7500167

PROGRAM DESCRIPTION

Sent Aug '83 by Michael Markov

Program Title HP-41 CASSETTE DATA FILE TRANSFER TO HP-75 DATA

Contributor Harry Phinney

Address Hewlett-Packard, 1000 NE Circle Blvd.

City Corvallis State OR Country USA

Telephone (503) 757-2000 Zip/Postal Code 97330

Program Description (include equations) This program uses the SENDIO and ENTIO$ commands to read a cassette data file created by the WRTR or WRTRX functions of an HP-41. The program then transforms each register (8 bytes) into the HP-75's internal representation for real numbers, or strings, (depending on what was in the register) and places the result in a BASIC data file in the HP-75. A short LEX file is used to transform an 8 byte string into an HP-75 real number. The decoding of ALPHA data is done in BASIC.

The program prompts for the HP-IL assigned device code of the cassette drive.

Necessary Accessories HP-82161A Cassette drive. HP-75 I/O Utilities Solutions Book

Operating limits and warnings The LEX function HP41" DTA will accept any string as an input. A random string input will often result in the display of a non-normalized number.

Minimum RAM Requirement

References HP-83/HP-85 Assembler ROM Manual, HP-82161A Cassette Drive Manual

The HP-IL SYSTEM by Kane, Harper, Ushijima Pub. by Osborne/Mcgraw Hill

VARIABLE DEFINITIONS

NAME	DEFINITION
A$	Volume Label Record contents
B$	Directory Record contents
C$	Dta file Record contents
D$	Cassette Device Specifier
F$	HP-41 Data file name
G$	HP-75 BASIC data file name
D	number of records in the tape directory
I	length of F$
J	loop control for reading directory records
K	loop control for directory enties
E$	file name extracted from B$
F1	variable indicating file type of E$
T$	input variable for yes or no test
F2	track number of data file
F3	starting record number of data file
F4	length of data file in records
F5	number of registers allocated in CREATE statement
L	loop control for reading each record of data file
M	loop control for reading each data file register
T1	current register and data file line number
U	variable for counting leading nulls in ALPHA data
N	string position pointer
O	loop counter for decoding each ALPHA character
Z$	string contained in register

Start by CREATEing an 11 register data file on a tape using the HP-41.
This can be done as follows:

1. Connect the HP-41 to the HP-82161A Cassette Drive
2. Key in: [XEQ] [ALPHA] SIZE [ALPHA] 011
   1 [EEX] 75 [STO] 00
   [ALPHA] ABCDEF [///] [STO] 01 [ALPHA]
   13 [EEX] [CHS] 75 [STO] 02
   41 [CHS] [EEX] [CHS] 75 [STO] 03
   5 [STO] 04 6 [STO] 05 7 [STO] 06 8 [STO] 07
   9 [STO] 08 10 [STO] 09 11 [STO] 10
3. CREATE the file TSTDTA on the cassette tape and store the data in the file
4. Disconnect the HP-41 and the cassette drive, and connect the cassette drive to the HP-75

TO CREATE the file press the following keys:
[ALPHA] TSTDTA [ALPHA] 11 [XEQ] [ALPHA] CREATE [ALPHA]
To store the data in the file, press the following:
[XEQ] [ALPHA] WRTR [ALPHA]

7500167 **SAMPLE PROBLEM SOLUTION**

DISPLAY CONTENTS	USER RESPONSE	COMMENTS
	[ATTN]	turn HP-75 ON
Cassette Device Spec.?	[RUN] "DATCNVRT" or [RTN]	assign the drive
41 DATA file name?	TSTDTA [RTN]	name of the tape file
75 DATA file name?	TSTDTA [RTN]	name of the internal HP-75
		file where the data will be
		left
	EDIT "TSTDTA"	EDIT the file
TSTDTA B	[↑]	view 1st line of file
1 DATA 1.E75	[↓]	
2 DATA'ABCDEF'	[↓]	
3 DATA 1.3 E -74	[↓]	
4 DATA -4.1 E-74	[↓]	
5 DATA 5	[↓]	
⋮		
11 DATA 11		last line of file

7500167

USER INSTRUCTIONS

INSTRUCTION	DISPLAY	INPUT
COPY both the BASIC program		
and the LEX file into the		
HP-75's memory. Be sure that		
the I/O Utilities program is		
also loaded.		
RUN the BASIC program	Cassette Device Spec.?	two character device
		specifier [RTN]
	41 DATA file name?	cassette data file name [RTN]
	75 DATA file name?	name of BASIC data file [RTN]
If the named tape file either		
doesn't exist or is not a data		
file, this message will be		
displayed:	file not found	
	read another file? (Y/N)	Y or N
If you wish to try a different		
file name, press Y RTN		
An N response terminates		
the program		

7500167

SUPPLEMENTARY INFORMATION

GLOBAL MACHINE STATUS

CONDITION	EXPLANATION, IF MANIPULATED OR ALTERED BY PROGRAM
ALARM	
ASSIGN IO	
ASSIGN #	1 to user specified BASIC file, and finally unassigned
BEEP	
DEFAULT	
DELAY	
DISPLAY IS	
ENDLINE	
EXACT	
Files	Creates a new BASIC file if named file doesn't exist
Keyboard	
MARGIN	
PRINTER IS	
PWIDTH	
STANDBY	
Trigonometric mode	
WIDTH	
OTHER	
BASIC keyword	HP41 DTA( )

NOTES:

The function HP41 DTA accepts a string input and returns a real number. A random string input can result in the display of a non-normalized number. It may be possible to 'hang' the system by performing computations with this number, although this has not been confirmed.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

7500167 PROGRAM LISTING

"DATCNVRT"
Page 1 of 2

```
100 DIM A$[256],B$[256],C$[256]
110 ! ****************************************
120 ! ** Get valid cassette device specifier **
130 ! ****************************************
140 INPUT 'Cassette Device Spec.?';D$
150 IF NOT POS(D$,':') THEN D$=':'&D$ ! Must have a :
160 ON ERROR BEEP @ DISP 'Invalid Device Spec.' @ GOTO 140
170 D1$=ENTIO$(D$,'TAD#,SAI') ! Get accessory ID
180 OFF ERROR
190 IF NUM(D1$)<>16 THEN BEEP @ DISP D$;' Not A Cassette Drive' @ GOTO 140
200 ! ******************************
210 ! ** Get valid data file names **
220 ! ******************************
230 INPUT '41 DATA file name?';F$ ! Name of file on tape
240 IF LEN(F$)>7 THEN BEEP @ DISP 'Name Too Long' @ GOTO 230
250 F$[LEN(F$)+1,10]=' ' ! Blank fill to ten characters
260 F$=UPRC$(F$) ! All file names are uppercase
270 INPUT '75 DATA file name?';G$ ! Name of file in 75 to store data into
280 IF LEN(G$)>8 THEN BEEP @ DISP 'Name Too Long' @ GOTO 270
290 ON ERROR BEEP @ DISP G$;' Not A Valid File Name' @ GOTO 270
300 ASSIGN # 1 TO G$ ! Open data file
310 OFF ERROR
320 ! ************************************
330 ! ** Search the tape for the data file **
340 ! ************************************
350 SENDIO D$,'UNL,LAD#,DDL4',CHR$(0)&CHR$(0) ! Seek cassette to track0,rec.0
360 A$=ENTIO$(D$,'UNL,TAD#,DDT2,SDA') ! Read volume label record
370 D=NUM(A$[20]) ! # of records in directory
380 SENDIO D$,'UNL,LAD#,DDL4',CHR$(0)&CHR$(2) ! Position tape to start of dir
390 FOR J=1 TO D ! For as many records as are in the dir.
400 SENDIO D$,'UNL,TAD#,DDT2,DDT4','' ! Read a record, exchange the buffers
410 B$=ENTIO$(D$,'UNL,TAD#,DDT1,SDA') ! Read in the contents of buffer 1
420 FOR K=0 TO 7 ! For each dir. entry in the record
430 E$=B$[K*32+1,K*32+10] ! Extract filename
440 IF E$<>F$ THEN GOTO 470 ! Not a match?
450 F1=NUM(B$[K*32+12]) ! File type
460 IF F1=208 THEN GOTO 550 ELSE GOTO 480 ! If not data file keep looking
470 IF NUM(B$[K*32+1])=255 THEN GOTO 500 ! End of directory?
480 NEXT K
490 NEXT J
500 DISP 'data file ';F$[1,7];' not found' ! IT ain't on the tape
510 GOTO 910
520 ! ********************************
530 ! ** Extract info. from directory **
540 ! ********************************
550 F2=NUM(B$[K*32+15]) ! Track #
560 F3=NUM(B$[K*32+16]) ! Record #
570 F4=NUM(B$[K*32+20]) ! Length in records
580 F5=NUM(B$[K*32+30]) ! # of registers allocated by "CREATE"
```

7500167 PROGRAM LISTING

"DATCNVRT"
Page 2 of 2

```
590 ! ***************************
600 ! ** Read record from file **
610 ! ***************************
620 SENDIO D$,'unl,lad#,ddl4',CHR$(F2)&CHR$(F3) ! Position to start of file
630 FOR L=1 TO F4 ! For each record
640 SENDIO D$,'unl,tad#,ddt2,ddt4','' ! tell cassette to read & exchange buf.
650 C$=ENTIO$(D$,'unl,tad#,ddt1,sda') ! Read record
660 ! ***************************************
670 ! ** Pull individual register contents **
680 ! ***************************************
690 FOR M=1 TO 32 ! Begin loop to extract 8 bytes at a time
700 I1=(L-1)*32+M
710 IF I1>F5 THEN GOTO 910 ! Up to the last register
720 ! *****************************************
730 ! ** Decode individual register contents **
740 ! *****************************************
750 IF MOD(NUM(C$[M*8-6]),16)=1 THEN 770 ! If it's ALPHA data, skip the
 numeric stuff
760 PRINT # 1,I1 ; HP41 DTA(C$[M*8-7]) @ GOTO 890 ! Transform to a real
 no.& get next bytes
770 J=7 ! Variable for counting nulls
780 IF C$[M*8-7+J,M*8-7+J]='' THEN J=J-1 @ GOTO 780 ! Count nulls
790 IF J=1 THEN N=-1 @ GOTO 870 ! If there is only 1 chr.
800 N=0 ! String position counter
810 FOR O=J+1 TO 5 STEP -1 ! Only down to the imbedded 0 byte
820 N=N+1 ! Increment str. position pointer
830 Z$[N,N]=CHR$(MOD(NUM(C$[M*8-8+O]),16)*16+NUM(C$[M*8-9+O])\16) ! Decode
 ALPHA char.
840 NEXT O
850 IF J=1 THEN 870 ! IF there was only one char.
860 Z$[N+1,N+1]=CHR$(MOD(NUM(C$[M*8-4]),16)*16+NUM(C$[M*8-6])\16) ! 2nd str.
 chr.
870 Z$[N+2,N+2]=C$[M*8-7] ! First str. chr.
880 PRINT # 1,I1 ; Z$ ! Store str. in 75 file
890 NEXT M
900 NEXT L
910 ASSIGN # 1 TO * ! Close file
920 INPUT 'Read another file?(Y/N)';T$
930 IF UPRC$(T$[1,1])='Y' THEN GOTO 230 ! Re-run or quit.
940 END ! That's all folks
```

7500157

PROGRAM DESCRIPTION

Program Title HP-41 CASSETTE DATA FILE TRANSFER TO HP-75 DATA

Contributor Harry Phinney

Address Hewlett-Packard, 1000 NE Circle Blvd.

