, : :			NEXT
DATA :	A, :	>E2, : O, : >E5, : :			MORE
DATA :	:	>E5, : OJO :			BOTH
DATA :	PAJE :		FISH		
DATA :	BE, :	>E6, : O :			HIGH
DATA :	:	>E6, : U, : >E2, : O :			KIND
DATA :	DAGA :		ALSO		
DATA :	CA, :	>E6, : O :			CALL
DATA :	:	>E5, : A, : >E3, : A :			MUST
DATA :	POZO :		NEAR		

DATA : : >E7, : APA : FIVE
 DATA : FOCA : OVER
 DATA : PE, : >E1, : O : KNOW
 DATA : VACA : ONLY
 DATA : : >E7, : O, : >E5, : O : DARK
 DATA : : >E6, : ACO : MUST
 DATA : : >E1, : O, : >E5, : O : HILL

*----- S E T S I X

DATA : HACE, : >E5, : : HOUSE
 DATA >E2, : EJO, : >E5, : : WRITE
 DATA : GU, : >E6, : >E7, : O : UNDER
 DATA : DE, : >E6, : : DE : ABOUT
 DATA : CA, : >E5, : >E7, : A : STORY
 DATA : CA, : >E5, : >E5, : O : WOULD
 DATA >E1, : A, : >E3, : ZA : BOOKS
 DATA >E3, : UEVO : THERE
 DATA >E1, : I, : >E6, : >E7, : O : PAPER
 DATA : DICHO : AFTER
 DATA >E5, : E, : >E6, : >E7, : A : WORDS
 DATA : BUE, : >E3, : A : OTHER
 DATA >E4, : UIE, : >E3, : : STAND
 DATA >E6, : I, : >E3, : >E7, : A : THINK
 DATA : PUEDO : AGAIN
 DATA >E4, : UE, : >E6, : O : WHICH
 DATA >E7, : O, : >E5, : >E7, : A : COULD
 DATA : A, : >E3, : >E7, : E, : >E6, : : COLOR
 DATA : CHICO : THING
 DATA : BA, : >E6, : >E7, : A : FOUND
 DATA : A, : >E2, : IGO : LUNCH
 DATA : CA, : >E1, : O, : >E5, : : WHERE
 DATA >E7, : E, : >E2, : O, : >E5, : : WHITE
 DATA : P, : >E5, : I, : >E2, : O : BRING

*----- S E T S E V E N

DATA : PE, : >E5, : >E5, : O : LEAVE
 DATA : DEJA, : >E5, : : NEVER
 DATA >E3, : U, : >E3, : CA : WHILE
 DATA >E3, : EG, : >E5, : O : BLACK
 DATA : A, : >E7, : >E5, : A, : >E6, : : RIGHT
 DATA : E, : >E6, : >E7, : O, : >E6, : : THESE
 DATA : PAD, : >E5, : E : UNTIL
 DATA >E2, : UJE, : >E5, : : WOMAN
 DATA >E2, : AD, : >E5, : E : MOTHER
 DATA >E7, : >E5, : I, : >E6, : >E7, : E : FATHER
 DATA : CHI, : >E6, : >E7, : E : LETTER
 DATA : BE, : >E6, : >E7, : IA : LITTLE
 DATA : B, : >E1, : A, : >E3, : CO : BEFORE
 DATA >E6, : EGU, : >E5, : O : SCHOOL
 DATA : CUA, : >E3, : DO : SISTER
 DATA >E6, : ABA, : >E3, : A : FRIEND
 DATA >E6, : E, : >E5, : E, : >E3, : O : FLOWER
 DATA : VA, : >E6, : IJA : PRETTY
 DATA >E6, : I, : >E2, : P, : >E1, : E : PLEASE
 DATA >E7, : A, : >E5, : IFA : SHOULD
 DATA : A, : >E5, : >E2, : ADA : BETTER
 DATA >E7, : IE, : >E2, : PO : SECOND
 DATA >E3, : AVAJA : CIRCLE
 DATA : FI, : >E3, : A, : >E1, : : FIRST

