Includes Magnetic Cards

HEWLETT-PACKARD

HP-75

USERS' LIBRARY SOLUTIONS

Games I

* * * HP-75 * * *

NOTICE

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

SERIES 70 SOLUTIONS BOOK/MEDIA ORDER FORM

Use this form to order Solutions Books and send to:

SERIES 70 USERS' LIBRARY
Hewlett-Packard Company
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330

OR, CONTACT YOUR LOCAL DEALER OR HP SALES OFFICE

HP-75 Solutions Books are $35.00 each, including magnetic cards (multiple mini-cassette copies of individual Solutions Books can be obtained from the Users' Library for $12.00 each).

HP-75 Users' Library Solutions Books

00075-13003	Math I	00075-13008	Electronics
00075-13004	Math II	00075-13009	Finance
00075-13005	Math III	00075-13010	Real Estate
00075-13006	Games I	00075-13011	Statistics
00075-13007	Games II	00075-13012	Test Statistics

00075-13013 I/O Utilities (use of this product requires an understanding of the HP-75 and of frame level HP-IL protocol)

HP-75 Solutions Books

PRODUCT NUMBER   TITLE

00075-13 |__|__|__| _____   00075-13 |__|__|__| _____
00075-13 |__|__|__| _____   00075-13 |__|__|__| _____
00075-13 |__|__|__| _____   00075-13 |__|__|__| _____
00075-13 |__|__|__| _____   00075-13 |__|__|__| _____

☐ CASH   ☐ PURCHASE ORDER _____

☐ VISA   ☐ MASTER CARD (BANK ID) |__|__|__|__|__|

CARD NUMBER _____ EXP _____

B2H9

PLEASE PRINT CLEARLY

Ship to _____ Name _____

Company (if applicable) _____

Address _____

City _____ State/Country _____

Telephone Number _____ Zip Code/Mail Code _____

_____ @ $35.00 / Solutions Book = _____
Quantity

Handling Charge = $3.50

SUBTOTAL = _____

State & Local Sales
or Use Tax + = $_____

For addresses outside the continental U.S.A. and Canada, please add 10% for postage and handling.

Total Amount
Enclosed = $_____

☐ If order is tax exempt check this box and furnish your exemption number or other proof of exemption

Exemption No _____

Have you...?
■ Enclosed check or money order
■ Signed your check
■ Included your zip or mail code
■ Written in correct Solutions Book numbers

Prices subject to change without notice.

HEWLETT PACKARD

"BLANK"

TABLE OF CONTENTS

"BLANC"

PROGRAM DESCRIPTION

ADVENTURE

This adventure is set in a mysterious mansion with 38 rooms. Some rooms
may have monsters and/or treasure. The object of the game is to find and
score all the treasures. This program requires a memory module.

Since the nature of the adventure is its mystery, all of the commands and
procedures cannot be revealed. The player can make moves in six directions
(east, south, north, west, up, down) by entering their respective first
letters for commands. The player must use intuition and guesswork to deter-
mine commands. A command consists of a single letter (direction), a single
verb (e.g. "LOOK"), or a verb and noun or noun phrase (such as "GO DOOR").

If the player decides to map out the mansion on paper, he should keep in
mind that some rooms may have different sizes. Connecting rooms will have
to be determined by their respective descriptions.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Load new copy of ADVFILE (see NOTES AND REFERENCES section).		
2	Run program	ADVENTURE	
3	Initialize	Waking up monsters!!!	
4	Display room description and objects visible	You see *description*	
5	Display possible directions to take	You can go .... *directions*	
6	Enter command	Command =	'direction' [RTN] or 'verb' [RTN] or 'verb noun' [RTN]
	The program will assess the command, take the desired steps, and goto step 4 for a new room or step 6 for remain- in the same room. To score a treasure, you must return it to the appropriate room and leave it.		
	When your score reaches 350 points, you will have mastered the adventure and the game will end.	You're an Adventure Grandmaster!	
		END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
L2	Length of command	M5	Monster number
L1	Length of verb part of command	B( )	Indicates location and value of objects
C5	Room number	P( )	Directions the player can go from each room
I	Looping & indexing	C1	Chest has been opened
Z	Indicates player is carrying something	U	Door is open or closed
T	Number of items player is carrying	E	Clock has been activated
N	Noun number	L	Line number for determining room objects
C	Indicates bottle empty	O$	Objects in a room
S	Score	P$	Room description
K	Indicates player's chance of killing monster	M$	Current monster
M	Indicates monster's chance of killing player	C$	Command
A5	Indicates player has already attacked once	N5$	List of possible nouns
V5	Verb number	V5$	List of possible verbs
F	Shows that player didn't attack immediately	A2$	Phrase for monster attack
D1	Shows that a door is open or closed	A1$	Phrase for player attack
S5	Indicates strength factor	D$	Directions
Q	Looping & indexing	M1$	Monsters
B1	Shows if any visible objects are in room	V$	Verb part of command
V	Indicates vial empty	N$	Noun part of command

NOTES AND REFERENCES

Notes: 1.   ADVFILE must be loaded each time to run the VENTURE program
             because changes are made in the data file during execution.
             This does not mean that the game status is preserved. It is
             usually a good idea to keep a copy of ADVFILE in RAM so that
             between games you can purge the old file and rename the copy
             to ADVFILE, and then make a new copy.

         2.   This program requires a memory module.

         3.   HINT: If the player has trouble with commands, he should
                    try examining things.

PROGRAM LISTING

```
 10 ! VENTURE - An
 20 ! adventure game.
 30 ! Revision 11/01/82
 40 DELAY 3 @ OPTION BASE 0 @ RANDOMIZE
 50 INTEGER L2,L1,C5,I,Z,T,N,C,S,K,M,A5
    ,V5,F,D1,S5,Q,B1,V,M5,B(43,1),P(6),
    C1,U,E,L
 60 DIM O$[32],P$[32],M$[9],C$[32],N5$[
    342],V5$[10],A2$[15],A1$[20],D$[30]
    ,M1$[108],V$[20]
 70 DIM N$[10]
 80 S5,F,T,D1,C5,C1,U,K,M,A5,C,V,Z,E,B1
    ,Q9,L,B(0,0),B(0,1)=0
 90 GOTO 2170
100 ON ERROR GOTO 110
110 INPUT 'Command:'; C$ @ C$=UPRC$(C$)      -Command entry
120 IF LEN(C$)=0 THEN 110
130 OFF ERROR
140 L2=LEN(C$) @ V$='' @ N$='' @ V5=0 @
    N=0 @ M$[1,1]='X'
150 IF L2=1 THEN V$=C$ @ GOTO 190            -Check for single letter
                                              command
160 FOR Q=1 TO L2
170 IF C$[Q,Q]#" " THEN V$=V$&C$[Q,Q] E      -Find verb part of command
    LSE Q=L2
180 NEXT Q
190 L1=LEN(V$) @ V$=V$&'         ' @ V$=
    V$[1,9]
200 RESTORE 2620
210 FOR Q=1 TO 24 @ READ V5$[1,9]            -Check for valid verb
220 IF V$[1,9]=V5$[1,9] THEN V5=Q
230 NEXT Q @ IF V$[1,4]='QUIT' THEN 210
    0
240 IF V5=0 THEN DISP 'What??' @ GOTO 1
    00
250 IF L1+1)=L2 THEN N=0 @ GOTO 300          -Check for single word command
260 N$[1,9]=C$[L1+2,L2]
270 FOR Q=1 TO 38
280 IF N$[1,9]=N5$[Q*9-8,Q*9] THEN N=Q       -Check for valid noun
    @ Q=38
290 NEXT Q
300 IF S5=1 THEN C=C+1 ELSE 330
310 IF C>11 THEN S5=0 @ C=0
320 L=N
330 GOSUB 2420                               -Read room description and
                                              objects
340 IF C5=12 AND M$[1,1]#'X' AND V5>6 A
    ND (V5#8 AND V5#23) THEN 350 ELSE 3
    60
350 DISP 'The dragon breathes and' @ DI
    SP 'burns you to a cinder!!' @ GOTO
     2100
360 IF F=1 AND V5>6 AND V5#8 AND V5#23        -Check to see if player took
    THEN DISP 'The ';M$;' attacks,' ELS      initiative in attack
    E 380
```

```
370 DISP 'and kills you !!' @ GOTO 2100
380 IF V5>10 THEN 400

390 ON V5 GOTO 450,450,450,450,450,450,
    490,570,970,1020
400 IF V5>20 THEN 420
410 ON V5-10 GOTO 1020,1020,1140,1190,1
    190,1300,1480,1500,1600,1600
420 ON V5-20 GOTO 1830,1960,1990,2070
430 DISP "I don't understand ";V$
440 GOTO 100
450 IF P(V5)#0 THEN 480
460 F=0
470 DISP "Can't go in that direction."
    @ GOTO 100
480 C5=P(V5) @ F=0 @ GOTO 1830
490 IF C5=3 AND N=1 AND D1=1 THEN DISP
    "O.K." @ C5=4 @ GOTO 1830
500 IF C5=3 AND N=1 THEN DISP 'The door
     is closed.' @ GOTO 100
510 IF C5=27 AND N=1 AND U=1 THEN DISP
    'O.K.' @ C5=29 @ GOTO 1830
520 IF C5=27 AND N=1 THEN DISP 'The doo
    r is locked.' @ GOTO 100
530 IF C5=14 AND N=10 AND O$[1,12]='an
    open gate' THEN DISP 'O.K.' @ C5=16
     @ GOTO 1830
540 IF C5=14 AND N=10 THEN DISP 'The ga
    te is closed.' @ GOTO 100
550 IF N=38 AND C5=24 THEN C5=1 @ GOTO
    1830
560 DISP "I don't understand." @ GOTO 1
    00
570 IF M$[1,1]='X' THEN DISP "I don't f
    ollow you." @ GOTO 100
580 IF B(2,0)#-1 THEN DISP "You don't h
    ave any weapon!" ELSE 600
590 DISP "The ";M$;" tears you apart!"
    @ DISP 'You are dead.' @ GOTO 2100
600 IF N$[1,9]#M$[1,9] THEN DISP "It is
    n't here." @ GOTO 100
610 IF C5=12 THEN 950
620 IF C5#5 AND C5#9 AND C5#16 AND C5#2
    0 THEN 750
630 IF RND>.3 THEN K=1
640 IF RND>.85 THEN M=1
650 IF A5=1 THEN DISP 'again, ';
660 DISP A1$;M$ @ IF K=1 THEN 670 ELSE
    720
670 IF C5=16 THEN DISP "and it disappea
    rs!" @ PRINT # 1,N+40 ; "X" @ GOTO
    700
680 DISP "and chop its head off!"
690 PRINT # 1,N+40 ; "a dead "&M$
```

—Find appropriate routine for
 verb (command)