City Corvallis State OR Country USA

Telephone (503) 757-2000 Zip/Postal Code 97330

Program Description (include equations) This program uses the SENDIO and ENTIO$ commands to read a cassette data file created by the WRTR or WRTRX functions of an HP-41. The program then transforms each register (8 bytes) into the HP-75's internal representation for real numbers, or strings, (depending on what was in the register) and places the result in a BASIC data file in the HP-75. A short LEX file is used to transform an 8 byte string into an HP-75 real number. The decoding of ALPHA data is done in BASIC.

The program prompts for the HP-IL assigned device code of the cassette drive.

Necessary Accessories HP-82161A Cassette drive. HP-75 I/O Utilities Solutions Book

Operating limits and warnings The LEX function HP41" DTA will accept any string as an input. A random string input will often result in the display of a non-normalized number.

Minimum RAM Requirement

References HP-83/HP-85 Assembler ROM Manual, HP-82161A Cassette Drive Manual

The HP-IL SYSTEM by Kane, Harper, Ushijima Pub. by Osborne/Mcgraw Hill

7500167

VARIABLE DEFINITIONS

NAME	DEFINITION
A$	Volume Label Record contents
B$	Directory Record contents
C$	Dta file Record contents
D$	Cassette Device Specifier
F$	HP-41 Data file name
G$	HP-75 BASIC data file name
D	number of records in the tape directory
I	length of F$
J	loop control for reading directory records
K	loop control for directory enties
E$	file name extracted from B$
F1	variable indicating file type of E$
T$	input variable for yes or no test
F2	track number of data file
F3	starting record number of data file
F4	length of data file in records
F5	number of registers allocated in CREATE statement
L	loop control for reading each record of data file
M	loop control for reading each data file register
T1	current register and data file line number
U	variable for counting leading nulls in ALPHA data
N	string position pointer
O	loop counter for decoding each ALPHA character
Z$	string contained in register

Start by CREATEing an 11 register data file on a tape using the HP-41.
This can be done as follows:

1. Connect the HP-41 to the HP-82161A Cassette Drive
2. Key in: [XEQ] [ALPHA] SIZE [ALPHA] 011
   1 [EEX] 75 [STO] 00
   [ALPHA] ABCDEF [///] [STO] 01 [ALPHA]
   13 [EEX] [CHS] 75 [STO] 02
   41 [CHS] [EEX] [CHS] 75 [STO] 03
   5 [STO] 04 6 [STO] 05 7 [STO] 06 8 [STO] 07
   9 [STO] 08 10 [STO] 09 11 [STO] 10
3. CREATE the file TSTDTA on the cassette tape and store the data in the file
4. Disconnect the HP-41 and the cassette drive, and connect the cassette drive to the HP-75

TO CREATE the file press the following keys:
[ALPHA] TSTDTA [ALPHA] 11 [XEQ] [ALPHA] CREATE [ALPHA]
To store the data in the file, press the following:
[XEQ] [ALPHA] WRTR [ALPHA]

7500167 **SAMPLE PROBLEM SOLUTION**

DISPLAY CONTENTS	USER RESPONSE	COMMENTS
	[ATTN]	turn HP-75 ON
Cassette Device Spec.?	[RUN] "DATCNVRT" or [RTN]	assign the drive
41 DATA file name?	TSTDTA [RTN]	name of the tape file
75 DATA file name?	TSTDTA [RTN]	name of the internal HP-75
		file where the data will be
		left
	EDIT "TSTDTA"	EDIT the file
TSTDTA B	[↑]	view 1st line of file
1 DATA 1.E75	[↓]	
2 DATA'ABCDEF'	[↓]	
3 DATA 1.3 E -74	[↓]	
4 DATA -4.1 E-74	[↓]	
5 DATA 5	[↓]	
⋮		
11 DATA 11		last line of file

7500167

USER INSTRUCTIONS

INSTRUCTION	DISPLAY	INPUT
COPY both the BASIC program and the LEX file into the HP-75's memory. Be sure that the I/O Utilities program is also loaded.		
RUN the BASIC program	Cassette Device Spec.?	two character device specifier [RTN]
	41 DATA file name?	cassette data file name [RTN]
	75 DATA file name?	name of BASIC data file [RTN]
If the named tape file either doesn't exist or is not a data file, this message will be displayed:	file not found read another file? (Y/N)	Y or N
If you wish to try a different file name, press Y RTN An N response terminates the program		

7500167

SUPPLEMENTARY INFORMATION

GLOBAL MACHINE STATUS

CONDITION	EXPLANATION, IF MANIPULATED OR ALTERED BY PROGRAM
ALARM	
ASSIGN IO	
ASSIGN #	1 to user specified BASIC file, and finally unassigned
BEEP	
DEFAULT	
DELAY	
DISPLAY IS	
ENDLINE	
EXACT	
Files	Creates a new BASIC file if named file doesn't exist
Keyboard	
MARGIN	
PRINTER IS	
PWIDTH	
STANDBY	
Trigonometric mode	
WIDTH	
OTHER	
BASIC keyword	HP41 DTA( )

NOTES:

The function HP41 DTA accepts a string input and returns a real number. A random string input can result in the display of a non-normalized number. It may be possible to 'hang' the system by performing computations with this number, although this has not been confirmed.

7500167 PROGRAM LISTING

"DATCNVRT"
Page 1 of 2

```
100 DIM A$[256],B$[256],C$[256]
110 ! ****************************************
120 ! ** Get valid cassette device specifier **
130 ! ****************************************
140 INPUT 'Cassette Device Spec.?';D$
150 IF NOT POS(D$,':') THEN D$=':'&D$ ! Must have a :
160 ON ERROR BEEP @ DISP 'Invalid Device Spec.' @ GOTO 140
170 D1$=ENTIO$(D$,'TAD#,SAI') ! Get accessory ID
180 OFF ERROR
190 IF NUM(D1$)<>16 THEN BEEP @ DISP D$;' Not A Cassette Drive' @ GOTO 140
200 ! ******************************
210 ! ** Get valid data file names **
220 ! ******************************
230 INPUT '41 DATA file name?';F$ ! Name of file on tape
240 IF LEN(F$)>7 THEN BEEP @ DISP 'Name Too Long' @ GOTO 230
250 F$[LEN(F$)+1,10]=' ' ! Blank fill to ten characters
260 F$=UPRC$(F$) ! All file names are uppercase
270 INPUT '75 DATA file name?';G$ ! Name of file in 75 to store data into
280 IF LEN(G$)>8 THEN BEEP @ DISP 'Name Too Long' @ GOTO 270
290 ON ERROR BEEP @ DISP G$;' Not A Valid File Name' @ GOTO 270
300 ASSIGN # 1 TO G$ ! Open data file
310 OFF ERROR
320 ! ****************************************
330 ! ** Search the tape for the data file **
340 ! ****************************************
350 SENDIO D$,'UNL,LAD#,DDL4',CHR$(0)&CHR$(0) ! Seek cassette to track0,rec.0
360 A$=ENTIO$(D$,'UNL,TAD#,DDT2,SDA') ! Read volume label record
370 D=NUM(A$[20]) ! # of records in directory
380 SENDIO D$,'UNL,LAD#,DDL4',CHR$(0)&CHR$(2) ! Position tape to start of dir
390 FOR J=1 TO D ! For as many records as are in the dir.
400 SENDIO D$,'UNL,TAD#,DDT2,DDT4','' ! Read a record, exchange the buffers
410 B$=ENTIO$(D$,'UNL,TAD#,DDT1,SDA') ! Read in the contents of buffer 1
420 FOR K=0 TO 7 ! For each dir. entry in the record
430 E$=B$[K*32+1,K*32+10] ! Extract filename
440 IF E$<>F$ THEN GOTO 470 ! Not a match?
450 F1=NUM(B$[K*32+12]) ! File type
460 IF F1=208 THEN GOTO 550 ELSE GOTO 480 ! If not data file keep looking
470 IF NUM(B$[K*32+1])=255 THEN GOTO 500 ! End of directory?
480 NEXT K
490 NEXT J
500 DISP 'data file ';F$[1,7];' not found' ! IT ain't on the tape
510 GOTO 910
520 ! ********************************
530 ! ** Extract info. from directory **
540 ! ********************************
550 F2=NUM(B$[K*32+15]) ! Track #
560 F3=NUM(B$[K*32+16]) ! Record #
570 F4=NUM(B$[K*32+20]) ! Length in records
580 F5=NUM(B$[K*32+30]) ! # of registers allocated by "CREATE"
```