*----- S E T E I G H T

 * GERMAN WORD LIST *

 GERMAN

```

DATA #GER0      ARCADEMIMC SKILLBUILDER
DATA #GER1      ALLIGATOR MIX
DATA #GER2      DUMMY LINE
DATA #GER3      IRGENDEINE
DATA #GER4      TASTE DR\CKEN
DATA #GER5      UM ANZUFANGEN
DATA #GER6      ***AID = GAME OPTIONS
DATA #GER7      ALLIGATOR MIX
DATA #GER8      SPIELWEISE
DATA #GER9      1 \BUNGSSTUFE 1-9
DATA #GER10     2 AUFGABENFELD 3,6,9
DATA #GER11     3 LAUFZEIT (MIN) 1-5
DATA #GER12     4 FERNBEDIENUNG J/N
DATA #GER13     5 KEINE INDERUNGEN MEHR
DATA #GER14     ZIFFER DER GEW\NSCHTEN
DATA #GER15     INDERUNG DR\CKEN
DATA #GER16     RICHTIG
DATA #GER17     FALSCH
DATA #GER18     ALLIGATOR MIX
DATA #GER19     PUNKTESTAND
DATA #GER20     ZULETZT
DATA #GER21     MINI
DATA #GER22     MAX
DATA #GER23     RICHTIG
DATA #GER24     FALSCH
DATA #GER25     IRGEND EINE TASTE DR\CKEN
DATA #GER26
DATA #GER27
DATA #GER28
DATA #GEREOL    * END OF LIST MARKER

```

```

*****
*                GERMAN WORD LIST                *
*****
*

```

```

GER0      DATA #>01E5, : ARCADEMIC SKILLBUILDER:
GER1      DATA #>02A9, : WJRTERRADAR:
GER2      DATA #>018B, : BELIEBIGE:
GER3      DATA #>01AD, : TASTE:
GER4      DATA #>01CB, : DR\CKEN UM:
GER5      DATA #>01EB, : ANZUFANGEN:
GER6      DATA #>02E8, : AID = SPIELWEISE:
GER7      DATA #>0049, : WJRTERRADAR:
GER8      DATA #>008A, : SPIELWEISE:
GER9      DATA #>00E2, : 1 GESCHWINDIGKEIT 1-9:
GER10     DATA #>0122, : 2 WJRTERLISTE 1-9:
GER11     DATA #>0162, : 3 LAUFZEIT (MIN) 1-5:
GER12     DATA #>01A2, : 4 \BUNGSSTUFE 1-3:
GER13     DATA #>01E2, : 5 KEINE INDERUNGEN MEHR:
GER14     DATA #>0285, : ZIFFER DER GEW\NSCHTEN:
GER15     DATA #>02CB, : INDERUNG DR\CKEN:
GER16     DATA #>02C0, : RICHTIG:
GER17     DATA #>02D5, : FALSCH:
GER18     DATA #>0049, : WJRTERRADAR:
GER19     DATA #>00AA, : PUNKTESTAND:
GER20     DATA #>0108, : ZULETZT:
GER21     DATA #>0112, : MIN:
GER22     DATA #>011A, : MAX:
GER23     DATA #>0161, : RICHTIG:
GER24     DATA #>01C1, : FALSCH:
GER25     DATA #>0284, : BELIEBIGE TASTE DR\CKEN:
GER26     DATA #>1A6D, : E, >E3, : E, >E5, : GY:
GER27     DATA #>1A84, >E5, : ICH, >E7, : IO:

```

GER28 DATA #>1A95, : FA: , >E1, >E6, : CH:

GEREOL

* DUTCH WORD LIST *

DUTCH

DATA #DCH0	ARCADEMIC SKILLBUILDER
DATA #DCH1	ALLIGATOR SOMMEN
DATA #DCH2	DUMMY LINE
DATA #DCH3	DRUK EEN
DATA #DCH4	TOETS IN
DATA #DCH5	OM TE BEGINNEN
DATA #DCH6	AID = SPELMOGELIJKHEDEN
DATA #DCH7	ALLIGATOR SOMMEN
DATA #DCH8	SPELMOGELIJKHEDEN
DATA #DCH9	1 VAARDIGHEIDSNIVEAU 1-9
DATA #DCH10	2 MOELILIJKHEIDSGRAAD 3, 6, 9
DATA #DCH11	3 TIJDSDUUR (MIN) 1-5
DATA #DCH12	4 AFSTANDBEDIENING J/N
DATA #DCH13	5 GEEN WIJZIGINGEN
DATA #DCH14	DRUK OP HET NUMMER VAN DE
DATA #DCH15	GEWENSTE VERANDERING
DATA #DCH16	GOED
DATA #DCH17	FOUT
DATA #DCH18	ALLIGATOR SOMMEN
DATA #DCH19	NOTEER JE SCORE
DATA #DCH20	NU
DATA #DCH21	LAAG
DATA #DCH22	HOOG
DATA #DCH23	GOED
DATA #DCH24	FOUT
DATA #DCH25	DRUK OP EEN WILLEKEURIGE TOETS
DATA #DCH26	
DATA #DCH27	
DATA #DCH28	
DATA #DCHEDL	* END OF LIST MARKER

* DUTCH WORD LIST *

*

DCH0	DATA #>01E5, : ARCADEMIC SKILLBUILDER:
DCH1	DATA #>02AA, : WOORD RADAR:
DCH2	DATA #>0000, :
DCH3	DATA #>01AB, : DRUK EEN:
DCH4	DATA #>01CB, : TOETS IN:
DCH5	DATA #>01E9, : OM TE BEGINNEN:
DCH6	DATA #>02E4, : AID = SPELMOGELIJKHEDEN:
DCH7	DATA #>004A, : WOORD RADAR:
DCH8	DATA #>0007, : SPELMOGELIJKHEDEN:
DCH9	DATA #>00E2, : 1 SPOED 1-9:
DCH10	DATA #>0122, : 2 WOORDENLIJST 1-9:
DCH11	DATA #>0162, : 3 TIJDSDUUR (MIN) 1-5:
DCH12	DATA #>01A2, : 4 VAARDIGHEIDSNIVEAU 1-3:
DCH13	DATA #>01E2, : 5 GEEN WIJZIGINGEN:
DCH14	DATA #>0285, : DRUK OP HET NUMMER VAN DE:
DCH15	DATA #>02C6, : GEWENSTE VERANDERING:
DCH16	DATA #>02C1, : GOED:
DCH17	DATA #>02D7, : FOUT:
DCH18	DATA #>004A, : WOORD RADAR:
DCH19	DATA #>00AB, : NOTEER DE SCORE:
DCH20	DATA #>010B, : NU:
DCH21	DATA #>0112, : LAAG:

DCH22 DATA #>011A, : HOOG:
DCH23 DATA #>0161, : GOED:
DCH24 DATA #>01C1, : FOOT:
DCH25 DATA #>0281, : DRUK OP EEN WILLEKEURIGE TOETS:
DCH26 DATA #>1A6D, : E:, >E3, : E:, >E5, : GY:
DCH27 DATA #>1A85, : GOED:
DCH28 DATA #>1A96, : FOU:, >E7
DCHEOL

END

GROM 5
ORG 0

* G R O M F I V E S P E E C H D A T A *

DUTWRD EQU * DUTCH WORD LIST START

DATA : I:, >E0, : : MY
DATA : I:, >E6, : : IS
DATA : WE : WE
DATA : OP : ON
DATA : I:, >E3, : : IN
DATA : ZO : TO
DATA : ZE : HE
DATA : GA : GO
DATA : DO : DO
DATA : :, >E2, : E : ME
DATA : O:, >E2, : : BE
DATA : A:, >E6, : : AT
DATA : DE : IT
DATA : OP : OF
DATA : :, >E3, : A : NO
DATA : :, >E7, : E : AM
DATA : FO:, >E0, : : FOR
DATA : BO:, >E6, : : BOY
DATA : :, >E0, : A:, >E3, : : CAN
DATA : UW : UP
DATA : :, >E0, : A:, >E7, : : CAT
DATA : HE:, >E7, : : HE
DATA : :, >E9, : EE : NOT
DATA : O:, >E6, : : OR