—Check for valid direction

—Routine for 'go' command

—Attack monster

—Check for rooms with monsters

—Probabilities for battle

PROGRAM LISTING

```
700 F=0 @ M=0 @ A5=0
710 K=0 @ M1$[M5*9-8,M5*9]="X" @ GOTO 1
    00
720 DISP 'and narrowly miss it.'
730 DISP 'The ';M$;' ';A2$ @ IF M=1 THE
    N DISP 'and it kills you!' @ GOTO 2
    100
740 DISP 'But it fortunately misses you
    .' @ A5=1 @ GOTO 630
750 IF N#30 AND N#23 AND N#11 AND N#8 T
    HEN 850
760 IF RND>.4 THEN K=1
770 IF RND>.8 THEN M=1
780 IF A5=1 THEN DISP 'Again, ';
790 DISP A1$;M$ @ IF K=1 THEN 800 ELSE
    820
800 DISP "and pierce its heart!" @ F=0
    @ A5=0 @ M=0
810 K=0 @ PRINT # 1,N+40 ; "a dead "&M$
    @ M1$[M5*9-8,M5*9]="X" @ GOTO 100
820 DISP 'But narrowly miss it.'
830 DISP 'The ';M$;' ';A2$ @ IF M=1 THE
    N DISP 'and kills you!' @ GOTO 2100
840 DISP 'But does not get you.' @ A5=1
    @ GOTO 760
850 IF RND>.25 THEN K=1          -Probabilities for battle
860 IF RND>.7 THEN M=1
870 IF A5=1 THEN DISP 'Again, ';
880 DISP A1$;M$ @ IF K=1 THEN 890 ELSE
    910
890 DISP "and it disappears!" @ K=0 @ M
    =0 @ A5=0 @ F=0
900 B(N,0)=0 @ M1$[M5*9-8,M5*9]="X" @ G
    OTO 100
910 DISP 'But nothing happens to it.'
920 DISP "The ";M$;" ";A2$ @ IF M=1 THE
    N 930 ELSE 940
930 DISP "and its touch paralyzes you!"
    @ GOTO 2100
940 DISP 'But you move out of its way.'
    @ A5=1 @ GOTO 850
950 DISP 'The dragon is impervious to'
    @ DISP 'ordinary weapons.  He breat
    hes'
960 DISP 'and burns you to a cinder!' @
    GOTO 2100
970 S=0 @ FOR I=1 TO 37          -Check score
980 IF B(I,0)=1 THEN S=B(I,1)+S
990 NEXT I @ DISP 'Out of a possible 35
    0 points,' @ DISP 'you have ';S;'.'
1000 IF S=350 THEN DISP "You're an Adven
     ture Grandmaster!" @ GOTO 2150
1010 GOTO 100
1020 IF N#4 AND N#22 AND N#26 THEN 1030   -Routine to eat something
     ELSE 1040
```

PROGRAM LISTING

```
1030 DISP V$;" a ";N$;"?" @ GOTO 100
1040 IF N=4 AND B(4,0)=-1 AND C=0 THEN 1
     050 ELSE 1080
1050 DISP "You feel a slight tingling."
     @ S5=1 @ C=1
1060 PRINT # 1,44 ; "an empty bottle"
1070 N5$[28,36]='BOTTLE' @ GOTO 100
1080 IF N=4 AND B(4,0)#-1 THEN DISP "You
      don't have it." @ GOTO 100
1090 IF N=26 AND B(N,0)=-1 AND V=0 THEN
     DISP "O.K. Nothing happened." ELSE
     1110
1100 V=1 @ PRINT # 1,66 ; "an empty vial
     " @ GOTO 100
1110 IF N=26 AND B(N,0)=-1 THEN DISP "Th
     e vial is empty." @ GOTO 100
1120 IF N=26 THEN DISP "You don't have i
     t." @ GOTO 100
1130 DISP "I don't see it here." @ GOTO
     100
1140 DISP "You are currently carrying:"     -Check to see what player is
     @ Z=0                                   carrying
1150 FOR I=1 TO 37
1160 IF B(I,0)=-1 THEN READ # 1,I+40 ; O
     $[1,25] @ DISP O$ @ Z=Z+1
1170 NEXT I @ IF Z=0 THEN DISP "Nothing
     at all."
1180 DISP @ GOTO 100
1190 IF N=0 AND LEN(N$)=0 THEN DISP V$;'    -Pick up something
     what?' @ GOTO 100
1200 IF N=0 THEN DISP "I don't know ";N$
     @ GOTO 100
1210 IF N=25 AND C5=32 AND B(N,0)=0 THEN
     B(N,0)=-1 @ DISP "Taken." @ GOTO 1
     00
1220 IF N=3+(B(8,0)=C5) OR (B(8,0)=-1)+B
     (N,0)=0 THEN DISP 'Taken.' @ B(N,0)
     =-1 @ GOTO 100
1230 IF B(N,0)=-1 THEN DISP 'You already
      have it!' @ GOTO 100
1240 IF B(N,0)#C5 THEN DISP "I don't see
      it here." @ GOTO 100
1250 IF N=22 THEN DISP 'It evaporates to
     o quickly.' @ GOTO 100
1260 IF N=1 OR N=10 OR N=29 THEN DISP "I
      can't." @ GOTO 100
1270 IF N=17 THEN DISP "It's much too he
     avy." @ GOTO 100
1280 IF T>=6 THEN DISP "You're carrying
     too much." @ GOTO 100
1290 B(N,0)=-1 @ T=T+1 @ DISP 'Taken.' @
     GOTO 100
1300 IF N#1 AND N#7 AND N#10 THEN DISP '    -Unlock or open something
     Huh?' @ GOTO 100
```

```
1310 IF D1=1 AND C5=3 AND N=1 THEN DISP
     "It's already open!" @ GOTO 100
1320 IF N=1 AND C5=3 THEN DISP "It just
     won't budge." @ GOTO 100
1330 IF N=7 AND B(7,0)=-1 AND C1=1 THEN
     DISP "It's already open." @ GOTO 10
     0
1340 IF N=7 AND B(7,0)=-1 THEN 1350 ELSE
     1360
1350 READ # 1,46 ; O$[1,14] @ DISP O$[1,
     14];' fall out!' @ B(6,0)=C5 @ C1=1
     @ GOTO 100
1360 IF N=7 THEN DISP "You don't have it
     ." @ GOTO 100
1370 IF N=10 AND C5=14 AND O$[1,12]="an
     open gate" THEN DISP "It's already
     open." @ GOTO 100
1380 IF N=10 AND C5=14 AND S5=1 THEN 139
     0 ELSE 1410
1390 DISP "After much struggling ..." @
     DISP "...the gate is open."
1400 PRINT # 1,50 ; "an open gate" @ GOT
     O 100
1410 IF N=1 AND U=1 THEN DISP "It's alre
     ady open." @ GOTO 100
1420 IF N=1 AND C5=27 AND B(25,0)=-1 THE
     N 1430 ELSE 1450
1430 DISP "After unlocking the door ..."
1440 DISP "...it swings open." @ U=1 @ P
     RINT # 1,78 ; "an open door" @ GOTO
     100
1450 IF N=10 AND C5=14 THEN DISP "This i
     s a heavy gate." @ GOTO 100
1460 IF N=1 AND C5=27 THEN DISP "The doo
     r is securely locked." @ GOTO 100
1470 DISP "I don't understand." @ GOTO 1
     00
1480 IF B(N,0)#-1 THEN DISP "You're not      -Leave something
     carrying it." @ GOTO 100
1490 B(N,0)=C5 @ DISP "O.K." @ T=T-1 @ G
     OTO 100
1500 IF C5#28 OR N#18 AND N#17 THEN DISP      -Change clock
     "What???" @ GOTO 100
1510 ON ERROR GOTO 1520
1520 DISP "To what time? e.g. 'TO 4'"
1530 DISP '(Use two words.) '; @ INPUT A
     $
1540 IF LEN(A$)=0 THEN 1530
1550 IF LEN(A$)<=4 THEN A$=A$&' ' @ GOTO
     1550
1560 IF A$[4,5]#"12" OR E=1 THEN DISP "N
     othing happened." @ GOTO 100
1570 OFF ERROR
1580 READ # 1,59 ; O$[1,15]
```

PROGRAM LISTING

```
1590 DISP "Out pops ";O$[1,15];"!!!" @ B
     (19,0)=28 @ E=1 @ GOTO 100
1600 IF N=0 THEN 1820
1610 IF N=1 AND C5=3 THEN DISP "It has n
     o knob." @ GOTO 100
1620 IF N=2 AND B(2,0)=-1 THEN DISP "It
     has a very sharp edge." @ GOTO 100
1630 IF N=4 AND B(4,0)=-1 THEN 1640 ELSE
     1650
1640 DISP "The bottle has a faded tag."
     @ DISP "It says: ST...TH PO...." @
     GOTO 100
1650 IF N=7 AND B(N,0)=-1 AND C1=0 THEN
     DISP "Seems to be something inside.
     " @ GOTO 100
1660 IF N=7 AND B(N,0)=-1 THEN DISP "You
      see nothing special." @ GOTO 100
1670 IF N=10 AND C5=14 THEN DISP "You se
     e a room beyond." @ GOTO 100
1680 IF N=17 AND C5=28 THEN DISP "There'
     s an inscription on it." ELSE 1700
1690 DISP "It says:HICKORY DICKORY DOCK.
     .." @ GOTO 100
1700 IF C5=28 AND N=18 THEN DISP "It's a
     bout 4:00." @ GOTO 100
1710 IF N=26 AND B(N,0)=-1 THEN 1720 ELS
     E 1730
1720 DISP "There's a label on it ..." @
     DISP "it says: 'DRAGON DISINTEGRATO
     R'." @ GOTO 100
1730 IF N=29 AND C5=32 AND B(25,0)=0 THE
     N 1740 ELSE 1750
1740 DISP "It's a skeleton clutching a k
     ey!" @ GOTO 100
1750 IF N=37 AND B(N,0)=-1 THEN 1760 ELS
     E 1770
1760 DISP "There's something written on
     it." @ DISP "Maybe you should read
     it?" @ GOTO 100
1770 IF N=8 AND B(3,0)=0 AND (B(8,0)=C5
     OR B(8,0)=-1) THEN 1780 ELSE 1790
1780 DISP "He's carrying a diamond purse
     !" @ GOTO 100
1790 IF N=2 OR N=4 OR N=7 OR N=26 OR N=3
     7 THEN 1800 ELSE 1810
1800 DISP "You don't have it." @ GOTO 10
     0
1810 DISP "You see nothing special." @ G
     OTO 100
1820 DISP "I don't know ";N$ @ GOTO 100
1830 DISP CHR$(27);"E" @ B1=0
1840 GOSUB 2440
1850 DISP P$ @ DISP "You see ";
1860 FOR I=1 TO 39 @ IF B(I,0)=C5 THEN R
     EAD # 1,I+40 ; O$[1,24] ELSE 1890
```