7500167 PROGRAM LISTING

"DATCNVRT"
Page 2 of 2

```
590 ! ***************************
600 ! ** Read record from file **
610 ! ***************************
620 SENDIO D$,'unl,lad#,ddl4',CHR$(F2)&CHR$(F3) ! Position to start of file
630 FOR L=1 TO F4 ! For each record
640 SENDIO D$,'unl,tad#,ddt2,ddt4','' ! tell cassette to read & exchange buf.
650 C$=ENTIO$(D$,'unl,tad#,ddt1,sda') ! Read record
660 ! ************************************
670 ! ** Pull individual register contents **
680 ! ************************************
690 FOR M=1 TO 32 ! Begin loop to extract 8 bytes at a time
700 I1=(L-1)*32+M
710 IF I1>F5 THEN GOTO 910 ! Up to the last register
720 ! ******************************************
730 ! ** Decode individual register contents **
740 ! ******************************************
750 IF MOD(NUM(C$[M*8-6]),16)=1 THEN 770 ! If it's ALPHA data, skip the
 numeric stuff
760 PRINT # 1,I1 ; HP41 DTA(C$[M*8-7]) @ GOTO 890 ! Transform to a real
 no.& get next bytes
770 J=7 ! Variable for counting nulls
780 IF C$[M*8-7+J,M*8-7+J]='' THEN J=J-1 @ GOTO 780 ! Count nulls
790 IF J=1 THEN N=-1 @ GOTO 870 ! If there is only 1 chr.
800 N=0 ! String position counter
810 FOR O=J+1 TO 5 STEP -1 ! Only down to the imbedded 0 byte
820 N=N+1 ! Increment str. position pointer
830 Z$[N,N]=CHR$(MOD(NUM(C$[M*8-8+O]),16)*16+NUM(C$[M*8-9+O])\16) ! Decode
 ALPHA char.
840 NEXT O
850 IF J=1 THEN 870 ! IF there was only one char.
860 Z$[N+1,N+1]=CHR$(MOD(NUM(C$[M*8-4]),16)*16+NUM(C$[M*8-6])\16) ! 2nd str.
 chr.
870 Z$[N+2,N+2]=C$[M*8-7] ! First str. chr.
880 PRINT # 1,I1 ; Z$ ! Store str. in 75 file
890 NEXT M
900 NEXT L
910 ASSIGN # 1 TO * ! Close file
920 INPUT 'Read another file?(Y/N)';T$
930 IF UPRC$(T$[1,1])='Y' THEN GOTO 230 ! Re-run or quit.
940 END ! That's all folks
```

7500168

PROGRAM DESCRIPTION

Sent Aug. 83 by Michael Markov

Program Title: Terminal Emulator

Contributor: Hewlett-Packard Company

Address: 1000 NE Circle Blvd.

City: Corvallis State: Oregon Country: U.S.A.

Telephone: Zip/Postal Code: 97330

Program Description (include equations): This program allows the HP-75 to be used as a terminal to access another computer. It uses the full capabilities of the HP 82168A Acoustic Coupler (modem) to set parity and protocal.

Other program features include an automatic Hewlett-Packard Interface Loop assignment without the program, keystroke echo toggling, and printer toggling.

User interface is enhanced by using several menus from which the user determines the choices of parity, protocol, etc. All the user needs to do is connect all HP-IL devices and load and run the program.

Necessary Accessories: 00075-13013 I/O Utilities Solutions Book; 82168A Acoustic Coupler.

Operating limits and warnings

Minimum RAM Requirement: 2403-static
2689-allocated

References

User Instructions

To use the terminal emulator program, first read the magnetic card included with the HP-75 "I/O Utilities" Users' Library Solutions Book through the card reader. Then enter the terminal emulator program into the HP-75 using either the program listing or the magnetic cards.

To begin the program, type "run filename" (using the file name under which you have stored the terminal emulator program) and press [RTN]. The HP-75 will initialize the interface loop. This program specifically searches the loop for the HP 82168A Acoustic Coupler, the HP 82163 Video Interface, the HP 82161A Digital Cassette Drive, the HP 82162A Thermal Printer, and the HP 82905B Printer. Other devices that may be on the loop are not used by the program.

After the HP-75 has initialized the loop, it will show the device assignments on its display. If an HP 82163 Video Interface is used, the HP-75 will display the device assignments on the video display.

When the HP-75 has finished assigning devices, it will prompt you with "ready . . ." on its display. You should now set the coupler to the parity and protocol that the host computer will be using. Also, you can select the display mode desired. To do this, press [EDIT] - the HP-75 will prompt you with:

Frame, Print:OFF, Echo:OFF

You can select a mode or setting by pressing the key corresponding to the underlined letter in a keyword. For example, to turn Print ON and Print OFF, press [P]. When print is ON, all information sent and received from the telephone line will be printed, regardless of which device is the display device.

If you select Echo, the program will display all information sent and received over the phone line on the display device and on the printer (if Print is ON). If Echo is OFF, only information received from the phone line will be displayed and printed.

> Note: Some host computers use an echo function and might echo information that is received. If you are linked with such a computer, you might receive a double echo when Echo is ON and a single echo when Echo is OFF.

If you press [F] for Frame, you will see the following prompt on the HP-75 display.

Parity:Even, PRotocol: XON/XOFF

If you press [P] , the HP-75 will prompt you to select a parity. If you press [R] , the HP-75 will prompt you to select a protocol.

After pressing [P] , the HP-75 will display:

Even, Odd, 0, 1, None

With this prompt in the display, you can select the parity that the coupler will use. Be sure to select the parity that the host computer will be using. To select a parity, press [E] , [O], [O], [1], or [N]. After selecting a parity, the HP-75 will display the "ready . . ." prompt again.

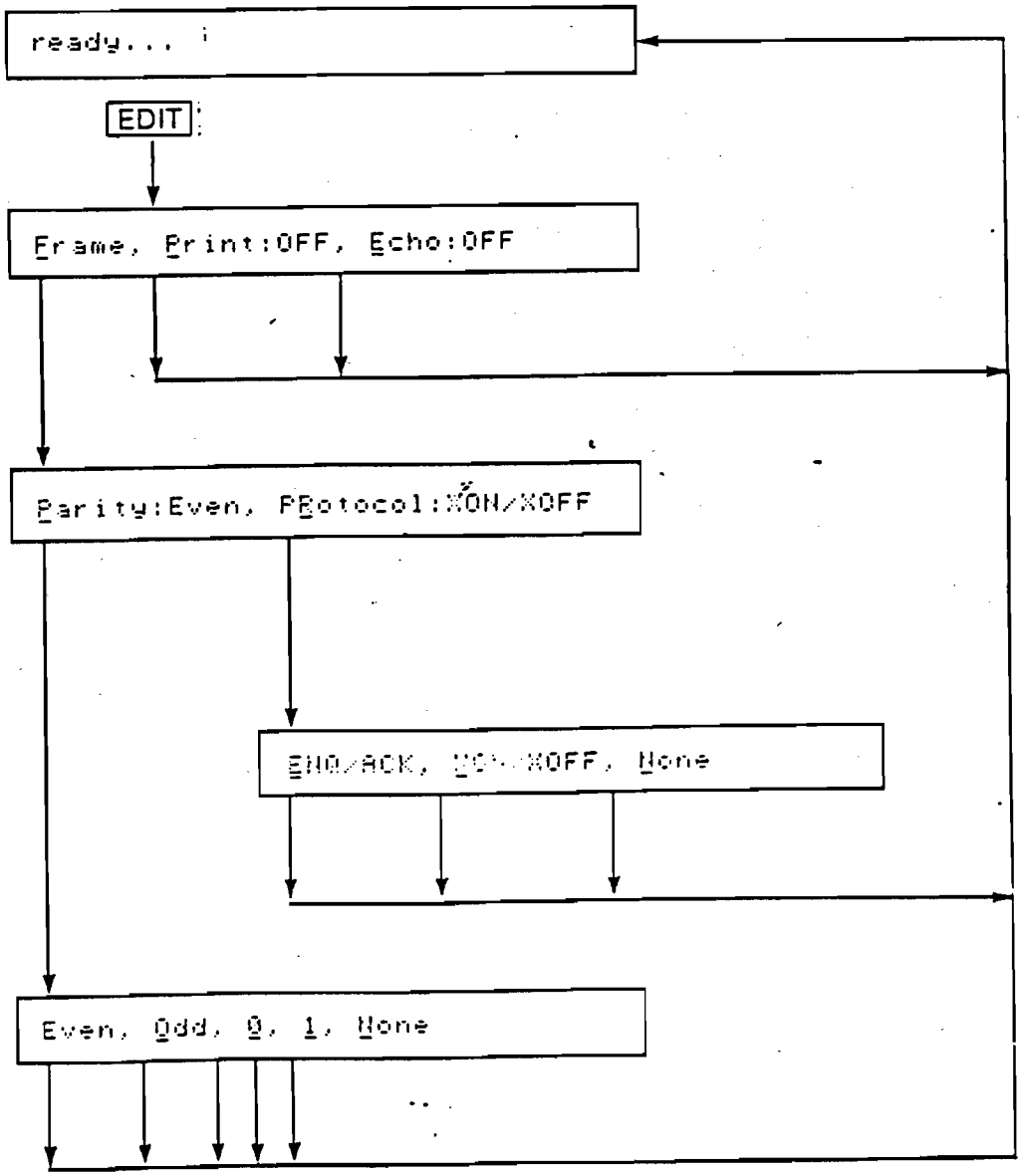
After pressing [R] for protocol, the HP-75 display will show the prompt:

ENQ/ACK, XON/XOFF, None

Pressing [E] will select ENQ/ACK protocol, pressing [x] will select XON/XOFF protocol, and pressing [N] will select no protocol. When you have selected a protocol (or no protocol), the HP-75 will display the "ready . . ." prompt.

> Note: Pressing any key other than one corresponding to a keyword in the prompt will return the HP-75 to the "ready . . ." prompt. This is useful when you want to view the coupler's settings without changing them.

The following diagram shows the sequence of keystrokes needed to set the mode, protocol, and parity on the coupler.



When you have finished initializing the coupler, you can dial up the host computer, listen for the carrier tone, and insert the telephone handset into the coupler. You are now ready to communicate with the computer.

If you need to change any of the coupler settings while connected to a host computer, you can do so by pressing [EDIT] and the appropriate keys.

If at any time you need to do a "break", press [CNTL] and [ESC] simultaneously. If the computer recognizes a break signal over the telephone line, it will halt operations and wait for you to send a system command. (This is the usual response; however, not all computers respond the same way to a break signal.)

If at anytime you want to halt the terminal emulator program, press [CLR]. If you do this, you may need to reinitialize the coupler and redial the host computer.

7500160

SAMPLE PROBLEM

Phone up a host system that communicates with ODD parity and ENQ/ACK protocol.
The host system will not echo your keystrokes.