*----- S E T O N E

DATA : ZOG : DOG
DATA : OO:, >E0, : : AND
DATA : :, >E2, : A:, >E3, : : MAN
DATA : WA:, >E7, : : ANY
DATA : HAD : HAD
DATA : WA:, >E6, : : WAS
DATA : HIP : HIS
DATA : BU:, >E6, : : BUY
DATA : ZIE : SEE
DATA : HOE : HOW
DATA : HIJ : HIM
DATA : O:, >E3, >E6, : : OUR
DATA : DOE : GET
DATA : A:, >E5, : E : ARE
DATA : HEB : HER
DATA : UI:, >E7, : : OUT
DATA : ZAG : SAW
DATA : DI:, >E7, : : DID
DATA : OUD : OLD
DATA : BUI : BUT
DATA : :, >E3, : E:, >E0, : : NEW
DATA : ZIJ : SHE
DATA : WIE : WHO
DATA : :, >E5, : AP : RAN

*----- S E T T W O

DATA : :, >E5, : UG : RUN
DATA : EE:, >E3, : : ONE
DATA : :, >E5, : O:, >E6, : : RED
DATA : :, >E7, : A:, >E6, : : TWO
DATA : ZE:, >E7, : : PUT
DATA : DO:, >E1, : : BIG

DATA : EE,,>E7,, : EAT
 DATA : ,,>E0,,A,,>E6,, : ASK
 DATA : ZEG : SAY
 DATA : HAP : HAS
 DATA : DAG : DAY
 DATA : ,,>E2,,AG : MAY
 DATA : UI,,>E1,, : USE
 DATA : ,,>E1,,E,,>E6,, : TEL
 DATA : FU,,>E7,, : FUN
 DATA : WA,,>E5,, : WHY
 DATA : BOX : BOX
 DATA : VE,,>E5,, : FAR
 DATA : ,,>E0,,A,,>E5,, : CAR
 DATA : ,,>E7,,OP : TOO
 DATA : ,,>E7,,AP : TRY
 DATA : ZI,,>E7,, : SIT
 DATA : WEG : WAY
 DATA : JOU : YOU

*----- S E T T H R E E

DATA : ,,>E1,,IJ,,>E3,, : LINE
 DATA : WI,,>E6,,>E7,, : WILL
 DATA : ,,>E7,,>E5,,E,,>E0,, : DRAW
 DATA : HA,,>E1,,F : HAVE
 DATA : DEZE : THIS
 DATA : ,,>E3,,AA,,>E2,, : NAME
 DATA : BOE,,>E0,, : BOOK
 DATA : ,,>E2,,EID : GIRL
 DATA : ,,>E1,,IJ,,>E0,, : LIKE
 DATA : HUI,,>E6,, : HOME
 DATA : A,,>E1,,>E1,,E : THEY
 DATA : F,,>E5,,I,,>E6,, : FROM
 DATA : ,,>E1,,EE,,>E6,, : READ
 DATA : WEE,,>E3,, : WHEN
 DATA : WE,,>E5,,D : WORD
 DATA : ,,>E6,,O,,>E2,,>E6,, : SOME
 DATA : ,,>E7,,HEE : THEM
 DATA : ZOU,,>E7,, : YOUR
 DATA : WIE,,>E1,, : WITH
 DATA : GOED : GOOD
 DATA : WAA,,>E5,, : WHAT
 DATA : WE,,>E1,,>E0,, : THAT
 DATA : BA,,>E1,,>E0,, : BALL
 DATA : WI,,>E1,,D : GAME

*----- S E T F O U R

DATA : ,,>E6,,PE,,>E1,, : PLAY
 DATA : E,,>E5,,OP : INTO
 DATA : ,,>E0,,WA,,>E2,, : CAME
 DATA : WE,,>E5,,>E0,, : WORK
 DATA : GEEF : GIVE
 DATA : BOO,,>E2,, : TREE
 DATA : ZEE,,>E5,, : VERY
 DATA : ,,>E3,,EE,,>E2,, : TAKE
 DATA : ,,>E1,,AAG : DOWN
 DATA : ,,>E1,,A,,>E3,,G : LONG
 DATA : WA,,>E5,,E : WERE
 DATA : ,,>E6,,>E3,,O,,>E5,, : SAID
 DATA : ,,>E6,,>E7,,AD : TOWN
 DATA : V,,>E1,,UG : SOON
 DATA : ,,>E0,,IJ,,>E0,, : LOOK
 DATA : BA,,>E0,,>E7,, : BACK
 DATA : VO,,>E3,,D : FIND
 DATA : JAA,,>E5,, : YEAR