-Study something

-Look at room

PROGRAM LISTING

```
1870 IF O$[1,1]='X' THEN 1890
1880 DISP O$ @ B1=B1+1
1890 NEXT I @ IF B1=0 THEN DISP "nothing
     ."
1900 DISP
1910 DISP "You can go... ";
1920 FOR I=1 TO 6
1930 IF P(I)#0 THEN DISP D$[I*5-4,I*5];'
     ';
1940 NEXT I @ DISP @ IF M$[1,1]#"X" THEN
     F=1
1950 GOTO 100
1960 IF C5=3 AND N=9 AND D1=0 THEN DISP      -Say something
     "The door swings open." ELSE 1980
1970 PRINT # 1,41 ; "an open door" @ D1=
     1 @ GOTO 100
1980 DISP "O.K." @ DISP N$ @ GOTO 100
1990 IF N#26 THEN DISP "I don't understa      -Throw something
     nd." @ GOTO 100
2000 IF B(N,0)#-1 THEN DISP "You don't h
     ave it." @ GOTO 100
2010 IF C5=12 AND V=0 THEN 2020 ELSE 205
     0
2020 DISP "The vial hits the dragon ..."
     @ DISP "breaks into a thousand pie
     ces..."
2030 DISP "completely disintegrates him.
     " @ PRINT # 1,71 ; "a pile of bones
     "
2040 F=0 @ M1$[M5*9-8,M5*9]="X" @ B(26,0
     )=0 @ N5$[271,279]="bones" @ GOTO 1
     00
2050 DISP "The vial shatters and ..." @
     DISP "liquid spills across the floo
     r."
2060 B(26,0)=0 @ GOTO 100
2070 IF N=37 AND B(N,0)=-1 THEN DISP "It      -Study something
     says 'DROWCIGAM' on it." @ GOTO 10
     0
2080 IF N=37 THEN DISP "You don't have i
     t." @ GOTO 100
2090 DISP "I don't understand." @ GOTO 1
     00
2100 S=0
2110 FOR I=1 TO 37
2120 IF B(I,0)=1 THEN S=S+B(I,1)
2130 NEXT I
2140 DISP "Out of a possible 350 points.
     .." @ DISP "You had ";S
2150 DELAY 1 @ ASSIGN # 1 TO *
2160 DISP TAB(9);"END OF GAME" @ DISP @       -End of game
     STOP
2170 DISP TAB(11);'ADVENTURE'                 -Initialize game
2180 ASSIGN # 1 TO *
```

PROGRAM LISTING

```
2190  ASSIGN # 1 TO "ADVFILE"
2200  DISP '        Waking up monsters !!!'
2210  RESTORE # 1
2220  RESTORE 2240
2230  C5=24
2240  DATA North,East,South,West,Up,Down
2250  FOR I=1 TO 6 @ READ D$[I*5-4,I*5] @
      NEXT I
2260  M1$="GOBLIN   KOBOLD   DRAGON   GHO
      ST      ARMOR      "
2270  M1$=M1$&"THIEF    VAMPIRE  WEREWOLF
         OGRE       "
2280  M1$=M1$&"LIZARDMANSPIDER    ZOMBIE
      "
2290  A1$="you rush at the "
2300  A2$="rushes at you"
2310  DATA DOOR,SWORD,DIAMONDS,POTION,VAM
      PIRE,GEMS,CHEST,OGRE,DROWCIGAM,GATE
      ,LIZARDMAN,COINS
2320  DATA WEREWOLF,GOLD,GOBLIN,PAINTING,
      CLOCK,TIME,OPAL,GHOST,IDOL,LIQUID,S
      PIDER,SAPPHIRE
2330  DATA KEY,VIAL,THIEF,CHESS SET
2340  DATA FORMATION,ARMOR,DRAGON,SKULL,S
      TATUE,KOBOLD,ZOMBIE,CROWN,PAPER,STA
      IRWAY
2350  FOR I=1 TO 38 @ READ N5$[I*9-8,I*9]
      @ NEXT I
2360  DATA 3,0,1,0,0,30,4,0,23,0,0,20,18,
      0,31,0,0,0,14,0,33,0
2370  DATA 19,20,26,0,36,20,5,0,26,20,28,
      0,0,0,0,30,16,0,16,30,0,0
2380  DATA 36,0,4,30,0,0,29,0,20,0,6,20,0
      ,0
2390  DATA 17,0,12,0,38,30,23,20,9,0,38,0
      ,12,80,25,0,27,0,24,0
2400  FOR I=1 TO 39 @ READ B(I,0),B(I,1)
      @ NEXT I
2410  GOTO 1830
2420  IF L=0 THEN Q$='X' @ GOTO 2440
2430  READ # 1,L+40 ; O$
2440  RESTORE # 1
2450  FOR I=1 TO C5 @ READ # 1 ; P$[1,32]
      ,P(1),P(2),P(3),P(4),P(5),P(6) @ NE
      XT I
2460  M5=0
2470  IF C5=5 THEN M5=1
2480  IF C5=9 THEN M5=2
2490  IF C5=12 THEN M5=3
2500  IF C5=16 THEN M5=4
2510  IF C5=17 THEN M5=5
2520  IF C5=20 THEN M5=6
2530  IF C5=23 THEN M5=7
```

-List of possible nouns

-List of objects' locations and
 point values

-Read room description and
 objects

-Determine monster

PROGRAM LISTING

```
2540 IF C5=26 THEN M5=8
2550 IF C5=31 THEN M5=9
2560 IF C5=33 THEN M5=10
2570 IF C5=36 THEN M5=11
2580 IF C5=38 THEN M5=12
2590 IF M5=0 THEN M$[1,1]="X" @ GOTO 261
     0
2600 M$[1,9]=M1$[M5*9-8,M5*9]
2610 RETURN
2620 DATA N,E,S,W,U,D,GO,ATTACK,SCORE,SI
     P,DRINK,TASTE,INVENTORY,GET,TAKE,OP
     EN,DROP,SET
2630 DATA EXAMINE,SEARCH,LOOK,SAY,THROW,
     READ
```

—List of possible verbs and commands

PROGRAM LISTING

ADVFILE

```
1 DATA "Sign says 'Leave treasures he
  re'",0,2,13,0,0,24,"You're in a hal
  lway.",0,3,0,1,0,0
2 DATA "You're in a library.",0,0,5,2
  ,0,0,"This room appears to be a vau
  lt.",0,0,0,3,0,0
3 DATA "You're in a hall.",3,0,6,0,0,
  0,"You're in a game room.",5,8,7,0,
  0,0
4 DATA "You're in a storeroom.",6,0,0
  ,0,0,0,"This is a stairway.",0,0,0,
  0,9,6
5 DATA "You're in a bedroom.",0,0,10,
  8,0,0,"This is a hallway.",9,14,15,
  0,0,0
6 DATA 'Sign says:  "BEWARE!"',12,0,2
  1,0,0,0,"You're in the dragon's lai
  r.",0,0,11,0,0,0
7 DATA "You're in an elegant hall.",1
  ,0,0,28,0,0,"This is a stairwell.",
  0,0,0,0,0,10
8 DATA "You're in a dressing room.",1
  0,0,17,0,0,0,"This is the attic.",0
  ,0,0,14,0,0
9 DATA "You're in the master bedroom.
  ",15,18,0,0,0,0,"This is a closet."
  ,0,0,0,17,0,0
10 DATA "You're in the thief's room.",
   0,0,0,20,0,0,"You're in a 4-way int
   ersection.",21,19
11 DATA 39,22,0,0,"You hear a growl.",
   11,0,20,0,0,0,"This room is a templ
   e.",23,20,0,0,0,0
12 DATA "You're in the altar room.",0,
   0,22,0,0,0,"You're in front of the
   mansion.",0,0,0,0,1,0
13 DATA "This is the kitchen.",28,0,26
   ,0,0,0,"You're in the music room.",
   25,0,27,0,0,0
14 DATA "This is a sewing room.",26,0,
   30,0,0,0,"You're in the dining hall
   .",0,13,25,0,0,0
15 DATA "This is a secret storeroom.",
   0,27,0,0,0,0,"This is a stairway.",
   0,0,0,0,27,31
16 DATA "You're in a cellar.",0,32,0,3
   0,0,0,"There's a strange formation
   here",0,0,33,31,0,0
17 DATA "You're in a damp area.",32,34
   ,0,0,0,0,"You're in a passageway.",
   0,35,0,33,0,0
18 DATA "This is a T-intersection.",39
   ,36,0,34,0,0,"This is a cavern.",0,
   37,0,35,0,0
```

PROGRAM LISTING

```
19 DATA "You're in a twisting passage.
   ",38,0,0,36,0,0,"You're in an alcov
   e.",0,0,37,0,0,0
20 DATA "You're in a north/south hallw
   ay.",20,0,35,0,0,0
41 DATA a door
42 DATA a glowing sword
43 DATA a bag of diamonds
44 DATA a bottle of potion
45 DATA a hooded vampire
46 DATA assorted gems
47 DATA an oak chest
48 DATA an ugly ogre
49 DATA X
50 DATA a heavy iron gate
51 DATA a lizardman
52 DATA a box of coins
53 DATA a werewolf
54 DATA a bar of gold
55 DATA a mean-looking goblin
56 DATA a valuable painting
57 DATA an old clock
58 DATA X
59 DATA a large opal
60 DATA a ghost
61 DATA a bronze idol
62 DATA X
63 DATA a huge spider
64 DATA a huge sapphire
65 DATA a key
66 DATA a vial of liquid
67 DATA a scruffy thief
68 DATA a jade chess set
69 DATA X
70 DATA animated armor
71 DATA a large red dragon
72 DATA a crystal skull
73 DATA a gold statue
74 DATA a small kobold
75 DATA a zombie
76 DATA a jeweled platinum crown
77 DATA a ragged-edged paper
78 DATA a door
79 DATA steps up
```

PROGRAM DESCRIPTION

ROCKET LANDER

This game simulates the controls of a rocket landing vehicle. The player must try to land the rocket on a planet of his choice with as little velocity as possible. The planets he may land on are: Mercury, Mars, Pluto, Venus, Earth, Uranus, Neptune, Saturn, Jupiter. (The larger the planet, the greater its gravitational pull).

To begin, the player is given the distance above the surface, the initial velocity, the total fuel supply, the maximum burn allowed, and the amount of fuel needed to cancel the effects of gravity. He then enters the amount of fuel to be burned during the next one second. If he burns zero fuel, the rocket's velocity will increase. If he burns exactly the amount required to overcome gravity, then the velocity will remain constant. If he burns more, then the rocket will slow down or even start to move upward (velocity will be positive in this case).