7500168

SAMPLE PROBLEM SOLUTION

DISPLAY CONTENTS	USER RESPONSE	COMMENTS
		Connect all HP-IL devices and turn them on.
		Load "HP75TERM" into the HP-75.
	Run "HP75TERM"	
Device #1 = ': ' . . .		see all devices assigned
ready . . .	[EDIT]	Now configure the 'terminal'
Frame, Print:OFF,Echo.OFF	F	Look at default parity & protocol.
Parity:EVEN,PRotocol:XON/XOFF	P	Change the parity
Even,Odd,Ø,1,None	O	Set to odd parity
ready . . .	[EDIT]	Need to change protocol
Frame, Print:OFF,Echo.OFF	F	
Parity:ODD,PRotocol:XON/XOFF	R	Look at protocol choice
ENQ/ACK,XON/XOFF,None	E	Set to ENQ/ACK protocol
ready . . .	[EDIT]	Change echo status
Frame, Print:OFF,Echo:OFF	E	
ready . . .	[EDIT]	Look at status
Frame,Ans,Org,Print:OFF,Echo:ON	[RTN]	Any other key returns to prompt.
ready . . .		Now, pick up the phone and dial the host system computer, listen for the carrier tone, and insert the handset into the coupler.

7500158

SUPPLEMENTARY INFORMATION

GLOBAL MACHINE STATUS

CONDITION	EXPLANATION, IF MANIPULATED OR ALTERED BY PROGRAM
ALARM	
ASSIGN IO	Program assigns only certain devices, others are ignored.
ASSIGN #	
BEEP	
DEFAULT	
DELAY	DELAY Ø if video interface is present, else DELAY .25
DISPLAY IS	DISPLAY is * during execution, and set to the video interface (if it exists) at the end.
ENDLINE	
EXACT	
Files	
Keyboard	
MARGIN	
PRINTER IS	All I/O is with PRINT, PRINTER IS depends on the loop configuration
PWIDTH	INF
STANDBY	OFF
Trigonometric mode	
WIDTH	INF
OTHER	

NOTES: See Program Comments for optional linefeed on program line #1140.

7500158

VARIABLE DEFINITIONS

NAME	DEFINITION
A$	This variable has three uses: FIRST, in lines 1540 and 1550, A$ contains a string which serves as the command list for the ENTIO$ instruction. SECOND, in lines 1630 through 1720 when the HP-IL loop is configured, A$ contains the string of loop assignment mnemonics used by the ASSIGN IO command. THIRD, in lines 1850 through 1890 when the I/O route is configured, A$ contains the string of loop assignment mnemonics used by the PRINTER IS command.
C$	Remote mode command sent to the modem (such as parity or protocol setting).
C1	The HP-IL loop address of the (first) HP 82161A Digital Cassette Drive found (or 0 if the device is not on the loop).
D	HP-IL accessory ID of a device on the loop, used to determine what device has been found.
E$	The character string "ON" or "OFF" indicating the status of E1, the Boolean flag for the echo.
E1	A Boolean flag for whether the user wants the echo on or off, (On: 1, Off: 0).
H$	The character string "ENQ/ACK", "XON/XOFF", or "NONE ", indicating which protocol instruction ("C0;", "C1;", or "C2;") was sent to the modem..
I	A counter used to determine the sequential loop address when searching for devices.
I1	A counter used to determine the sequential loop address when setting up A$ for use in the ASSIGN IO command.
K$	This is a buffer for the KEY$ input from the keyboard when a program menu is displayed. K$ is used to determine which branch the program should take.
K1$	Contains the character strings equivalent to pressing the RTN, EDIT, ESC, and CLR keys on the HP-75 keyboard. K1$ is used to determine which branch the program should take.

Mi The HP=IL loop address of the (first) HP 82168A modem found (or 0 if the device is not on the loop.

P$ The character string "ON" or "OFF" indicating the status of P2, the Boolean flag for the printer.

P1 The HP-IL loop address of the (first) HP 82162A thermal printer or HP 82905B dot matrix printer found (or 0 if neither printer is on the loop).

P1$ The character string "EVEN", "ODD ", "0 ", "1 ", or "NONE", indicating which parity instruction ("P0;", "P1;", "P2;", "P3;" or "P4;") was sent to the modem..

P2 This variable has two uses:

FIRST, in program lines 1590 and 1730 when the HP-IL loop is configured, P2 denotes which type of printer has been found. The default value 0 indicates an HP 82162A thermal printer, and the value 33 (an accessory ID) indicates an HP 82905B dot matrix printer.

SECOND, in program lines 1280 and 1840, P2 is a Boolean flag for whether the user wants the printer on or off, (On: 1, OFF: 0).

T1 The HP-IL loop address of the (first) HP 82163A Video Interface found (or 0 if the device is not on the loop).

Lines 1020 - 1070:
Set up HP-75 display control, initialize variables.

Line 1080:
Call the subroutine which automatically configures most devices on the HP-IL loop.

Line 1090:
End the program if a modem was not found on the loop.

Line 1100:
Clear the modem.

Line 1120:
Wait for input from the keyboard. While there is none, check the modem for data and print incoming characters.

Line 1130:
Program branching determined by the K$ value input from the keyboard. Program will branch as follows:
Line 1140 if [RTN] was pressed,
Line 1230 if [EDIT] was pressed,
Line 1200 if [ESC] was pressed,
Line 1790 if [CLR] was pressed, and
Line 1150 if any other key was pressed.

Line 1140:
Optional line feed following the [RTN] key. Use of the line feed option depends on the requirements of the computer with which the user is communicating. Activate and deactivate the line feed by removing or adding the "!" before the statement (which makes it a comment).

Line 1150:
Send the K$ character to the modem.

Line 1160:
Print K$ if the echo is turned on.

Lines 1180 - 1190:
Subroutine called from line 1120 which prints data received from the modem.

Lines 1200 - 1220:
These program lines are called from line 1130 (by pressing the [ESC] key). They send an "escape" to the other computer.

Lines 1230 - 1250:
These program lines are called from line 1130 (by pressing the [EDIT] key). They are the text for the first terminal configuration menu.

PROGRAM COMMENTS (continued)

Line 1260:
Wait for input from the keyboard. While there is none, check the modem for data and print incoming characters.

Line 1270:
Program branching determined by the K$ value input from the keyboard. Input is prompted from the terminal configuration menu. The program will branch as follows:
Line 1330 if "F" or "f" was pressed,
Line 1280 if "P" or "p" was pressed,
Line 1310 if "E" or "e" was pressed, and
Line 1110 if any other key was pressed.

Lines 1280 - 1300:
These program lines are called from line 1270 (by pressing a "P" or a "p" on the keyboard). They toggle the printer flag on or off and call the subroutine which reconfigures the printer/display control.

Line 1310:
This program line is called from line 1270 (by pressing an "E" or an "e" on the keyboard). It toggles the echo flag on of off.

Line 1330:
This program line is called from line 1270 (by pressing an "F" or an "f" on the keyboard). It is the menu for selecting either a parity or protocol change.

Line 1340:
Wait for input from the keyboard. While there is none, check the modem for data and print incoming characters.

Line 1350:
Program branching determined by the K$ value input from the keyboard. The program will branch as follows:
Line 1360 if "P" or "p" was pressed,
Line 1450 if "R" or "r" was pressed, and
Line 1110 if any other key was pressed.

Lines 1360 - 1370:
These program lines are called from line 1350 (by pressing a "P" or a "p" on the keyboard). They are the menu for the parity selection.

Line 1380:
Wait for input from the keyboard. While there is none, check the modem for data and print incoming characters.

Line 1390:
Program branching determined by the K$ value input from the keyboard. The program will branch as follows:
Line 1400 if "E" or "e" was pressed,
Line 1410 if "O" or "o" was pressed,
Line 1420 if "0" was pressed,
Line 1430 if "1" was pressed,
Line 1440 if "N" or "n" was pressed, and
Line 1110 if any other key was pressed.

Lines 1400 - 1440:
Each of these program lines is called from line 1390 based on which parity is desired. The parity abbreviation is placed in P1$ and the remote mode command for the modem to initiate this selected parity is placed in C$, the sent on line 1820.

Lines 1450 - 1460:
These program lines are called from line 1350 (by pressing an "R" or an "r" on the keyboard). They are the menu for the protocol selection.

Line 1470:
Wait for input from the keyboard. While there is none, check the modem for data and print incoming characters.

Line 1480:
Program branching determined by the K$ value input from the keyboard. The program will branch as follows:
Line 1490 if "E" or "e" was pressed,
Line 1500 if "X" or "x" was pressed,
Line 1510 if "N" or "n" was pressed, and
Line 1110 if any other key was pressed.

Lines 1490 - 1510:
Each of these program lines is called from line 1480 based on which protocol is desired. The protocol abbreviation is placed in H$ and the remote mode command for the modem to initiate this selected protocol is placed in C$, the sent on Line 1820.

Line 1520:
This is the first line of the loop configuration subroutine which is called from line 1080. It performs a "dummy" ASSIGN IO statement to wake up all devices on the loop.

Line 1530:
Initialize the counter used to determine the sequential loop address when searching for devices.

Lines 1540 - 1550:
Get an accessory ID from the next device on the loop.

Line 1560:
Quit searching if there are no more devices on the loop.

Lines 1570 - 1610:
Look for the accessory ID of certain devices on the loop and configure them:
D=65 ==> HP 82168A modem,
D=48 ==> HP 82163A video interface,
D=33 ==> HP 82905B printer,
D=32 ==> HP 82162A thermal printer,
D=16 ==> HP 82161A digital cassette drive.

Line 1620:
Increment the loop address by 1 and go back to get the next accessory ID.

Line 1630:
Initializes the counter used to determine the sequential loop address when setting up for the ASSIGN IO command.

Line 1640:
If the loop device is not listed on the configuration list in program lines 1570 - 1610, then go to line 1700 to assign a "dummy" device code.

Lines 1650 - 1680:
Assign device codes for the devices found.

Line 1700:
Generate a "dummy" device code.

Line 1710:
Increment the loop address by 1. Go to line 1720 if there are no more devices on the loop, otherwise insert a comma as a separator between device codes and go to line 1640.

Line 1720:
Perform the loop assignment.