DATA : : >E1, : IE, : >E7, : : LEFT
 DATA : VIE, : >E5, : : FOUR
 DATA : AA, : >E5, : D : AWAY
 DATA : WA, : >E3, >E7, : : WANT
 DATA : WE, : >E3, >E6, : : WISH
 DATA : : >E1, : A, : >E6, >E7, : : LAST

*----- S E T F I V E
 DATA : : >E6, >E7, : U, : >E0, : : SHOW
 DATA : BEC, : >E3, : : BEEN
 DATA : : >E7, : AA, : >E1, : : TALL
 DATA : BE, : >E6, >E7, : : BEST
 DATA : JU, : >E5, >E0, : : JUMP
 DATA : VEE, : >E1, : : MUCH
 DATA : E, : >E1, >E0, : E : EACH
 DATA : VO, : >E1, : G : NEXT
 DATA : : >E2, : EE, : >E5, : : MORE
 DATA : : >E7, : WEE : BOTH
 DATA : VI, : >E6, >E7, : : FISH
 DATA : HOOG : HIGH
 DATA : : >E0, : I, : >E3, : D : KIND
 DATA : A, : >E1, : ZO : ALSO
 DATA : : >E5, : DEP : CALL
 DATA : : >E2, : DE, : >E7, : : MUST
 DATA : : >E3, : AA, : >E5, : : NEAR
 DATA : VIJF : FIVE
 DATA : OVE, : >E5, : : OVER
 DATA : WEE, : >E7, : : KNOW
 DATA : E, : >E3, : IG : ONLY
 DATA : DA, : >E5, >E2, : : DARK
 DATA : : >E2, : A, : >E6, >E7, : : MOST
 DATA : BE, : >E5, : G : HILL

*----- S E T S I X
 DATA >E7, : HUI, : >E6, : : HOUSE
 DATA : WIJZE : WRITE
 DATA : O, : >E3, : DE, : >E5, : : UNDER
 DATA >E3, : ABIJ : ABOUT
 DATA >E6, >E7, : OO, : >E5, : : STORY
 DATA : WI, : >E1, : DE : WOULD
 DATA : BOE, : >E0, >E7, : : BOOKS
 DATA >E7, : HA, : >E3, >E6, : : THERE
 DATA : PA, : >E7, : E, : >E5, : : PAPER
 DATA : OUDE, : >E5, : : AFTER
 DATA : WOO, : >E5, : D : WORDS
 DATA : A, : >E3, : DE, : >E5, : : OTHER
 DATA >E6, >E7, : A, : >E3, : D : STAND
 DATA : DE, : >E3, >E0, >E7, : : THINK
 DATA : WEDE, : >E5, : : AGAIN
 DATA : WE, : >E1, >E0, : E : WHICH
 DATA >E0, : U, : >E3, >E6, >E7, : : COULD
 DATA >E0, >E1, : EU, : >E5, : : COLOR
 DATA : ZA, : >E0, : E, : >E3, : : THING
 DATA : O, : >E5, : O, : >E3, : D : FOUND
 DATA >E1, : U, : >E3, : CH : LUNCH
 DATA : WEE, : >E5, >E7, : : WHERE
 DATA : WI, : >E7, >E7, : E : WHITE
 DATA : B, : >E5, : E, : >E3, : G : BRING

*----- S E T S E V E N
 DATA : VE, : >E5, >E5, : E : LEAVE
 DATA >E3, : OOI, : >E7, : : NEVER
 DATA : EUVE, : >E1, : : WHILE
 DATA : ZWA, : >E5, >E7, : : BLACK
 DATA >E5, : ECH, : >E7, : : RIGHT