If the player uses all his fuel, the rocket will free-fall and crash. The severity of the crash will depend upon the current height above the surface and the velocity at the time the fuel was spent.

Equations: $X = X_0 + V_0 t + \frac{1}{2} at^2$

$V = V_0 + at$

$V^2 = V_0^2 + 2a (X - X_0)$

Where: X is distance (height); X_0 is initial height; V is velocity; V_0 is initial velocity; a is acceleration; t is time.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		*** ROCKET LANDER ***	
1	Select planet	Choose planet (1-9):	1 or
	If '1' then	*** MERCURY ***	2 or
	If '2' then	*** MARS ***	3 or
	If '3' then	*** PLUTO ***	4 or
	If '4' then	*** VENUS ***	5 or
	If '5' then	*** EARTH ***	6 or
	If '6' then	*** URANUS ***	7 or
	If '7' then	*** NEPTUNE ***	8 or
	If '8' then	*** SATURN ***	9 then
	If '9' then	*** JUPITER ***	[RTN]
2	Display: initial height above	Initial height: n	
	surface, initial velocity,	Initial velocity: -n	
	total fuel supply,	Total fuel: n	
	maximum burn allowed,	Maximum burn: n	
	fuel needed to cancel gravity	Cancel gravity by: n	
3	Display condition headings:	Time Height Velocity Fuel Burn	
3a	Display current conditions and		
	enter burn:	n n n n :	fuel to burn [RTN]
3b	Goto step 3a until:		
	1) out of fuel		
	2) crashed, or		
	3) landed safely		
4	If out of fuel, display the		
	time when fuel ran out:	n OUT OF FUEL!!!	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	Show remaining conditions	Time Height Velocity	
	of fall until impact:	n n -n	
5	If crashed or landed:	Time: n	
		Impact velocity: -n	
	If fuel remained then:	Fuel remaining: n	
	Goto step 6		
6	If impact velocity very small		
	then display:	You are a qualified astronaut	
	If slightly greater, then:	Well, you're still in one piece!	
	If high velocity, then:	You made a new crater!!!	
	If very high velocity, then:	Next of kin will be notified	
7	Option to land on same surface		
	with same conditions:	Try same conditions again?Y	[RTN] or N [RTN]
	If 'Y' then step 2		
8	Option to land on same surface		
	with new conditions:	Try new conditions?Y	[RTN] or N [RTN]
	If 'Y' then change and goto		
	step 2		
9	Option to try a different		
	planet:	Try a different planet?Y	[RTN] or N [RTN]
	If 'Y' then step 1, else	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
X	Height	T8	Preliminary calculations to test rocket height
F	Fuel	M	Maximum allowable fuel burn
V	Velocity	G	Acceleration due to gravity
B1,B	Fuel burned during one second interval	K	Relative mass of planets, used to compute other constants
R	Used to test rockets velocity, acceleration and height	I	Looping - used in sound routines
T9	Preliminary calculations for time spent	Q$	User interaction - options
T	Time	A$	Choice of planets

NOTES AND REFERENCES

References: 1. "LANDER", HP-2000, HP Basic Program Library, Hewlett-Packard, 1973.

2. "MOON LANDING SIMULATOR", HP-33E Applications, Hewlett-Packard, 1978.

PROGRAM LISTING

```
 10 ! LANDER - Rocket
 20 ! landing simulation.
 30 ! Revision  11/01/82
 40 SHORT X,F,V,B1,T9,T8,T,M,G,B
 50 DIM Q$[3],A$[3]
 60 DEF FND(X)
 70 IF X<0 THEN FND=-IP(-X*100+.5)/100
    ELSE FND=IP(X*100+.5)/100
 80 END DEF
 90 DELAY 3
100 DISP TAB(5);'* * * ROCKET LANDER *
    * *'
110 ON ERROR GOTO 120
120 INPUT 'Choose planet (1-9):',CHR$(9
    5);A$
130 A$=UPRC$(A$[1,1])
140 ON POS('123456789',A$)+1 GOTO 110,1
    50,160,170,180,190,200,210,220,230
150 K=.25 @ DISP TAB(8);'* * MERCURY *
    *' @ GOTO 240
160 K=.5 @ DISP TAB(10);'* * MARS * *'
    @ GOTO 240
170 K=.75 @ DISP TAB(10);'* * PLUTO * *
    ' @ GOTO 240
180 K=1 @ DISP TAB(10);'* * VENUS * *'
    @ GOTO 240
190 K=1.25 @ DISP TAB(10);'* * EARTH *
    *' @ GOTO 240
200 K=1.5 @ DISP TAB(10);'* * URANUS *
    *' @ GOTO 240
210 K=1.75 @ DISP TAB(8);'* * NEPTUNE *
     *' @ GOTO 240
220 K=2 @ DISP TAB(10);'* * SATURN * *'
    @ GOTO 240
230 K=2.25 @ DISP TAB(8);'* * JUPITER *
    *'
240 OFF ERROR
250 G=5+27*K

260 M=30+60*K

270 X=500+1500*K

280 V=-50-100*K

290 F=IP(SQR(M*(V^2+2*G*X)/(M-G))*.13+.
    5)*10
300 DISP USING 310 ; 'Initial height: '
    ,X
310 IMAGE k,mdddddd.dd
320 DISP USING 310 ; 'Initial velocity:
    ',V
330 DISP USING 310 ; 'Total fuel: ',F
340 DISP USING 310 ; 'Maximum burn: ',M
```

-G is the acceleration due to gravity
-M is the maximum allowable burn
-X is the initial height above the surface of the planet
-V is the velocity (negative if going down)
-F is the total fuel supply for the rocket

PROGRAM LISTING

```
350 DISP USING 310 ; 'Cancel gravity by
    : ',G
360 DISP 'Time Height Velocity Fuel Bur
    n'
370 T=-1
380 T=T+1
390 ON ERROR GOTO 400
400 DISP T;FND(X);FND(V);FND(F);
410 INPUT ':','0';B
420 B1=ABS(B)
430 OFF ERROR
440 IF B1<=M AND B1<=F THEN 480        -Check for valid fuel burn
450 BEEP 100,.5
460 DISP 'Maximum burn is: ';MIN(FND(M)
    ,FND(F))
470 GOTO 390
480 T9,T8=2
490 IF B1=0 THEN 510                   -Avoid division by zero
500 T9=F/B1
510 A=B-G
520 R=V*V-2*A*X
530 IF R<0 THEN 590
540 IF A=0 THEN 570                    -Avoid division by zero
550 T8=-(V+SQR(R))/A
560 GOTO 590
570 IF V>=0 THEN 590
580 T8=-X/V
590 IF T8>0 AND T8<=1 OR T9<=1 THEN 660
600 X=X+V+A/2
610 V=V+A
620 F=F-B1
630 IF X>.0001 THEN 380                -Test altitude for safe landing
640 T=T+1
650 GOTO 880
660 IF T8>0 AND T8<=T9 THEN 850        -Test fuel supply
670 DISP T+T9;'  OUT OF FUEL !!!'
680 F,B1=0
690 X=X+V*T9+A*T9^2/2
700 V=V+A*T9
710 A=-G
720 T8=(V+SQR(V*V-2*A*X))/G
730 IF T8<1-T9 THEN 840                -Test altitude after fuel
                                        supply is depleted

740 X=X+V*(1-T9)+A*(1-T9)^2/2
750 V=V+A*(1-T9)
760 DISP 'TIME   HEIGHT   VELOCITY'
770 T=T+1
780 DISP T;FND(X);FND(V)
790 T8=(V+SQR(V^2-2*A*X))/G
800 IF T8<=1 THEN 850                  -Test fuel supply
810 X=X+V+A/2
820 V=V+A
830 GOTO 770
840 T=T+T9
```

PROGRAM LISTING

```
 850 F=F-B1*T8
 860 T=T+T8
 870 V=V+A*T8
 880 DISP 'Time: ';T
 890 DISP 'Impact velocity: ';FND(V)
 900 IF F=0 THEN 920
 910 DISP 'Fuel remaining: ';FND(F)
 920 IF V<-2 THEN 970

 930 BEEP 300,.3 @ BEEP 350,.1 @ BEEP 35
     0,.6 @ BEEP 310,.1 @ BEEP 300,.1 @
     BEEP 270,.1
 940 BEEP 300,.4 @ BEEP 320,.4 @ BEEP 33
     0,.4 @ BEEP 350,.4 @ BEEP 400,.3 @
     BEEP 475,.1 @ BEEP
 950 BEEP 550,.1 @ BEEP 475,.1 @ BEEP 40
     0,.1 @ BEEP 350,.4
 960 DISP ' You are a qualified astronau
     t.' @ GOTO 1080
 970 IF V<-8 THEN 1000
 980 FOR I=1 TO 5 @ BEEP 600,.2 @ BEEP 3
     00,.2 @ NEXT I
 990 DISP "Well, you're still in one pie
     ce!" @ GOTO 1080
1000 IF V<-15 THEN 1040
1010 FOR I=1 TO 15 @ BEEP 800-I*50,.05 @
     NEXT I
1020 FOR I=1 TO 8 @ BEEP 100,.02 @ BEEP
     50,.02 @ NEXT I
1030 DISP '   You made a new crater !!!'
     @ GOTO 1080
1040 BEEP 100,.5 @ BEEP 100,.5 @ BEEP 10
     0,.1 @ BEEP 100,.5
1050 BEEP 120,.5 @ BEEP 112,.1 @ BEEP 11
     2,.5
1060 BEEP 100,.1 @ BEEP 100,.5 @ BEEP 95
     ,.1 @ BEEP 100,1
1070 DISP '   Next of kin will be notifie
     d.'
1080 WAIT 1
1090 ON ERROR GOTO 1100
1100 INPUT 'Try same conditions again?',
     'Y';Q$
1110 Q$=UPRC$(Q$[1,1])
1120 IF Q$='Y' THEN 240
1130 IF Q$#'N' THEN 1100
1140 ON ERROR GOTO 1150
1150 INPUT 'Try new conditions?','Y';Q$
1160 Q$=UPRC$(Q$[1,1])
1170 IF Q$='N' THEN 1240
1180 IF Q$#'Y' THEN 1150
1190 X=IP(RND*(100+K*100))*10+100
1200 V=-IP(RND*(10+K*10))*5
1210 IF V*V>2*(M-G)*X THEN 1200
```

-Test for velocity at impact to
 determine skill of player

PROGRAM LISTING

```
1220 OFF ERROR
1230 GOTO 290
1240 ON ERROR GOTO 1250
1250 INPUT 'Try a different planet?','Y'
     ;Q$
1260 Q$=UPRC$(Q$[1,1])                    -Check for valid initial
                                           velocity
1270 IF Q$='Y' THEN 110
1280 IF Q$#'N' THEN 1250
1290 OFF ERROR
1300 DISP '          END OF GAME'
1310 DISP
1320 DELAY 1
1330 STOP
```

PROGRAM DESCRIPTION

ECHO

Echo is a memory game in which the player is challenged to repeat a sequence of musical notes. The player may choose from an easy game, consisting of four different notes, a medium game with a six note scale, and a hard game with eight notes. The game starts with one note and gradually adds more notes if the player repeats the sequence correctly. The same sequence of notes is not repeated each time. The notes are chosen at random on each turn. As an aid to recognizing the notes, a bell is displayed on the screen at a position relative to the position of the note in the scale of 4, 6 or 8 notes.