Line 1730:
If an HP 82905B printer is on the loop then send a skip-over-perforation command to that printer.

Lines 1740 - 1750:
If an HP 82163A video interface is on the loop then clear the screen of the monitor or tv.

Line 1770:
Call the subroutine which configures the destination for the PRINT statements in the program.

Line 1780:
Return to the main program (line 1090) from this automatic loop configuration subroutine.

Lines 1790 - 1810:
End the program if a [CLR] was pressed on line 1130.

Line 1820:
Send a remote mode command (C$) to the modem.

Lines 1840 - 1900:
Configure (or re-configure) the destination for the PRINT statements in the program. This subroutine assists lines 1280 and 1290 toggle the printer off and on.

```
1000 ! . T E R M - 7 5 C
1010 !       rev 1.3
1020 STANDBY OFF
1030 DELAY 0 @ WIDTH INF @ PWIDTH INF
1040 C1,E1,M1,P1,P2,T1=0
1050 DIM A$[50],C$[3],E$[3],H$[8],K$[2],K1$[4],P$[3],P1$[4]
1060 P$='OFF' @ P1$='EVEN' @ H$='XON/XOFF' @ E$='OFF'
1070 K1$=CHR$(13)&CHR$(131)&CHR$(27)&CHR$(139)
1080 GOSUB 1520
1090 IF NOT M1 THEN PRINT '**** no modem ****' @ BEEP 200,.5 @ END
1100 CLEAR ':ph'
1110 DISP @ DISP 'ready...'
1120 K$=KEY$ @ IF K$='' THEN GOSUB 1180 @ GOTO 1120
1130 ON POS(K1$,UPRC$(K$))+1 GOTO 1150,1140,1230,1200,1790
1140 ! K$=K$&CHR$(10) ! Optional Line Feed.
1150 SENDIO ':ph','unl,lad#',K$
1160 IF E1 THEN PRINT K$;
1170 GOTO 1120
1180 PRINT ENTIO$(':ph','unl,tad#,sda');
1190 RETURN
1200 PRINT '*** break ***' @ SENDIO ':ph','unl,ren,lad#','B1;'
1210 WAIT .17 @ SENDIO ':ph','lad#','B0;'
1220 SENDIO '','nre','' @ GOTO 1110
1230 DISP @ DISP CHR$(NUM('F')+128)&'rame, ';
1240 DISP CHR$(NUM('P')+128)&'rint:'&P$;
1250 DISP ', '&CHR$(NUM('E')+128)&'cho:'&E$;
1260 K$=KEY$ @ IF K$='' THEN GOSUB 1180 @ GOTO 1260
1270 ON POS('FPE',UPRC$(K$))+1 GOTO 1110,1330,1280,1310
1280 IF P1 AND NOT P2 THEN P$='ON ' @ P2=1 ELSE P$='OFF' @ P2=0
1290 GOSUB 1840
1300 GOTO 1110
1310 IF E1 THEN E1=0 @ E$='OFF' ELSE E1=1 @ E$='ON '
1320 GOTO 1110
1330 DISP @ DISP CHR$(NUM('P')+128)&'arity:'&P1$&', P'&CHR$(NUM('R')+128)&'otoc
l:'&H$
1340 K$=KEY$ @ IF K$='' THEN GOSUB 1180 @ GOTO 1340
1350 ON POS('PR',UPRC$(K$))+1 GOTO 1110,1360,1450
1360 DISP @ DISP CHR$(NUM('E')+128)&'ven, '&CHR$(NUM('O')+128)&'dd, ';
1370 DISP CHR$(NUM('O')+128)&', '&CHR$(NUM('1')+128)&', '&CHR$(NUM('N')+128)&'o
e';
1380 K$=KEY$ @ IF K$='' THEN GOSUB 1180 @ GOTO 1380
1390 ON POS('EOO1N',UPRC$(K$))+1 GOTO 1110,1400,1410,1420,1430,1440
1400 P1$='EVEN' @ C$='P0;' @ GOTO 1820
1410 P1$='ODD ' @ C$='P1;' @ GOTO 1820
1420 P1$='0   ' @ C$='P2;' @ GOTO 1820
1430 P1$='1   ' @ C$='P3;' @ GOTO 1820
1440 P1$='NONE' @ C$='P4;' @ GOTO 1820
1450 DISP @ DISP CHR$(NUM('E')+128)&'NQ/ACK, '&CHR$(NUM('X')+128)&'ON/XOFF, ';
1460 DISP CHR$(NUM('N')+128)&'one';
1470 K$=KEY$ @ IF K$='' THEN GOSUB 1180 @ GOTO 1470
1480 ON POS('EXN',UPRC$(K$))+1 GOTO 1110,1490,1500,1510
1490 H$='ENQ/ACK' @ C$='C1;' @ GOTO 1820
1500 H$='XON/XOFF' @ C$='C2;' @ GOTO 1820
1510 H$='NONE   ' @ C$='C0;' @ GOTO 1820
1520 ASSIGN IO ':zz'
1530 I=1
1540 A$='tad'&STR$(I)&',sai'
1550 D=NUM(ENTIO$('',A$))
1560 IF NOT D THEN 1630
```

```
1570 IF D=65 AND NOT M1 THEN M1=I
1580 IF D=48 AND NOT T1 THEN T1=I
1590 IF D=33 AND NOT P1 THEN P1=I @ P2=33
1600 IF D=32 AND NOT P1 THEN P1=I
1610 IF D=16 AND NOT C1 THEN C1=I
1620 I=I+1 @ GOTO 1540
1630 A$='' @ I1=1
1640 IF NOT (C1=I1) AND NOT (M1=I1) AND NOT (T1=I1) AND NOT (P1=I1) THEN 1700
1650 IF C1=I1 THEN A$=A$&':ca'
1660 IF M1=I1 THEN A$=A$&':ph'
1670 IF P1=I1 THEN A$=A$&':pr' @ P1=1
1680 IF T1=I1 THEN A$=A$&':tv' @ T1=1
1690 GOTO 1710
1700 A$=A$&':'&CHR$(IP(I1/10)+65)&STR$(MOD(I1,10))
1710 I1=I1+1 @ IF I1<I THEN A$=A$&',' @ GOTO 1640
1720 ASSIGN IO A$
1730 IF P2=33 THEN PRINTER IS ':pr' @ PRINT CHR$(27)&'&l1L' @ P2=0
1740 IF T1 THEN DISPLAY IS ':tv' @ CLEAR ':tv' ELSE DELAY .25
1750 LIST IO
1760 DISPLAY IS *
1770 GOSUB 1840
1780 RETURN
1790 IF T1 THEN DISPLAY IS ':tv'
1800 DISP
1810 END
1820 SENDIO ':ph','unl,ren,lad#',C$ @ SENDIO '','nre',''
1830 GOTO 1110
1840 ON 2*T1+P2*P1+1 GOTO 1850,1860,1870,1880
1850 A$='' @ GOTO 1890
1860 A$=':pr' @ GOTO 1890
1870 A$=':tv' @ GOTO 1890
1880 A$=':tv,:pr'
1890 PRINTER IS A$
1900 RETURN
```

Sent Aug 83
by Michael Markov

TABLE OF CONTENTS

PROGRAM DESCRIPTION

FOOTBALL

This is pro-style football. The player assumes the role of quarterback and defensive signal-caller for the team. The opponent is the HP-75. There are 14 available offensive plays (8 runs, 6 passes) and 4 defensive alignments, along with field goal, punt, and quick kick capabilities.

There is a two-minute warning before the end of each half. Five to seven plays are left in the half at that point. At halftime, the player may view cumulative statistics for their team and the opponent.

Call all plays by their corresponding number:

OFFENSE	Runs:	1 = Dive	2 = Off Tackle
		3 = Scissors	4 = Trap
		5 = Sweep	6 = Option
		7 = Reverse	8 = Draw
	Passes:	9 = Sideline	10 = Look-in
		11 = Rollout	12 = Screen
		13 = 'Fly'	14 = 'Post'
	Kicks:	15 = Field Goal	16 = Punt
		17 = Quick Kick	
DEFENSE		1 = 'Pass Rush' (4-3)	2 = 'Okie' (5-2)
		3 = 'Short-Ydg'	4 = 'Prevent'

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		FOOTBALL	
1	Option to view play codes	Do you need a list of plays?N	[RTN] or Y [RTN]
	If 'Y' then display codes		
2	Enter team name (<=10 chars.)	Name of your team?	aaaa [RTN]
3	Opponent	Your opponent is the HP-75	
4	Coin toss. If HP-75 wins:	I won the toss and will receive	
	If player wins toss:	aaaa won the toss	
	Option to kick or receive ball:	Kick of Receive?	K or R [RTN]
5	Kickoff	Get ready for the kickoff!	
5a		nn yard kickoff!	
		nn yard runback	
6	Ball is on HP-75 or aaaa yard		
	line	Ball on [HP-75 / aaaa] nn yard line	
7	If player is offense:	X down aaaa X to go	
	Enter offensive play code:	Your play (1-17)	1-17 [RTN]
	Goto step 8		
7a	If HP-75 is offense:	X down HP-75 X to go	
	Enter defensive play code:	Enter defense (1-4):	1-4 [RTN]
8	Display plays chosen:	aaaa = Play	
		HP-75 = Play	
9	Display play results:	play results	
	Display current yard line,	Ball on [HP-75 / aaaa] nn yard line	
	possession, down:	1st down nn to go	
	Goto step 7		
10	Touchdowns. Add 6 points to	Touchdown [HP-75 / aaaa] !	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	appropriate socre, attempt	Kicking extra point	
	extra point (if good, add to		
	score) and goto step 5.		
11	Safety. Add 2 points to		
	appropriate score.		
	Kickoff from 20 yard line and		
	goto step 5a.	Kickoff from [HP-75 aaaa] 20 yd line	
12	Penalties. If holding:	HOLDING	
	If offside:	OFFSIDE	
		Penalty on [HP-75 aaaa]	
12a	If penalty on aaaa:	Penalty accepted: nn yards	
	If penalty on HP-75:	Accept the penalty (Y/N)	Y or N [RTN]
	If 'Y' then step 12a		
13	At halftime and end of game,	[*** HALFTIME *** *** END OF GAME ***]	
	give option to view statistics:	View statistics?	Y or N [RTN]
	If 'Y' then:	Statistics for aaaa	
		:	
		Statistics for HP-75	
		:	
14	After end-of-game statistics,		
	give option to play another		
	game:	Another game?Y	[RTN] or N [RTN]
	If 'Y' then step 4 else stop		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
B( )	Statistics	L( ),G( )	Storage for L, G
S8	Indicates safety	U	Constant for Y
R2	Indicates extra point	M	Fudge factor for Y
P8	Penalty yards	D9	Defensive play
Z9	Flag to show game over	I	Index and looping
D	Counts downs	W	Same as Q
T	Timer	Q	Indicates team in possession
S1	Player's score	X	Yard line
S2	HP-75's score	Y	Yards gained, lost
P1	Previous play	R	Yards runback
P	Offensive play	P9	Extra points
S	Yards gained before 1st down	T5	Indicates possession in 2nd half
G1	Indicates = 10 yards to goal	E$	Offensive plays
T1,Z	Indicates second half	D$	Downs
H9,E9	Time constants	F$	Statistics
F1,F2	Figure distance to goal	R$	Players team name
L,G	Help determine yards gained or lost for a play	Q$	User Interaction
E2$	Defensive plays		

NOTES AND REFERENCES

Notes:

1. There is a slight pause for initialization before the program begins.
2. To omit play list from program: Omit step 1 from User Instructions. Delete lines 140-280. Delete lines 5020-5190.