DATA : E, , >E3, >E0, : E, , >E1, : :	THESE
DATA : UI, : >E1, : E, , >E3, : :	UNTIL
DATA : V, : >E5, : OUW :	WOMAN
DATA >E2, : OEDE, : >E5	MOTHER
DATA : VADE, : >E5, >E6	FATHER
DATA >E1, : E, , >E7, >E7, : E, , >E5	LETTER
DATA : WEI, : >E3, : IG:	LITTLE
DATA : VOO, : >E5, : A, : >E1	BEFORE
DATA >E6, : CHOO, : >E1	SCHOOL
DATA : ZU, : >E6, >E7, : E, , >E5	SISTER
DATA : V, : >E5, : IE, : >E3, : D:	FRIEND
DATA >E5, : UI, : >E0, : E, , >E5	FLOWER
DATA : AA, : >E5, : DIG:	PRETTY
DATA >E0, : EU, : >E5, : IG:	PLEASE
DATA >E2, : OE, : >E7, : E, , >E3	SHOULD
DATA >E3, : E, , >E7, >E7, : E, , >E5	BETTER
DATA : A, : >E1, >E6, >E3, : OG:	SECOND
DATA : CI, : >E5, >E0, : E, , >E1	CIRCLE
DATA : EE, : >E5, >E6, >E7, : :	FIRST

*----- S E T E I G H T

ORG >0800

*

GERWRD	EQU \$	GERMAN WORD	LIST	START
DATA : , : >E2, : E1, : >E3, : :			MY	
DATA : I, : >E6, >E7, : :			IS	
DATA : WI, : >E5, : :		WE		
DATA : AUF :		ON		
DATA : I, : >E3, : :			IN	
DATA : ZU :		TO		
DATA : E, : >E5, : :			HE	
DATA : GEH :		GO		
DATA : , : >E7, : U, : >E3, : :			DO	
DATA : , : >E2, : I, : >E5, : :			ME	
DATA : , : >E6, : E1 :			BE	
DATA : BEI :		AT		
DATA : E, : >E6, : :			IT	
DATA : VO, : >E3, : :			OF	
DATA : , : >E0, : E1, : >E3, : :			NO	
DATA : BI, : >E3, : :			AM	
DATA : F, : >E5, : :				
DATA : JU, : >E3, : GE :			BOY	
DATA : , : >E0, : A, : >E3, >E3, : :				CAN
DATA : AUF :		UP		
DATA >E0, : A, : >E7, : ZE :			CAT	
DATA : DA, : >E6, : :		HE		
DATA : , : >E3, : ICH, : >E7			NOT	
DATA : ODE, : >E5, : :		OR		

*----- S E T O N E

DATA : HU, : >E3, : D :		DOG		
DATA : U, : >E3, : D :		AND		
DATA : , : >E2, : A, : >E3, >E3, : :			MAN	
DATA : JEDE, : >E5, : :		ANY		
DATA : HA, : >E7, >E7, : E :			HAD	
DATA : WA, : >E5, : :		WAS		
DATA : , : >E6, : E1, : >E3, : :			HIS	
DATA >E0, : AUFE, : >E3			BUY	
DATA >E6, : EHE, : >E3, : :			SEE	
DATA : WIE :		HOW		
DATA : IH, : >E2, : :			HIM	
DATA : U, : >E3, >E6, : E, : >E5, : :				OUR
DATA : HABE, : >E3, : :		GET		
DATA : , : >E6, : I, : >E3, : D :			ARE	

DATA : IH, >E5, : E :	HER
DATA : AU, >E6, : :	OUT
DATA >E6, : [GE, >E3, : :	SAW
DATA : , >E7, : A, >E7, : :	DID
DATA : A, >E1, >E7, : :	OLD
DATA : ABE, >E3, : :	BUT
DATA : , >E3, : EU :	NEW
DATA : , >E6, : IE :	SHE
DATA : WE, >E5, : :	WHO
DATA : , >E1, : IEF :	RAN