The numbered keys <1> through <8> are used to enter the notes (<1> through <4> for the easy game and <1> through <6> for the medium game).

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		ECHO	
1	Choose game difficulty	Easy, Medium, or Hard game?	E,M, or H [RTN]
	If 'E' then use 4 notes	You will be using 4 notes . . .	
	keys <1> to <4>	◢◣ ◢◣ ◢◣ ◢◣	
	If 'M' then use 6 notes	You will be using 6 notes . . .	
	keys <1> to <6>	◢◣ ◢◣ ◢◣ ◢◣ ◢◣ ◢◣	
	If 'H' then use 8 notes	You will be using 8 notes . . .	
	keys <1> to <8>	◢◣ ◢◣ ◢◣ ◢◣ ◢◣ ◢◣ ◢◣ ◢◣	
2	Listen for 1st note. Bell	Get ready . . .	
	will be displayed at a		
	position relative to the note	◢◣	
3	Try to repeat the note by	Now it's your turn . . .	
	pressing the key that you		<1>,<2>,<3>
	think corresponds to the note		<4>,<5>,<6>, <7> or <8>
3a	If correct then increase	GOOD JOB!!!	
	the number of notes by 1 and		
	goto step 2		
3b	If incorrect then sound a	OOPS!!!	
	low tone		
4	Option to try same number	Try another? Y	N [RTN] or [RTN]
	of notes again. If 'Y' then		
	step 2		
5	Option to try a new game.	Start a new game? Y	N [RTN] or [RTN]
	If 'Y' then step 1 else end	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
N( )	8-note scale	K	Number of key pressed when entering sequence
T( )	Stores the sequence of notes	F	Flag to indicate when a wrong key has been pressed
R	Random number used to generate sequence	B( )	Tab positions
N	Number of notes in the sequence	G$	Game difficulty
I	Looping	K$	Key pressed when entering notes
J	Used as an aid to defining tab positions for ⫞ symbol	Q$	User options
G	Number of different types of notes to be used in building the sequence		

PROGRAM LISTING

```
 10 ! ECHO - A musical
 20 ! memory game.
 30 ! Revision  11/01/82
 40 !
 50 OPTION BASE 1
 60 SHORT T(30),N(8),B(8)
 70 INTEGER R,N,I,J,G,K,F
 80 DIM G$[4],K$[2],Q$[3]
 90 FOR I=0 TO 7
100 N(I+1)=439.99+I*(220/7)
```
—Set up the frequencies for 8 notes

```
110 NEXT I
120 DELAY 1
130 RANDOMIZE
140 DISP TAB(10);CHR$(7);' ';CHR$(7);'
    ECHO ';CHR$(7);' ';CHR$(7)
150 BEEP 450,.3 @ BEEP 500,.3 @ BEEP 40
    0,.3 @ BEEP 200,.3 @ BEEP 300,.8
160 WAIT 1
```
—Introduction - play tune from 'Close Encounters'

```
170 ON ERROR GOTO 180
180 DISP CHR$(197);'asy, ';CHR$(205);'e
    dium, or ';CHR$(200);
190 INPUT 'ard game?',CHR$(95);G$
200 G$=UPRC$(G$[1,1])
210 OFF ERROR
220 DELAY 0
230 ON POS('EMH',G$)+1 GOTO 170,240,350
    ,440
```
—Select game difficulty

```
240 DISP 'You will be using 4 notes...'
    @ WAIT 2 @ DISP
250 J=2
```
—J sets up the tab positions for the bell

```
260 FOR I=1 TO 4
270 B(I)=J
280 DISP TAB(B(I));CHR$(7)
290 BEEP N(I),.2 @ WAIT .5
300 J=J+8
310 NEXT I
320 DISP @ WAIT .5
330 G=4
340 GOTO 540
350 DISP 'You will be using 6 notes...'
    @ WAIT 2 @ DISP
360 FOR I=1 TO 6
370 B(I)=I*5
```
—Bell positions are 5 spaces apart

```
380 DISP TAB(B(I));CHR$(7)
390 BEEP N(I),.2 @ WAIT .5
400 NEXT I
410 DISP @ WAIT .5
420 G=6
430 GOTO 540
440 DISP 'You will be using 8 notes...'
    @ WAIT 2 @ DISP
```

PROGRAM LISTING

```
450  J=1
```
— J sets up the tab positions for the bell

```
460  FOR I=1 TO 8
470  B(I)=J
480  DISP TAB(B(I));CHR$(7)
490  BEEP N(I),.2 @ WAIT .5
500  J=J+4
510  NEXT I
520  DISP @ WAIT .5
530  G=8
540  N=1
550  WAIT 1 @ DISP '        Get ready ..
     '
560  WAIT 1
570  DISP
580  FOR I=1 TO N
590  R=IP(RND*G)+1
```
— Random number between 1 and the number of notes in play

```
600  DISP TAB(B(R));CHR$(7)
610  BEEP N(R),.2 @ WAIT .5
620  T(I)=N(R)
630  NEXT I
640  DISP "Now it's your turn ..." @ WAI
     T .5 @ DISP @ F=0
650  FOR I=1 TO N
660  ON ERROR GOTO 670
670  K$=KEY$
680  IF VAL(K$)<1 OR VAL(K$)>G THEN 670
```
— Check for valid key input

```
690  K=VAL(K$)
700  DISP TAB(B(K));CHR$(7)
710  IF T(I)=N(K) THEN BEEP T(I),.2 ELSE
     BEEP 100,.2 @ F=1 @ I=N
```
— Check for correct response to sequence

```
720  NEXT I
730  IF F=1 THEN 790 ELSE N=N+1
740  ON IP(RND*4)+1 GOTO 750,760,770,780
750  DISP TAB(11);'VERY GOOD!' @ WAIT 1
     @ GOTO 550
760  DISP TAB(11);'EXCELLENT!' @ WAIT 1
     @ GOTO 550
770  DISP TAB(11);'WELL DONE!!' @ WAIT 1
      @ GOTO 550
780  DISP TAB(11);'GOOD JOB!!!' @ WAIT 1
     @ GOTO 550
790  DISP TAB(10);'O O P S  !!!'
800  WAIT 1
810  ON ERROR GOTO 820
820  INPUT 'Try another?','Y';Q$
830  Q$=UPRC$(Q$[1,1])
840  IF Q$='N' THEN 880
850  IF Q$#'Y' THEN 820
860  OFF ERROR
870  GOTO 550
880  ON ERROR GOTO 890
890  INPUT 'Start a new game?','Y';Q$
```

PROGRAM LISTING

```
900  Q$=UPRC$(Q$[1,1])
910  DELAY 1
920  IF Q$='N' THEN 960
930  IF Q$#'Y' THEN 890
940  OFF ERROR
950  GOTO 170
960  DISP TAB(10);'END OF GAME' @ WAIT 2
     @ DISP
970  STOP
```

PROGRAM DESCRIPTION

BLACKJACK

In this card game, the HP-75 is the dealer and up to eight people may play.
Each player begins with $200 and may bet as much or all of it as he wishes
(in $10 increments) up to $200. Before the cards are dealt, each player
enters his bet (to leave the game, enter 0 for a bet). Then the dealer
deals the cards from a pack of 104 (double deck). If the dealer is holding
a natural, he immediately collects the bets of all players who do not have
naturals. (A natural is an ace and any face card or ten, giving a count of
21 in two cards. A natural is also called Blackjack). If the dealer and
any other player both have naturals, the bet is a standoff and the player
keeps his money. If the dealer does not have a natural, the game continues
with each player deciding what he will do with his hand. Aces may be
counted as 1 or 11, as the player wishes. All face cards count as 10
(K, Q and J) and any other cards count as their pip value (numerical value),
i.e., "T" counts 10, 9 counts as 9, 8 counts as 8, and so on.

Each player has the option to Hit (H), Double (D), Split (/), or Stand (S).
A hit means he would like another card to add to the ones he already has. A
double (down-for-double) means he wants only one more card and he is doubling
his original bet. This is usually done if the player has a total of 11 al-
ready in his hand, taking a chance on getting a ten or face card to total 21.
A split means the player has two identical cards, i.e., two queens, or two
fives, etc., and would like to separate them into two different hands draw-
ing two more cards to go with the two like cards. The bet that the player
originally made is then placed also on the second hand. In other words, he
is now playing for twice as much money. The player must play each hand
separately, finishing his 1st hand before going to the 2nd. A stand means
the player is satisfied with the cards he has.

When all the players are done, the dealer settles his hand. If his total
is 17 or more he must stand. If his total is less, he must take a card and
must continue to take cards until his total is 17 or more, at which point
he must stand. If dealer has an ace, and counting it as 11 would bring his
total to 17 or more (but not over 21), he must count the ace as 11 and stand.