Reference: "FOOTBALL", HP-85 Basic, Hewlett-Packard.

PROGRAM LISTING

```
 10 ! FOOTBALL
 20 !
 30 ! Revision  11/01/82
 40 !
 50 DELAY 3
 60 RANDOMIZE
 70 INTEGER B(15),S8,R2,P8,Z9,D,T,S1,S2
    ,P1,P,S,G1,T1,Z
 80 INTEGER H9,E9,R,F1,L,G,D9,I,W
 90 INTEGER L(15),G(15)
100 DIM E$[187],D$[12],F$[55],R$[10],Q$
    [4],E2$[52]
110 DEF FNF(X) = SIN(PI*RND)

120 DISP TAB(12);'FOOTBALL'
130 S8,R2,P8,Z9,D,T,S1,S2,P1,P,S,G1,T1,
    Z=0
150 FOR I=1 TO 15
160 B(I)=0
170 NEXT I
180 RESTORE
190 FOR I=1 TO 5
200 READ F$[11*I-10,11*I]
210 NEXT I
220 DATA POINTS,FIRST DOWNS,YDS-RUSH,YD
    S-PASS,YDS-PENALTY
240 ON ERROR GOTO 250
250 INPUT 'Do you need a list of plays?
    ','N';Q$
260 Q$=UPRC$(Q$[1,1]) @ IF Q$#'Y' AND Q
    $#'N' THEN 250
270 OFF ERROR
280 IF Q$='Y' THEN GOSUB 5020
290 ON ERROR GOTO 300
300 INPUT 'Name of your team?',CHR$(95)
    ;R$
310 IF R$=CHR$(95) THEN 300
320 IF LEN(R$)>10 THEN R$=UPRC$(R$[1,10
    ]) ELSE R$=UPRC$(R$)
330 OFF ERROR
340 DISP '   Your opponent is the HP-75
    '
350 DISP
360 FOR I=1 TO 17
370 READ E$[11*I-10,11*I]
380 NEXT I
390 DATA DIVE,OFF TACKLE,SCISSORS,TRAP
400 DATA SWEEP,OPTION,REVERSE,DRAW,SIDE
    LINE
410 DATA LOOK-IN,ROLLOUT,SCREEN,FLY,POS
    T
420 DATA FIELD GOAL,PUNT,QUICK KICK
430 FOR I=1 TO 4
440 READ E2$[I*13-12,I*13]
```

-Function for generating random number

PROGRAM LISTING

```
450 NEXT I
460 DATA PR,OKIE,SHORT YARDAGE,PREVENT
470 FOR I=1 TO 4
480 READ D$[I*3-2,I*3]
490 NEXT I
500 DATA '1st','2nd','3rd','4th'
510 FOR I=1 TO 14
520 READ L(I),G(I)
530 NEXT I
540 DATA 2,7,3,10,5,15,5,20,3,10,7,15,9
    ,25
550 DATA 5,15,5,10,5,10,7,15,10,20,10,5
    0,10,40
560 H9=25
570 E9=50
580 T5,Q=1-2*IP(RND*2)
590 IF Q>0 THEN 620
600 DISP 'I won the toss and will recei
    ve.'
610 GOTO 690
620 DISP R$;' won the toss.'
630 ON ERROR GOTO 640
640 DISP CHR$(203);'ick or ';CHR$(210);
650 INPUT 'eceive?',CHR$(95);Q$
660 Q$=UPRC$(Q$[1,1])
670 OFF ERROR
680 ON POS('KR',Q$)+1 GOTO 630,710,690
690 Q=-Q
700 IF Z9>0 THEN 4920
710 X=50-10*Q
720 DISP 'Get ready for the kickoff !!'
730 FOR N=1 TO 20
740 Y=30+IP(40*FNF(1))
750 BEEP @ BEEP @ BEEP
760 DISP Y;'yard kickoff!'
770 X=X+Q*Y
780 IF Q=1 THEN 810
790 IF X>0 THEN 900
800 GOTO 820
810 IF X<100 THEN 900
820 DISP TAB(6);'* * * TOUCHBACK * * *'


830 D=0
840 IF P#15 THEN 870
850 X=X-Q*Y
860 IF ABS(50-X)<30 THEN 880
870 X=50+30*Q
880 Q=-Q
890 GOTO 950
900 IF P>14 THEN 920
910 R=IP(40*FNF(1))
920 Q=-Q
930 DISP R;'yard runback.'
```

-Yardage loss/gain data for offensive plays

-Touchback- ball will be brought out to the 20 yard line

PROGRAM LISTING

```
 940 X=X+Q*R
 950 R,P1=0
 960 GOSUB 980
 970 GOTO 1030
 980 IF X>50 THEN 1010
 990 DISP 'Ball on ';R$;X;'yd line'
1000 GOTO 1020
1010 DISP 'Ball on HP-75';ABS(X-100);'yd
      line'
1020 RETURN
1030 D=D+1
1040 GOTO 1090
1050 F1=ABS(X-(Q+1)/2*100)

1060 RETURN
1070 F2=ABS(X-(Q-1)/2*100)
1080 RETURN
1090 IF D#1 THEN 1200
1100 IF P8#0 THEN 1120
1110 S=0
1120 GOSUB 1050
1130 IF F1<=10 THEN 1190
1140 IF Q=-1 THEN 1170
1150 DISP '1st down ';R$;' ';ABS(10-S);'
     to go'
1160 GOTO 1230
1170 DISP '1st down HP-75 ';ABS(10-S);'t
     o go'
1180 GOTO 1230
1190 G1=1
1200 DISP D$[D*3-2,D*3];' down ';
1210 IF Q=-1 THEN DISP 'HP-75 ';ELSE DIS
     P R$;' ';
1220 IF G1<=0 THEN DISP ABS(10-S);'to go
     ' ELSE DISP ' goal to go'
1230 P8=0
1240 P1=P @ P=0
1250 GOSUB 4440
1260 IF Z>0 THEN 4340
1280 IF Q>0 THEN 1640
1290 GOTO 2260
1300 IF D>1 THEN 1330

1310 P=1+IP(3.5*RND)*4
1320 GOTO 1720
1330 IF D>2 THEN 1420
1340 IF ABS(10-S)>3 THEN 1370
1350 IF P1>12 THEN 1310
1355 P=1+IP(5*RND)
1360 GOTO 1720
1370 IF P>12 THEN 1310
1380 IF P1>8 THEN 1410
1390 P=P+1
1400 GOTO 1720
```

-Aid in determining distances to goals

-Begin analyzing defense (player) to choose offense

PROGRAM LISTING

```
1410 ON IP(2*RND)+1 GOTO 1310,1390
1420 IF D>3 THEN 1540
1430 IF ABS(10-S)>3 THEN 1460
1440 P=1+4*IP(2*RND)+IP(2*RND)
1450 GOTO 1720
1460 IF ABS(10-S)>6 THEN 1480
1470 GOTO 1390
1480 IF ABS(10-S)>11 THEN 1510
1490 P=P+1
1500 GOTO 1390
1510 IF ABS(10-S)<25 THEN 1350
1520 P=17
1530 GOTO 1720
1540 GOSUB 1050
1550 IF F1>40 THEN 1620
1560 IF ABS(10-S)>3 THEN 1580
1570 GOTO 1440
1580 GOSUB 1050
1590 IF F1>25 THEN 1620
1600 P=15
1610 GOTO 1720
1620 P=16
1630 GOTO 1720
1640 ON ERROR GOTO 1650
1650 INPUT 'Your play (1-17):',CHR$(95);
     P                                    -Player is offense - must enter
                                           play number
1660 IF P>17 THEN 1700
1670 IF P<1 THEN 1700
1680 OFF ERROR
1690 GOTO 1720
1700 DISP 'Try again.'
1710 GOTO 1640
1720 IF Q=1 THEN DISP R$;': ';ELSE DISP
     'HP-75: ';
1730 DISP E$[P*11-10,P*11]
1740 IF P<15 THEN 1760
1750 ON P-14 GOTO 1800,2040,2100
1760 L=L(P)
1770 G=G(P)
1780 IF Q<0 THEN 2325
1790 GOTO 2160
1800 ! FIELD GOAL ROUTINE.
1810 Y=IP(FNF(1)*50)
1820 GOSUB 1050
1830 IF Y>.6*F1 THEN 1860
1840 BEEP @ DISP '   Field goal is block
     ed !!!'
1850 GOTO 1930
1860 X=X+Q*Y
1870 IF Y<F1+10 THEN 1920
1880 IF .05+10/F1<RND THEN 1920
1890 BEEP @ DISP TAB(6);'Field goal is g
     ood !!!'
1900 P9=3
```