*----- S E T T W O

DATA >E5, : E, >E3, >E3, : E, >E3		RUN
DATA : EI, >E3, >E6, : :	ONE	
DATA : , >E5, : O, >E7, : :	RED	
DATA : ZWEI :	TWO	
DATA >E1, : EGE, >E3, : :	PUT	
DATA : G, >E5, : O, >E6, >E6, : :		BIG
DATA : E, >E6, >E6, : E, >E3, : :		EAT
DATA : F, >E5, : AGE, >E3	ASK	
DATA >E6, : AGE, >E3, : :	SAY	
DATA : HABE :	HAS	
DATA : , >E7, : AG :	DAY	
DATA : WD, >E1, >E1, : E, >E3		MAY
DATA : B, >E5, : AUCH:	USE	
DATA >E1, : A, >E6, >E6, : E, >E3		TEL
DATA >E6, : PA, >E6, >E6, : :		FUN
DATA : WA, >E5, : U, >E2, : :		WHY
DATA >E0, : OFFE, >E5	BOX	
DATA : WEI, >E7, : :	FAR	
DATA : WAGE, >E3, : :	CAR	
DATA : ZU :	TOO	
DATA : P, >E5, : \FE, >E3		TRY
DATA >E6, : I, >E7, : ZE, >E3		SIT
DATA : WEG :	WAY	
DATA : DU :	YOU	

*----- S E T T H R E E

DATA >E1, : I, >E3, : IE :		LINE
DATA : WI, >E1, >E1, : E :		WILL
DATA : , >E1, : O, >E6, : :		DRAW
DATA : HABE, >E3, : :	HAVE	
DATA : DIE, >E6, : :	THIS	
DATA : , >E3, : A, >E2, : E :		NAME
DATA : BUCH :	BOOK	
DATA : E, >E6, : :	GIRL	
DATA : WIE :	LIKE	
DATA : HEI, >E2, : :	HOME	
DATA : , >E6, : IE :	THEY	
DATA : VO, >E3, : :	FROM	
DATA >E1, : E, >E6, : E, >E3, : :		READ
DATA : WE, >E3, >E3, : :		WHEN
DATA : WD, >E5, >E7, : :		WORD
DATA : EI, >E3, : ICE:	SOME	
DATA : IH, >E3, : E, >E3, : :		THEM
DATA : EU, >E5, : E :	YOUR	
DATA : , >E2, : I, >E7, : :		WITH
DATA : GU, >E7, : :	GOOD	
DATA : WA, >E6, : :	WHAT	
DATA : DA, >E6, : :	THAT	
DATA : BA, >E1, >E1, : :		BALL
DATA >E6, : PIE, >E1, : :		GAME

*----- S E T F O U R

DATA >E6, : PIE, >E1, : :	PLAY
---------------------------	------

DATA : I, >E3, : : INTO
 DATA : , >E0, : A, >E2, : : CAME
 DATA : A, >E5, : BEI, >E7 : WORK
 DATA : GEBE, >E3, : : GIVE
 DATA : BAU, >E2, : : TREE
 DATA : , >E6, : EH, >E5, : : VERY
 DATA : , >E3, : I, >E2, >E2, : : TAKE
 DATA : U, >E3, >E7, : E, >E3, : : DOWN
 DATA : , >E1, : A, >E3, : O : LONG
 DATA : WA, >E5, : E, >E3, : : WERE
 DATA >E6, : AG, >E7, : E : SAID
 DATA >E6, >E7, : AD, >E7, : : TOWN
 DATA : BA, >E1, : D : SOON
 DATA >E6, : CHAU : LOOK
 DATA >E5, : \C, >E0, : E, >E3 : BACK
 DATA : F1, >E3, : DE, >E3 : FIND
 DATA : JAH, >E5, : : YEAR
 DATA >E1, : I, >E3, >E0, >E6, : : LEFT
 DATA : VIE, >E5, : : FOUR
 DATA : FO, >E5, >E7, : : AWAY
 DATA : , >E3, : O, >E7, : : WANT
 DATA : WU, >E3, >E6, : CH : WISH
 DATA : E, >E3, : DE : LAST