When the dealer is done, all bets are settled. Any player with a natural
wins 1½ times his bet. Any player whose total went over 21 loses his bet.
If the dealer went over 21, he pays to each player under 21 the amount of
his bet. If the dealer stands at 21 or less, the pays the bet of any
player having a higher total (not over 21), collects the bet of any player
having a lower total, and is at a stand-off ("PUSH") with any player who
has the same total.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		BLACKJACK	
1	Enter number of players.	Number of players?	1-8 [RTN]
		SHUFFLING...Please wait	
2	Enter bets in increments of	Place your bets?	
	$10.00. If a players enters	Player #n $	bet [RTN]
	0 then his total winnings are	Player #n $	bet [RTN]
	displayed. If all players	Player #n $	bet [RTN]
	enter 0, then the game ends	Player #n $	bet [RTN]
	(Goto step 14).	⋮	⋮
2a	Deal cards - if dealer has		
	natural then step 13		
3	Display number of current		
	player	Player #n	
4	Display dealer's face-up card,		
	and all cards the player holds	d░cc:	
	Enter option:		H or
	If 'H' then goto step 5		D or
	If 'D' then goto step 6		/ or
	If '/' then goto step 7		S then
	If 'S' then goto step 8		[RTN]
5	HIT. Take another card from	n nnn:	
	the deck. If total is over		
	21 then step 9 else step 4.		
6	DOUBLE. Take one card and	n nnn	
	double the amount of the		

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	original bet. If cannot		
	double bet, then display:	You don't have enough money	
	and goto step 4, else step 9.		
7	SPLIT. If player doesn't have		
	enough money, then display:	Cannot split	
	and goto step 4 else:		
	Play 1st hand using one of	Play first hand	
	the identical cards. Use		
	same options as in step 4		
	except cannot split again.		
	Then play 2nd hand. After	Play second hand	
	each hand goto step 9.		
8	STAND. Take no more cards.		
	If all players have played,		
	goto step 10. Goto step 3		
	for next player.		
9	Display dealer's face up card,		
	all cards the player holds,		
	and 'BUST!'. If all players	d cc...BUST!	
	have played, step 10 else		
	step 3.		
10	After all players have played		
	dealer plays his hand. Dis-		
	play dealer's cards and his		
	total points.	dd..=tt	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
11	SETTLE BETS. If a player has		
	busted, then he loses (a). If	(a) Player #n LOSES $(bet) $(total)	
	he has a total greater than		
	the dealer's and less than or		
	equal to 21, then he wins (b).	(b) Player #n WINS $(bet) $(total)	
	If he has a natural, he wins		
	1½ times his bet (c). If the	(c) Player #n has BLACKJACK	
	player's total is less than	WIN $(bet) * 1.5 Total $(total)	
	the dealer's (dealer 21) then		
	he loses (a). If the player		
	and dealer tie at or below 21,		
	the player keeps his bet (d).	(d) Player #n PUSH $(total)	
	If the player's hand was		
	split, his winnings (or losses)		
	are totalled. Should he end		
	up even (e) will be displayed.	(e) Player #n EVEN $(total)	
11a	If a player loses all of his		
	money, then he must quit.	Player #n is out	
12	After settling bets, goto		
	step 2.		
13	DEALER NATURAL. If dealer has		
	21, then settle all bets im-		
	mediately. Goto step 11.	**DEALER HAS NATURAL:cc	
14	END OF GAME. Option to run a-		
	gain. If 'Y' then step 1, else	Would you like to play again?	
	end.	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
N	Number of players	F2	Used to convert 'STAND' routine to a scoring routine to check for 'busted' hands
H(,)	Players hands and totals		
J,I	Looping, index	P	Player counter
K1	Counts cards used in deck	Z	Loop to shuffle cards
D( )	Dealers hand and total	Z1	Random number used in shuffling
T1,T2	Index used for manipulating and evaluating hands	F	Flag to indicate all players bet $0 - end game
B( )	Bets	F5	Flag to show all players out of money
T( )	Total cash	F( )	Indicate if a player has split his hand
N( )	Flags for naturals	B	Total gain or loss on split hands
C	Number of cards in a hand	A1	Indicates hand has an ace
A	Value of cards in a hand	D$	Shuffled deck
F1	Indicates split - used to convert parts of programs to subroutines	T$	Cards used to form deck
		Q$	User interaction - options on hands
D1$	Used to shuffle deck - temporary storage		

NOTES AND REFERENCES

Note: 1.   This game uses a double-deck (104 cards). A deck of 52
cards consists of 4 suits (hearts, clubs, diamonds, spades)
each having 13 cards: Ace (A), king (K), queen (Q), jack
(J), 10 (represented in this game by 'T'), 9, 8, 7, 6, 5,
4, 3, and 2. Therefore, in a double-deck, there will be
8 aces, 8 kings, 8 queens, etc.

References: 1.   "BLJACK", HP-2000 BASIC, Hewlett-Packard.

2.   "OFFICIAL RULES OF CARD GAMES", The United States Playing
Card Company, c. 1969, pp. 228-230.

PROGRAM LISTING

```
 10 ! BLACKJ
 20 !
 30 ! Revision 11/01/82
 40 INTEGER N,H(16,12),J,I,K1,D(10),T1,
    T2,B(8),T(8),N(9),C,A,F1,F2,P,Z,Z1,
    I1,J1,F,F5,F(8),B,A1
 50 DIM T$[13],D$[104],Q$[3],D1$[1]
 60 RANDOMIZE
 70 T$='A23456789TJQK'
 80 DELAY 3
 90 K1=1
100 F=0
110 DISP TAB(8);CHR$(7);' ';CHR$(7);' B
    LACKJACK ';CHR$(7);' ';CHR$(7)
120 ON ERROR GOTO 130
130 INPUT 'Number of players?',CHR$(95)
    ;N
140 IF N<1 OR N>8 THEN 130
150 OFF ERROR
160 D$=T$&T$&T$&T$&T$&T$&T$&T$
170 ! SHUFFLE CARDS.
180 GOSUB 2670
190 FOR I=1 TO N
200 T(I)=200                                -Players begin with $200
210 NEXT I
220 FOR I=1 TO 16
230 FOR J=1 TO 12
240 H(I,J)=0
250 NEXT J
260 NEXT I
270 FOR I=1 TO N
280 F(1)=0 @ B(I)=0 @ N(I)=0
290 NEXT I
300 N(9)=0
310 FOR I=1 TO 10
320 D(I)=0
330 NEXT I
340 ! PLACE BETS.
350 F5=0 @ F=0
360 FOR I=1 TO N
370 IF T(I)#0 THEN I=N @ F5=1            -Check to see if player is out
                                          of money
380 NEXT I
390 IF F5=0 THEN 2910
400 DISP '        Place your bets !!!'
410 IF N#1 THEN 480                      -Check for only one player
420 ON ERROR GOTO 430
430 INPUT '       $',CHR$(95);B(1)
440 IF B(1)=0 THEN DISP 'Total $';T(1)   -Quit if bet is zero
    @ GOTO 2910
450 IF B(1)<10 OR B(1)>200 OR B(1)/10#I  -Check for valid bet between
    P(B(1)/10) THEN 430                    $10 and $200
460 IF B(1)>T(1) THEN DISP '     You on
    ly have $';T(1) @ GOTO 430
```

PROGRAM LISTING

```
470 GOTO 580
480 FOR I=1 TO N
490 IF T(I)=0 THEN 570                    -Check to see if player is out
                                           of money
500 ON ERROR GOTO 510
510 DISP 'Player #';I;
520 INPUT '   $',CHR$(95);B(I)
530 OFF ERROR
540 IF B(I)=0 THEN DISP 'Total $';T(I)    -Quit if bet is zero
    @ T(I)=0 @ GOTO 570
550 IF B(I)<10 OR B(I)>200 OR B(I)/10#I   -Check for valid bet between
    P(B(I)/10) THEN 500                     $10 and $200
560 IF B(I)>T(I) THEN DISP '     You on
    ly have $';T(I) @ GOTO 500
570 NEXT I
580 OFF ERROR
590 DISP
600 FOR I=1 TO N
610 IF B(I)#0 THEN I=N @ F=1               -Check for all players wanting
                                            to quit
620 NEXT I
630 IF F=1 THEN 650
640 GOTO 2910
650 ! DEAL CARDS.
660 FOR I=1 TO N+1
670 N(I)=0
680 NEXT I
690 D(10)=0
700 FOR I=1 TO N
710 IF T(I)=0 THEN 780                     -Check to see if player is out
                                           of money
720 IF K1=104 THEN GOSUB 2670
730 K1=K1+1
740 H(I,1)=NUM(D$[K1,K1])                  -Deal card
750 IF K1=104 THEN GOSUB 2670             -Check to see if we need to
                                           reshuffle the deck
760 K1=K1+1
770 H(I,2)=NUM(D$[K1,K1])
780 NEXT I
790 ! DEALERS HAND.
800 IF K1=104 THEN GOSUB 2670
810 K1=K1+1
820 D(1)=NUM(D$[K1,K1])
830 IF K1=104 THEN GOSUB 2670
840 K1=K1+1
850 D(2)=NUM(D$[K1,K1])
860 ! CHECK FOR DEALER NATURAL.
870 T1=1 @ T2=2
880 IF D(T1)#65 THEN 900                   -Check for ace for blackjack
890 IF D(T2)=84 OR D(T2)=74 OR D(T2)=81
    OR D(T2)=75 THEN N(N+1)=1
900 IF T1=1 THEN T1=2 @ T2=1 @ GOTO 880
910 ! CHECK FOR PLAYER NATURAL.
920 FOR I=1 TO N
```

PROGRAM LISTING

```
 930 IF T(I)=0 THEN 990                    -Check to see if a player is
                                            out of money

 940 T1=1 @ T2=2
 950 IF H(I,T1)#65 THEN 980                -Check for ace for blackjack
 960 IF H(I,T2)=84 OR H(I,T2)=74 OR H(I,
     T2)=81 OR H(I,T2)=75 THEN N(I)=1
 970 GOTO 990
 980 IF T1=1 THEN T1=2 @ T2=1 @ GOTO 950
 990 NEXT I
1000 ! SETTLE DEALER NATURAL.
1010 IF N(N+1)#1 THEN 1160
1020 FOR I=1 TO 3 @ BEEP 500 @ BEEP 600
     @ NEXT I
1030 DISP 'DEALER HAS NATURAL:  ';CHR$(D  -Display dealers hand for
     (1));' ';CHR$(D(2))                   blackjack (natural)
1040 FOR I=1 TO N
1050 IF T(I)=0 THEN 1120
1060 IF N(I)#1 THEN 1090
1070 DISP USING 1130 ; I,CHR$(H(I,1)),CH  -Display players hand for
     R$(H(I,2)),T(I) @ GOSUB 2880          natural (keep bet)
1080 GOTO 1120
1090 T(I)=T(I)-B(I)
1100 DISP USING 1140 ; I,CHR$(H(I,1)),CH  -Display players hand for
     R$(H(I,2)),B(I),MAX(0,T(I)) @ GOSUB   non-natural (lose bet)
     2880
1110 IF T(I)=0 THEN DISP 'Player #';I;'i
     s out.'
1120 NEXT I
1130 IMAGE 'Player#',d,': ',2a,2a,'EVEN'
     ,7x,'$',ddddd
1140 IMAGE 'Player#',d,': ',2a,2a,'LOSE
     $',3d,2x,'$',ddddd
1150 GOTO 400
1160 ! REGULAR PLAY.
1170 FOR P=1 TO N
1180 I=P @ F1=0 @ F2=0
1190 IF T(I)=0 THEN 1990
1200 IF N=1 THEN 1220
1210 DISP '             PLAYER #';I
1220 C=2
1230 ON ERROR GOTO 1240
1240 GOSUB 2790
1250 ON POS('SHD/',Q$)+1 GOTO 1240,1260,  -Options- Stand, Hit, Double or
     1420,1530,1660                        Split(/)
1260 ! STAND.
1270 H(I,12)=0
1280 A1=0
1290 FOR J=1 TO C
1300 IF H(I,J)>=50 AND H(I,J)<=57 THEN A  -Check for regular numeric card
     =VAL(CHR$(H(I,J))) @ GOTO 1340
1310 IF H(I,J)#65 THEN A=10 @ GOTO 1340   -Check for face card
1320 IF A1=1 THEN A=1 @ GOTO 1340         -Check for previous aces
1330 A1=1 @ GOTO 1350
1340 H(I,12)=H(I,12)+A
```