PROGRAM LISTING

```
1910 GOTO 4200
1920 BEEP @ DISP TAB(7);'Field goal fail
     ed.'
1930 D,S=0
1940 GOSUB 1050
1950 IF Y>F1 THEN 820
1960 R=IP(FNF(1)*Y)
1970 GOTO 2150
1980 IF RND>.2 THEN 2020
1990 DISP TAB(7);'Extra point failed.'
2000 P9=0
2010 GOTO 4340
2020 DISP TAB(6);'Extra point is good !!
     !'
2030 GOTO 4190
2040 !                                    -Punts
2050 Y=20+IP(30*FNF(1))
2060 DISP Y;'yard punt.'
2070 X=X+Q*Y
2080 R=IP(35*FNF(1))
2090 GOTO 2150
2100 !                                    -Kickoffs
2110 Y=30+IP(20*FNF(1))
2120 DISP Y;'yard kick.'
2130 X=X+Q*Y
2140 R=IP(10*FNF(1))
2150 GOTO 3040
2160 IF Q<0 THEN 2260
2170 GOSUB 1050
2180 IF F1>20 THEN 2210
2190 D9=3
2200 GOTO 2300
2210 IF (D=2 OR D=3) AND 10-S>5 THEN D9=  -Choose defense for 75. If 2nd
     4 ELSE D9=1+IP(2*RND)                 or 3d w/>5 to go PREVENT
2230 GOTO 2300
2260 ON ERROR GOTO 2270
2270 INPUT 'Enter defense (1-4):',CHR$(9  -Player is defense
     5);D9
2280 IF D9>4 OR D9<1 THEN 2270
2290 OFF ERROR
2300 IF Q=-1 THEN DISP R$;': ';ELSE DISP
      'HP-75: ';
2320 DISP E2$[D9*13-12,D9*13]
2325 IF P#0 THEN 2340
2330 IF Q=-1 THEN 1300
2340 IF P>8 THEN 2470
2350 IF P>4 THEN 2370
2360 ON P GOTO 2470,2380,2400,2400
2370 ON P-4 GOTO 2380,2420,2420,2440
2380 IF P1#1 THEN 2470
2390 GOTO 2450
2400 IF P1#2 THEN 2470
2410 GOTO 2450
2420 IF P1#5 THEN 2470
```

PROGRAM LISTING

```
2430 GOTO 2450
2440 IF P1#9 THEN 2470
2450 U=1
2460 GOTO 2480
2470 U=1.25
2480 IF D9#1 THEN 2510
2485 IF D9#1 THEN 2510
2490 IF P>5 THEN 2630
2500 ON P GOTO 2650,2630,2610,2630,2650
2510 IF D9#2 THEN 2540
2520 IF P>5 THEN 2630
2530 ON P GOTO 2630,2630,2610,2630,2650
2540 IF D9#3 THEN 2580
2550 IF P>5 THEN 2570
2560 ON P GOTO 2650,2650,2630,2630,2650
2570 IF P#9 THEN 2630
2580 IF P<13 THEN 2600
2590 GOTO 2650
2600 IF P#8 THEN 2630
2610 M=1.25
2620 GOTO 2660
2630 M=1
2640 GOTO 2660
2650 M=.8
2660 Y=IP((G*RND-L*RND)*(U*M))                    -Determine yardage
2670 IF R2#0 THEN 4130
2680 IF RND<.98 THEN 2720
2690 GOSUB 1050
2700 Y=F1+1
2710 GOTO 3630
2720 IF P>8 THEN 2900
2730 IF RND<.93 THEN 3070                         -Determine chances of fumble
                                                   for running plays
2740 Y=IP(.5*FNF(1)*Y)
2750 DISP 'Fumble after';
2760 IF Y<0 THEN 2790
2770 DISP Y;'yard gain.'
2780 GOTO 2830
2790 IF Y#0 THEN 2820
2800 DISP ' no gain.'
2810 GOTO 2830
2820 DISP ABS(Y);'yard loss.'
2830 IF 1-IP(2*RND)=0 THEN 2860                   -Determine outcome of fumble
2840 DISP '       Fumble recovered !'
2850 GOTO 3070
2860 DISP '          Fumble lost !'
2870 X=X+Q*Y
2880 Q=-Q
2890 GOTO 4280
2900 IF RND<.07 THEN 3010
2910 IF Y>0 THEN 2940
2920 DISP 'QB tackled for';
2930 GOTO 3630
2940 IF ABS(RND-.5)<.23 THEN 3630
```

PROGRAM LISTING

```
2950 IF 1P(RND*2)=0 THEN 2980
2960 DISP '         Batted down.'
2970 GOTO 2990
2980 DISP '          Incomplete.'
2990 Y=0
3000 GOTO 3070
3010 DISP '    * * * INTERCEPTION * * *'
3020 X=X+Q*Y
3030 R=IP(15*FNF(1))
3040 P=18
3050 D,S,G1=0
3060 GOTO 3630
3070 IF ABS(RND-.5)>.04 THEN 3630
3080 P8=1
3090 GOTO 3750
3100 IF IP(2*RND)=0 THEN 3170
3110 W=-1
3120 IF 1P(2*RND)=0 THEN 3150
3130 P8=15
3140 GOTO 3190
3150 P8=5
3160 GOTO 3210
3170 W=1
3180 GOTO 3120
3190 DISP '            HOLDING'
3200 GOTO 3220
3210 DISP '          OFFSIDE'
3220 DISP 'Penalty on ';
3230 IF W=1 THEN DISP R$;'.' ELSE DISP '
     HF-75.'
3240 IF W<0 THEN 3400
3250 IF Q<0 THEN 3350
3260 IF P8=15 THEN 3280
3270 IF Y<0 THEN 3370
3280 GOSUB 3470
3290 Y=P8*W
3300 DISP 'Penalty accepted: ';P8;'yards
     .'
3310 D=D-1
3320 X=X+Q*Y
3330 B(14-W)=B(14-W)+P8
3340 GOTO 4240
3350 IF S+Y>ABS(10-S) THEN 3370
3360 GOTO 3280
3370 P8=0
3380 GOTO 3630
3390 IF W>0 THEN 3260
3400 ON ERROR GOTO 3410
3410 INPUT 'Accept the penalty (Y/N)',CH
     R$(95);Q$
3420 Q$=UPRC$(Q$[1,1])
3430 IF Q$='N' THEN 3370
3440 IF Q$#'Y' THEN 3410
3450 OFF ERROR
```

-Passing errors

-Penalties

PROGRAM LISTING

```
3460 GOTO 3280
3470 IF Q>0 THEN 3570
3480 IF W>0 THEN 3530
3490 GOSUB 1070
3500 IF 2*P8<F2 THEN 3620
3510 P8=IP(F2/2)
3520 GOTO 3620
3530 GOSUB 1050
3540 IF 2*P8<F1 THEN 3620
3550 P8=IP(F1/2)
3560 GOTO 3620
3570 IF W>0 THEN 3600
3580 W=1
3590 GOTO 3530
3600 W=-1
3610 GOTO 3490
3620 RETURN
3630 IF P>14 THEN 3720
3640 X=X+Q*Y
3650 GOSUB 4650
3660 IF Q>0 THEN 3700
3670 IF X<1 THEN 3930
3680 IF X>99 THEN 3850
3690 GOTO 3720
3700 IF X>99 THEN 3930
3710 IF X<1 THEN 3930
3720 IF P>14 THEN 4310
3730 IF R=0 THEN 3750
3740 GOTO 930
3750 IF Y<=0 THEN 3790
3770 DISP Y;'yard gain.'
3780 GOTO 3830
3790 IF Y#0 THEN 3820
3800 DISP ' No gain.'
3810 GOTO 3830
3820 DISP ABS(Y);'yard loss.'
3830 IF P8#0 THEN 3100
3840 GOTO 4240
3850 DISP 'Safety vs ';
3860 IF Q=1 THEN DISP R$;'.' ELSE DISP '
     HP-75.'
3870 P9=2
3880 Q=-Q
3890 S8=1
3900 GOSUB 4590
3910 Q=-Q
3920 GOTO 4340
3930 DISP '       Touchdown ';
3940 IF Q=1 THEN DISP R$;'!' ELSE DISP '
     HP-75 !'
3950 P9=6
3960 GOSUB 4590
3970 IF Q>0 THEN 4030
3980 IF S2+1=S1 THEN 1980
```

-Safety

PROGRAM LISTING

```
3990 IF S2+2=S1 THEN 4010
4000 GOTO 1980
4010 P=4
4020 GOTO 4100
4030 DISP '       Kicking extra point.'
4040 GOTO 1980
4050 ON ERROR GOTO 4060
4060 INPUT 'Enter play (1-14):',CHR$(95)
     ;P
4070 IF P>14 OR P<1 THEN 4060
4080 OFF ERROR
4090 IF P>12 THEN 4220
4100 R2=1
4110 D9=2
4120 GOTO 2340
4130 R2=0
4140 IF Y<4.1 THEN 4220
4150 P9=2
4160 DISP '      Extra point is good !!'
4170 GOSUB 4590
4180 GOTO 4340
4190 P9=1
4200 GOSUB 4590
4210 GOTO 4340
4220 DISP '  Extra point attempt failed.
     '
4230 GOTO 4340
4240 S=S+Y
4250 IF S>9 THEN 4280
4260 IF D<4 THEN 4310
4270 Q=-Q
4280 D,S,G1=0
4290 IF P>14 THEN 4310
4300 B(5+Q)=B(5+Q)+1
4310 IF P=18 THEN 780
4320 GOSUB 980
4330 GOTO 1030
4340 DISP R$;':';S1;'   HP-75:';S2          -Scoreboard
4350 D,S,Z,P1,G1,P,P9=0
4360 IF S8#0 THEN 4390
4370 Q=-Q
4380 GOTO 690
4390 X=50-30*Q
4400 DISP 'Kickoff from';
4410 IF Q=1 THEN DISP R$;' 20 yard' ELSE
      DISP 'HP-75 20 yard'
4420 S8=0
4430 GOTO 740
4440 T=T+1
4450 IF T=H9-5 THEN 4540
4460 IF T=E9-5 THEN 4540
4470 IF T>H9 THEN 4490
4480 GOTO 4580
4490 IF T1>0 THEN 4560
```