*----- S E T F I V E
 DATA >E6, : CHAU : SHOW
 DATA : , >E6, : EI, >E3, : : BEEN
 DATA : G, >E5, : O, >E6, >E6, : : TALL
 DATA : BE, >E6, >E7, : E : BEST
 DATA >E6, : P, >E5, : U, >E3, : G : JUMP
 DATA : VIE, >E1, : : MUCH
 DATA : JEDE, >E5, : : EACH
 DATA >E3, : ICH, >E6, >E7 : NEXT
 DATA : , >E2, : EH, >E5, : : MORE
 DATA : BEIDE : BOTH
 DATA : FI, >E6, : CH : FISH
 DATA : HOCH : HIGH
 DATA : , >E1, : IEB : KIND
 DATA : AUCH : ALSO
 DATA : , >E3, : UF : CALL
 DATA : , >E2, : U, >E6, >E6, : : MUST
 DATA : , >E3, : AHE : NEAR
 DATA : F, >E3, : F : FIVE
 DATA : \BE, >E5, : : OVER
 DATA : WI, >E6, >E6, : E, >E3 : KNOW
 DATA : , >E3, : U, >E5, : : ONLY
 DATA : DU, >E3, >E0, : E, >E1 : DARK
 DATA >E2, : EI, >E6, >E7, : : MOST
 DATA : BE, >E5, : G : HILL

*----- S E T S I X
 DATA : HAU, >E6, : : HOUSE
 DATA : BEFEH, >E1 : WRITE
 DATA : U, >E3, >E7, : E, >E5, : : UNDER
 DATA : \BE, >E5, : : ABOUT
 DATA : , >E6, : AGE : STORY
 DATA : W, >E5, : DE : WOULD
 DATA : B\CHE, >E5 : BOOKS
 DATA : HIE, >E5, : : THERE
 DATA : PAPIE, >E5 : PAPER
 DATA : , >E3, : ACH : AFTER
 DATA : WO, >E5, >E7, : E : WORDS
 DATA : A, >E3, : DE, >E5, : E : OTHER
 DATA >E6, >E7, : A, >E3, : D : STAND

DATA : DE, , >E3, >E0, : E : THINK
 DATA : WIEDE, , >E5 AGAIN
 DATA : WA, , >E6, : WHICH
 DATA >E0, : J, , >E3, >E3, : E : COULD
 DATA : FA, , >E5, : BE : COLOR
 DATA : DI, , >E3, : G : THING
 DATA : FA, , >E3, : D : FOUND
 DATA : I, , >E2, : BI, , >E6, >E6 LUNCH
 DATA : WO : WHERE
 DATA : WEI, , >E6, >E6, : : WHITE
 DATA : B, , >E5, : I, , >E3, : G : BRING

*----- S E T S E V E N

DATA : U, , >E5, >E1, : AUB: LEAVE
 DATA : , , >E3, : IE : NEVER
 DATA : WEI, , >E1, : E : WHILE
 DATA : DU, , >E3, >E0, : E, , >E1 BLACK
 DATA >E5, : ECH, , >E7, >E6 RIGHT
 DATA : DIE, , >E6, : E : THESE
 DATA : BI, , >E6, : : UNTIL
 DATA : F, , >E5, : AU : WOMAN
 DATA >E2, : U, , >E7, >E7, : E, , >E5 MOTHER
 DATA : VA, , >E7, : E, , >E5, : : FATHER
 DATA : B, , >E5, : IEF : LETTER
 DATA >E0, >E1, : EI, , >E3, : : LITTLE
 DATA : VO, , >E5, : : BEFORE
 DATA >E6, : CHU, , >E1, : E: SCHOOL
 DATA : , , >E2, : AID : SISTER
 DATA : F, , >E5, : EU, , >E3, : D: FRIEND
 DATA : B, , >E1, : U, , >E2, : E : FLOWER
 DATA : , , >E3, : E, , >E7, >E7, : : PRETTY
 DATA : BI, , >E7, >E7, : E : PLEASE
 DATA >E6, : O, , >E1, >E1, >E7, : E: SHOULD
 DATA : BE, , >E6, >E6, : E, , >E5 BETTER
 DATA : ZWEI, , >E7, : E: SECOND
 DATA >E0, >E5, : EI, , >E6, : : CIRCLE
 DATA : E, , >E5, >E6, >E7, : E : FIRST

*----- S E T E I G H T

END