PROGRAM LISTING

```
1350 NEXT J
1360 IF A1=0 THEN 1380                     -Check for previous aces
1370 IF H(I,12)>10 THEN H(I,12)=H(I,12)+
     1 ELSE H(I,12)=H(I,12)+11
1380 IF H(I,12)>21 THEN DISP 'BUST !'      -Check for players total
                                            exceeding 21
1390 IF F1=0 AND F2=0 THEN GOTO 1990       -Check to see if stand routine
                                            is used as subroutine
1400 IF H(I,12)>21 AND F1=0 AND F2=1 THE
     N POP @ GOTO 1990
1410 RETURN
1420 ! HIT.
1430 C=C+1
1440 IF K1=104 THEN GOSUB 2670
1450 K1=K1+1
1460 H(I,C)=NUM(D$[K1,K1])
1470 F2=1                                  -Set up flag to indicate use of
                                            routines as subroutines
1480 GOSUB 2790
1490 GOSUB 1260
1500 F2=0
1510 IF F1=1 THEN DISP @ RETURN            -Check to see if routine is for
                                            SPLIT
1520 GOSUB 2850 @ GOTO 1250
1530 ! DOWN FOR DOUBLE.
1540 IF B(I)*2<=T(I) THEN 1560             -CHeck to see if player has
                                            enough money to double bet
1550 BEEP @ DISP "  You don't have enoug
     h money." @ GOTO 1240
1560 B(I)=B(I)*2
1570 C=C+1
1580 IF K1=104 THEN GOSUB 2670
1590 K1=K1+1
1600 H(I,C)=NUM(D$[K1,K1])
1610 F2=1
1620 GOSUB 2790
1630 GOSUB 1260
1640 F2=0 @ DISP
1650 IF F1=1 THEN RETURN ELSE GOTO 1990    -Check to see if routine is for
                                            split
1660 ! SPLIT.
1670 IF H(I,1)#H(I,2) OR B(I)*2>T(I) THE   -Check for valid split and if
     N DISP TAB(10);'CANNOT SPLIT' @ GOT    player has enough money
     O 1240
1680 F(I)=1                                -Set flag to show player split
                                            hand
1690 H(I+8,1)=H(I,2)
1700 DISP '           PLAY FIRST HAND'
1710 F1=1                                  -Set up flag to indicate split
                                            subroutine
1720 IF K1=104 THEN GOSUB 2670
1730 K1=K1+1
1740 H(1,2)=NUM(D$[K1,K1])
1750 C=2
```

PROGRAM LISTING

```
1760 ON ERROR GOTO 1770
1770 GOSUB 2790
1780 ON POS('SHD',Q$)+1 GOTO 1770,1790,1
     910,1950
1790 ! STAND ON SPLIT.
1800 OFF ERROR
1810 F2=1
1820 GOSUB 1260
1830 IF F1=0 THEN GOTO 1990
1840 DISP '            PLAY SECOND HAND'
1850 F1=0 @ F2=0 @ I=I+8
1860 IF K1=104 THEN GOSUB 2670
1870 K1=K1+1
1880 H(I,2)=NUM(D$[K1,K1])
1890 C=2
1900 GOTO 1770
1910 ! HIT ON SPLIT.
1920 OFF ERROR
1930 GOSUB 1420
1940 IF H(I,12)>21 THEN 1830 ELSE GOTO 1
     770
1950 ! DOUBLE ON SPLIT.
1960 OFF ERROR
1970 GOSUB 1530
1980 GOTO 1830
1990 ! NEXT PLAYER.
2000 NEXT P
2010 ! DEALERS HAND.
2020 J=0

2030 FOR I=1 TO N
2040 IF H(I,12)<=21 THEN J=1 @ I=N @ GOT
     O 2060
2050 IF F(I)=1 AND H(I+8,12)<=21 THEN J=
     1 @ I=N
2060 NEXT I
2070 IF J=0 THEN D(10)=21 @ GOTO 2270
2080 IF D(1)=65 THEN D(1)=D(2) @ D(2)=65
2090 DISP CHR$(D(1));' ';CHR$(D(2));' ';
     @ WAIT 1
2100 C=2
2110 T1=1 @ A=0
2120 IF D(T1)>=50 AND D(T1)<=57 THEN A=V
     AL(CHR$(D(T1))) @ GOTO 2170
2130 IF D(T1)#65 THEN A=10 @ GOTO 2170
2140 IF T1=1 THEN A=1 @ GOTO 2170
2150 IF T1=C AND D(10)+11>21 THEN A=1 @
     GOTO 2170
2160 A=11
2170 D(10)=D(10)+A
2180 IF T1#C THEN T1=T1+1 @ GOTO 2120
2190 IF D(10)>=17 THEN DISP '   =';D(10)
     @ GOTO 2270
2200 T1=T1+1 @ C=C+1
```

-Options for split hands-
Stand, Hit or Double

-Arrange dealers hand so ace is furthest right

-Check for regular numeric card

-Check for face card

-Check for dealer's total 17 or over

PROGRAM LISTING

```
2210 IF K1=104 THEN GOSUB 2670
2220 K1=K1+1
2230 D(C)=NUM(D$[K1,K1])
2240 DISP CHR$(D(T1));' '; @ WAIT 1
2250 IF D(C-1)=65 THEN D(C-1)=D(C) @ D(C
     )=65 @ D(10)=0 @ GOTO 2110
2260 GOTO 2120
2270 ! SETTLE BETS.
2280 FOR I=1 TO N
2290 IF T(I)=0 THEN 2590
2300 IF H(I,12)>21 AND F(I)=0 THEN 2550
2310 IF N(I)#1 THEN 2360
2320 T(I)=T(I)+1.5*B(I)
```

-BLACKJACK - player wins 1.5 times his bet

```
2330 DISP 'Player #';I;'has BLACKJACK.'
2340 DISP 'Win $';1.5*B(I);'        Total
     $';T(I)
2350 GOTO 2590
2360 IF F(I)=0 THEN 2470
```

-Check for split hand- if so assess each hand

```
2370 J=0 @ B=0
2380 IF H(I+J,12)>21 THEN B=B-B(I) @ GOT
     O 2430
2390 IF D(10)>21 THEN 2420
2400 IF H(I+J,12)<D(10) THEN B=B-B(I) @
     GOTO 2430
2410 IF H(I+J,12)>D(10) THEN B=B+B(I) @
     GOTO 2430
2420 IF H(I+J,12)<=21 THEN B=B+B(I) ELSE
     B=B-B(I)
2430 IF J=0 THEN J=8 @ GOTO 2380
2440 B(I)=B @ IF B>0 THEN 2500
2450 IF B<0 THEN B(I)=-B(I) @ GOTO 2550
2460 DISP 'Player #';I;' EVEN';'
     $';T(I) @ GOTO 2590
```

-Player ended up with even money from split hand

```
2470 IF H(I,12)>21 THEN 2550
2480 IF H(I,12)<D(10) AND D(10)<=21 THEN
     2550
2490 IF H(I,12)=D(10) THEN 2530
2500 T(I)=T(I)+B(I)
2510 DISP 'Player #';I;'WINS $';B(I);'
     $';T(I)
2520 GOTO 2590
2530 DISP 'Player #';I;'PUSH';'
     $';T(I)
2540 GOTO 2590
2550 T(I)=T(I)-B(I)
2560 DISP 'Player #';I;'LOSES $';B(I);'
     $';T(I)
```

-Players with less than $10 drop out, as min. bet is $10

```
2570 IF T(I)<10 THEN T(I)=0
2580 IF T(I)=0 THEN DISP 'Player #';I;'i
     s out.'
```

-Check to see if a player is out of money

```
2590 NEXT I
2600 J=0
```

PROGRAM LISTING

```
2610 FOR I=1 TO N
2620 IF T(I)=0 THEN 2640
2630 J=1
2640 NEXT I
2650 IF J=0 THEN 2910
2660 GOTO 220
2670 ! SHUFFLE DECK.
2680 DELAY 0
2690 DISP '      SHUFFLING ... Please wait
     .'
2700 FOR Z=1 TO 104
2710 Z1=IP(104*RND+1)
2720 D1$=D$[Z,Z]
2730 D$[Z,Z]=D$[Z1,Z1]
2740 D$[Z1,Z1]=D1$
2750 NEXT Z
2760 DELAY 3
2770 K1=0
2780 RETURN
2790 ! DISPLAY HAND.
2800 DISP CHR$(D(1));CHR$(31);'  ';
2810 FOR J1=1 TO C
2820 DISP CHR$(H(I,J1));' ';
2830 NEXT J1
2840 IF F2=1 THEN WAIT 1 @ GOTO 2870
2850 INPUT ':',CHR$(95);Q$
2860 Q$=UPRC$(Q$[1,1])
2870 RETURN
2880 !
2890 IF NUM(KEY$)#13 THEN 2890
2900 RETURN
2910 !
2920 ON ERROR GOTO 2930
2930 INPUT 'Would you like to play again
     ?';Q$
2940 Q$=UPRC$(Q$[1,1])
2950 IF Q$='Y' THEN 120
2960 IF Q$#'N' THEN 2930
2970 OFF ERROR
2980 DISP TAB(9);'END OF GAME' @ DISP
2990 DELAY 1
3000 STOP
```

-Check to see if routine is used as subroutine

-Wait for 'RTN' key to be pressed

PROGRAM DESCRIPTION

WORD SCRAMBLE

In this game, the player tries to form a single word form the group of let-
ters that appears on the screen in as short a time as he can. The player
may choose from easy, medium, or hard words, the lengths of these words
being 5, 6, and 7 letters, respectively. The player may take as much time
as he wants to find the word, but his score will decrease with time. For
easy (5 letters) words, the player must answer in less than 15 seconds to
receive a score of 1 or more. If he takes longer than 15 seconds, his
score will be 0 for that word. For medium words, the time "limit" for
scoring is 20 seconds, and for hard words, 25 seconds. If the player
enters the wrong answer, he will be shown the correct word and then con-
tinue on to the next word.

After the player has tried five words, he will be asked if he wants to con-
tinue and perhaps try harder or easier words.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		WORD SCRAMBLE	
1	Select degree of difficulty:	Easy, Medium or Hard?	E, M, or H [RTN]
2	If 'E' then use 5-letter words;		
	time "limit" for scoring is		
	15 seconds: if 'M' then use 6-		
	letter words; time limit for		
	scoring is 20 seconds: if 'H'		
	then use 7-letter words; time		
	limit for scoring is 25 seconds.		
2	Display word number and the		
	scrambled word:	n : AAAAAA = ▓	
3	Enter word:		aaaaaa [RTN]
4	If answer incorrect display:	The correct answer is AAAAAA	
	then goto step 2		
5	Display solution time:	Solution time xx seconds	
6	Display score for word n and		
	the total score	Score = s *** Total = +	
7	After playing 5 words:	Do you wish to continue?Y	[RTN] or N [RTN]
	else goto step 2 for next word.		
7a	If 'Y' then step 1 else:	Start a new game?Y	[RTN] or N [RTN]
7b	If 'Y' then set scores to zero		
	and goto step 1 else	END OF GAME	

NOTE: AAAAAA represents a word with 5 to 7 letters
        n represents the number of the word
        s represents the score for word n (s = time "limit" - solution time)
        + represents the total score up to and including word n

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
I	Looping	Q$	User interaction - difficulty chosen
C	Counts number of words	W$	Scrambled word
S	Counts number of correct responses	W1$	Single letter of word - used for temporary storage in scrambling routine
N	Number of letters in word		
T1	Beginning time for word	A$	Correct word
T2	Ending time for word	S$	Players solution
S1	Total accumulative score	Z$	User interaction
D	Score for a word	L$	Time "limits"
W1	Random number used to mix letters in a word		

NOTES AND REFERENCES

Note: There are 80 5-letter words, 70 6-letter words and 60 7-letter words, none of which are plural forms ending in 's'. For example: LAHLS would be SHALL and not HALLS. All the words were taken from The Merriam-Webster Dictionary (1974), and all are single, unabbreviated non-hyphenated words.

Reference: "WORD", HP-2000 - HP BASIC PROGRAM LIBRARY, Hewlett-Packard.

PROGRAM LISTING

```
 10 ! SCRAMBLE - unscramble the
 20 ! word.
 30 !
 40 ! Revision  11/01/82
 50 !
 60 DELAY 1
 70 RANDOMIZE
 80 INTEGER I,C,S,N,T1,T2,S1,D,W1
 90 DIM Q$[4],W$[8],W1$[1],A$[8],S$[10]
    ,Z$[3],L$[2]
100 DISP '             WORD SCRAMBLE' @ WA
    IT 2
110 C,S,S1=0
120 ON ERROR GOTO 130
130 DISP CHR$(197);'asy, ';CHR$(205);'e      -Easy=5 letters, Medium=6,
    dium, or ';CHR$(200);                     Hard=7
140 INPUT 'ard?'; Q$ @ Q$=UPRC$(Q$[1,1]
    )
150 IF Q$#'E' AND Q$#'M' AND Q$#'H' THE
    N 130
160 IF Q$='E' THEN L$='15' @ GOTO 180    -Time limits are 15,20 and 25
                                          seconds
170 IF Q$='M' THEN L$='20' ELSE L$='25'
180 ON POS('EMH',Q$)+1 GOTO 120,190,350
    ,450
190 N=5
200 ON IP(RND*8)+1 GOTO 210,220,230,240  -Move data pointer to random
    ,250,260,270,280                      data stmt for each word set
210 RESTORE 960 @ GOTO 290
220 RESTORE 970 @ GOTO 290
230 RESTORE 980 @ GOTO 290
240 RESTORE 990 @ GOTO 290
250 RESTORE 1000 @ GOTO 290
260 RESTORE 1010 @ GOTO 290
270 RESTORE 1020 @ GOTO 290
280 RESTORE 1030
290 FOR I=1 TO IP(RND*10)+1               -There are 10 words per data
                                           statement
300 READ A$[1,N]
310 NEXT I
320 W$=A$[1,N]
330 GOSUB 860
340 GOTO 540
350 N=6
360 ON IP(RND*7)+1 GOTO 370,380,390,400
    ,410,420,430
370 RESTORE 1050 @ GOTO 440
380 RESTORE 1060 @ GOTO 440
390 RESTORE 1070 @ GOTO 440
400 RESTORE 1080 @ GOTO 440
410 RESTORE 1090 @ GOTO 440
420 RESTORE 1100 @ GOTO 440
430 RESTORE 1110
440 GOTO 290
```

PROGRAM LISTING

```
450  N=7
460  ON IP(RND*6)+1 GOTO 470,480,490,500
     ,510,520
470  RESTORE 1130 @ GOTO 530
480  RESTORE 1140 @ GOTO 530
490  RESTORE 1150 @ GOTO 530
500  RESTORE 1160 @ GOTO 530
510  RESTORE 1170 @ GOTO 530
520  RESTORE 1180
530  GOTO 290
540  C=C+1                              -Count the number of words
550  ON ERROR GOTO 570
560  T1=TIME
570  DISP C;': ';W$[1,N];
580  INPUT ' =';S$
590  T2=TIME @ IF T2<=T1 THEN T2=T2+8639 -Adjust for midnight going by
     9                                    while playing game
600  S$=UPRC$(S$[1,N])
610  OFF ERROR
620  IF S$[1,N]#A$[1,N] THEN 700         -Check for correct response
630  S=S+1
640  BEEP @ BEEP @ BEEP
650  DISP 'Solution time';T2-T1;'seconds
     .' @ WAIT 1
660  D=MAX(0,VAL(L$)-(T2-T1))            -Compute score. Score is zero
                                          if time limit exceeded
670  S1=S1+D
680  DISP 'Score =';D;' *** Total =';S1
     @ WAIT 2
690  GOTO 720
700  BEEP 100 @ BEEP 100 @ BEEP 100
710  DISP 'The correct answer is ';A$[1,
     N] @ WAIT 2
720  IF IP(C/5)#C/5 THEN 180            -Have five consecutive words
                                         been tried?
730  ON ERROR GOTO 740
740  INPUT 'Do you wish to continue? ','
     Y';Z$
750  Z$=UPRC$(Z$[1,1])
760  IF Z$='Y' THEN 120
770  IF Z$#'N' THEN 740
780  ON ERROR GOTO 790
790  INPUT 'Start a new game? ','Y';Z$
800  Z$=UPRC$(Z$[1,1])
810  IF Z$='Y' THEN 110
820  IF Z$#'N' THEN 790
830  OFF ERROR
840  DISP '            END OF GAME' @ WAI
     T 2 @ DISP
850  STOP
860  ! SCRAMBLE WORD.
870  FOR I=N TO 2 STEP -1
880  W1=IP(I*RND+1)                     -Random number used to scramble
                                         the letters in the word
```

PROGRAM LISTING

```
 890  W1$=W$[I,I]
 900  W$[I,I]=W$[W1,W1]
 910  W$[W1,W1]=W1$
 920  NEXT I
 930  IF W$[1,N]=A$[1,N] THEN GOTO 860
```

—Check to see if word is scrambled

```
 940  RETURN
 950  ! EASY WORDS.
 960  DATA ALBUM,ABOUT,AWARE,BEACH,BUGLE,
      BASIC,CANDY,CHAIN,COBRA,DITTO
 970  DATA DWELL,DEPOT,EVENT,ENTER,EBONY,
      FENCE,FLING,FRUIT,GORGE,GROUT
 980  DATA GOOSE,HEATH,HANDY,HOIST,INDEX,
      IRONY,IDEAL,JUICE,KNEAD,KOALA
 990  DATA LURID,LEGAL,MAPLE,MOUSE,MAYBE,
      NYLON,NOVEL,OTTER,OWLET,PASTA
1000  DATA POLKA,POSSE,PLANT,PECAN,PULSE,
      RUMOR,ROSIN,ROUGH,RATIO,ROAST
1010  DATA STYLE,SWELL,SCENE,SAUNA,SHREW,
      SQUAT,SPOOK,SAUTE,TEPEE,THUMB
1020  DATA TUMOR,THYME,TOTAL,TIMID,UNDER,
      USHER,VOICE,VALUE,VALVE,VINYL
1030  DATA WHIRL,WATCH,WEDGE,WORLD,WRONG,
      WIDOW,YACHT,YIELD,YOUTH,ZEBRA
1040  ! MEDIUM WORDS.
1050  DATA ABACUS,ASSESS,ADHERE,ALMOST,AN
      SWER,BABOON,BOGGLE,BONBON,BRUISE,BY
      PASS
1060  DATA BLUING,BLIGHT,CHOICE,CRISIS,CU
      RFEW,CANINE,CATTLE,DEBRIS,DEVOUR,DO
      MINO
1070  DATA DIMPLE,EFFORT,EITHER,ENAMEL,FA
      STEN,FLIGHT,FLUENT,FUNNEL,GARAGE,GO
      SSIP
1080  DATA GOVERN,HARBOR,HUMBLE,INDOOR,IR
      ONIC,ISLAND,JIGSAW,JUNIOR,KIDNEY,KN
      IGHT
1090  DATA MAMMAL,MILDEW,MOBILE,MURMUR,NU
      TMEG,NITWIT,NAUSEA,OODLES,ORCHID,OM
      ELET
1100  DATA OUTLET,POWDER,PARADE,PATROL,PE
      WTER,POLITE,PRISON,QUAINT,QUARTZ,RA
      GLAN
1110  DATA RATTAN,REALTY,RIBBON,RUSTIC,SQ
      UIRM,SQUASH,STUDIO,TRIVIA,TIPTOE,UN
      PACK
1120  ! HARD WORDS.
1130  DATA ALMANAC,ANATOMY,ATTEMPT,BETWEE
      N,BOLOGNA,BROILER,CENTURY,CAPITAL,C
      OLLEGE,DOLPHIN
1140  DATA DEPOSIT,DROUGHT,ECOLOGY,EXHIBI
      T,EYEBALL,FOREIGN,FIFTEEN,FURNACE,G
      RAPHIC,GUMDROP
1150  DATA GARBAGE,HARPOON,HOODLUM,HILLTO
      P,INQUIRE,ISTHMUS,ILLEGAL,JUSTICE,K
      ETCHUP,KNUCKLE
```

PROGRAM LISTING

```
1160 DATA LECTURE,LADYBUG,LULLABY,MATADO
     R,MONSTER,MUFFLER,NATURAL,NOTHING,O
     BSCENE,OCTOPUS
1170 DATA PAPOOSE,PHANTOM,PROTEIN,QUARRE
     L,QUALIFY,RACCOON,REDWOOD,REVENGE,R
     OYALTY,SAWMILL
1180 DATA SHAMPOO,SOLDIER,SUPREME,TYPICA
     L,TERRIFY,UTILITY,UNLUCKY,VACCINE,W
     ELCOME,WHISPER
```

GAMES I

ADVENTURE
ROCKET LANDER
ECHO
BLACKJACK
WORD SCRAMBLE

ALL HP-75 SOLUTIONS BOOKS ARE AVAILABLE RECORDED ON MINI-DATA CASSETTES FROM EITHER A HEWLETT-PACKARD DEALER OR THE HP USERS' LIBRARY.