PROGRAM LISTING

```
4500 IF RND(.35 THEN 4580
4510 GOSUB 4700
4520 Q=T5
4530 GOTO 4340
4540 BEEP @ DISP '> > > TWO-MINUTE WARNI
     NG < < <'
4550 GOTO 4580
4560 IF T<E9 THEN 4580
4570 IF RND>.5 THEN 4720
4580 RETURN
4590 B(2+Q)=B(2+Q)+P9
4600 IF Q>0 THEN 4630
4610 S2=S2+P9
4620 GOTO 4640
4630 S1=S1+P9
4640 RETURN
4650 IF P>8 THEN 4680
4660 B(8+Q)=B(8+Q)+Y
4670 GOTO 4690
4680 B(11+Q)=B(11+Q)+Y
4690 RETURN
4700 DISP '       * * * HALFTIME * * *'
4710 GOTO 4740
4720 DISP '     * * * END OF GAME * * *'
4730 Z9=1
4740 ON ERROR GOTO 4750
4750 INPUT 'View statistics?'; Q$ @ Q$=U
     PRC$(Q$[1,1])
4760 OFF ERROR
4770 IF Q$='Y' THEN 4800
4780 IF Q$#'N' THEN 4750
4790 GOTO 4890
4800 DISP 'STATISTICS FOR ';R$
4810 IMAGE 11A,3X,4D
4820 FOR I=0 TO 4
4830 DISP USING 4810 ; F$[(I+1)*11-10,(I
     +1)*11],B(3+I*3)
4840 NEXT I
4850 DISP 'STATISTICS FOR HP-75'
4860 FOR I=0 TO 4
4870 DISP USING 4810 ; F$[(I+1)*11-10,(I
     +1)*11],B(1+I*3)
4880 NEXT I
4890 Z,T1=1
4900 T=H9
4910 RETURN
4920 ON ERROR GOTO 4930
4930 INPUT 'Another game?','Y';Q$
4940 Q$=UPRC$(Q$[1,1])
4950 OFF ERROR
4960 IF Q$='N' THEN 5010
4970 IF Q$#'Y' THEN 4920
4980 Z9,D,T,S1,S2,P1,P,S,G1,T1,Z=0
4995 FOR I=1 TO 15 @ B(I)=0 @ NEXT I
```

-Statistics review for halftime and end of game

PROGRAM LISTING

```
5000 GOTO 560
5010 STOP
5020 !
5030 DISP 'Call plays as follows ...'
5040 DISP '         * * * RUNS * * *'
5050 DISP '1 = DIVE' @ DISP '2 = OFF TAC
     KLE'
5060 DISP '3 = SCISSORS' @ DISP '4 = TRA
     P'
5070 DISP '5 = SWEEP' @ DISP '6 = OPTION
     '
5080 DISP '7 = REVERSE' @ DISP '8 = DRAW
     '
5090 DISP '       * * * PASSES * * *'
5100 DISP '9 = SIDELINE' @ DISP '10 = LO
     OK-IN'
5110 DISP '11 = ROLLOUT' @ DISP '12 = SC
     REEN'
5120 DISP "13 = 'FLY'" @ DISP "14 = 'POS
     T'"
5130 DISP '         * * * KICKS * * *'
5140 DISP '15 = FIELD GOAL' @ DISP '16 =
      PUNT'
5150 DISP '17 = QUICK KICK'
5160 DISP 'Call defenses as follows ...'
5170 DISP "1 = 'PR' (4-3)" @ DISP "2 = '
     OKIE' (5-2)"
5180 DISP "3 = 'SHORT YARDAGE'" @ DISP "
     4 = 'PREVENT'"
5190 RETURN
```

-Key to plays

PROGRAM DESCRIPTION

GOLF

This game is a simulation of an 18-hole golf course. The player is supplied with 12 clubs (a driver, 2 woods, 8 irons, and putter) to tackle the course. In addition, he may choose to use a partial swing (i.e. a percent of his full swing) on any of the irons (see club list below).

Distances for each hole range from 180 yards to 560 yards, with pars of 3, 4 or 5 strokes. Par for the course is 72. Hazards include trees, sand-traps and water.

To begin the game the player must enter his handicap (0-30) and indicate his worst difficulty at golf: hook, slice, poor distance, trapshot, or putt. Then the distance to the hole, the par for that hole, and the conditions of the right and left sides of the fairway are displayed. The player then proceeds as on a golf course, selecting appropriate clubs for particular shots. If he hits a ball into the water or out of bounds he must hit again from the previous location and a penalty stroke is assessed. If the player selects a partial swing club, he must enter the percent of a full swing that he desires (1-99%).

When the player reaches the green, he must enter a putt potency number (>0). For example, a distance of 3 feet to the pin (hole) suggests a putt potency number of around 1.

CLUBS:	WOODS	IRONS (FULL)		IRONS (PARTIAL)	
	1) Driver	12) 2 Iron	16) 6 Iron	22) 2 Iron	26) 6 Iron
	2) 2 Wood	13) 3 Iron	17) 7 Iron	23) 3 Iron	27) 7 Iron
	3) 3 Wood	14) 4 Iron	18) 8 Iron	24) 4 Iron	28) 8 Iron
		15) 5 Iron	19) 9 Iron	25) 5 Iron	29) 9 Iron

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		WELCOME TO THE GOLF COURSE	
1	Enter handicap (0-30)	What is your handicap?	0-30 [RTN]
2	Enter worst difficulty at golf	Choose your worst difficulty ...	
		Hook,Slice,Dist.,Trap or Putt?	H,S,D,T, or P [RTN]
3	Display conditions of hole e.g.:	You are on tee of hole 1	
		Distance 361 yards PAR 4	
		On right is adjacent fairway	
		On left is rough	
4	Enter club: woods 1-3,	What club do you want?	1-3,12-19,or 22-29 [RTN]
	irons 12-19,		
	or if partial swing 22-29:	Percent of full swing (1-99)?	1-99 [RTN]
5	If ball landed on green		
	then step 6		
5a	Display flight and distance		
	of ball and new location e.g.:	You sliced	
		Shot went 228 yards ...	
		133 yards from the hole	
		Ball is 11 yards off line ...	
		... in adjacent fairway	
	Goto step 4		
6	Ball on green	On green # feet from pin	
7	Enter strength of putt	Putt potency number?	p [RTN]
	must be greater than 0		
7a	If ball went in hole then		
	step 8 else	Passed by cup	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	Goto step 6		
8		You holed it!!!	
	Display number of strokes, and	Score on hole # was # #	
	total strokes so far:		
	IF PAR:	A Par. Nice going!	
	if PAR - 1:	A Birdie. Very good!	
	if PAR - 2:	An Eagle! Excellent!!!	
	if 1 stroke:	*** A HOLE IN ONE ***	
	if PAR + 1 or + 2:	Keep your head down	
9	To continue game goto step 3		
	for next hole and new condi-		
	tions. If hole 18 has been		
	played, then goto step 10.		
10	Display total score:	Total score for 18 holes was ##	
	if PAR:	Par game!	
	if <PAR:	## under par!!!	
	if >PAR:	## over par	
11	Option to play again:	Would you like to play again?Y	[RTN] or N [RTN]

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
L(1)	Location of ball	T	Difficulty at golf
L(2)	Terrain on right side of fairway	T5	Total par for course
L(3)	Terrain on left side of fairway	Z	Indicates use of subroutine
W	Percent of full swing on irons	H	Handicap
S1	Score on a particular hole	Q	Aid in determining dubbed shot
S2	Accumulated score	X	Index for terrain subroutine
F	Number of hole	J	Determines penalty stroke
D	Distance from tee to hole	K	Counts putts
P	Par for a hole	N	Random number used to find probability of getting out of trap
D2	Current distance to hole		
D1	Distance of shot	B( )	Temp. storage for D
O	Yards off line	I	Putt strength
C	Club number	T$	Golf difficulty
		Q$	User interaction

NOTES AND REFERENCES

Notes: 1. To omit club list from program change:

580 if C>=1 and C<=29 then 600

630!

Delete 2220-2490

Reference: 1. "GOLF", HP-85, Hewlett-Packard.

PROGRAM LISTING

```
 10 ! GOLF
 20 !
 30 ! Revision 11/01/82
 40 INTEGER L(11),J,K,Q,X,B,H,Z,S1,S2,F
    ,O,P,D2,D1,C,T,T5
 50 REAL N,O,W,I
 60 DIM T$[4],Q$[4]
 70 RANDOMIZE
 80 DELAY 2.5
 90 DISP '  WELCOME TO THE GOLF COURSE.
    '
100 X=3 @ N=.8 @ F=1
110 S1,S2,T5=0
120 ON ERROR GOTO 130
130 INPUT 'What is your handicap?';H
140 IF H>30 OR H<0 THEN 130                  -PGA rules - handicap 0-30
150 ON ERROR GOTO 170
160 DISP 'Choose your worst difficulty.
    .'
170 DISP CHR$(200);'ook ';CHR$(211);'li
    ce ';CHR$(196);'ist. ';CHR$(212);'r
    ap or ';CHR$(208);
180 INPUT 'utt?',CHR$(95);T$
190 T$=UPRC$(T$[1,1])
200 ON POS('HSDTP',T$)+1 GOTO 170,210,2
    20,230,240,250
210 T=1 @ GOTO 260
220 T=2 @ GOTO 260
230 T=3 @ GOTO 260
240 T=4 @ GOTO 260
250 T=5
260 OFF ERROR
270 !
280 J,Q,K,L(1)=0
290 S2=S2+S1
300 IF F=1 THEN 460
310 DISP 'Score on hole';F-1;'was';S1;T
    AB(29);S2
315 ON ERROR GOTO 330
320 ON P-S1+3 GOTO 450,450,350,370,390   -Determines comparative skill
                                          on hole
330 DISP 'Keep your head down.'
340 GOTO 450
350 DISP 'A Par.  Nice going!'
360 GOTO 450
370 DISP 'A Birdie.  Very good !!'
380 GOTO 450
390 IF P=3 THEN 420
400 DISP 'An Eagle!  Excellent!!!'
410 GOTO 450
420 DISP '   * * * A HOLE IN ONE * * *'
430 BEEP @ BEEP @ BEEP @ BEEP
450 OFF ERROR
455 IF F=19 THEN 1970                        -Check for end of game
```

PROGRAM LISTING

```
460 S1=0
470 IF S1=0 THEN 1800
480 IF L(1)<1 THEN 1330
490 X=1
500 IF L(1)>5 THEN 1370                    -Determine penalty shot
510 DISP 'Shot went';D1;'yards...'
520 DISP D2;'yards from the hole.'
530 DISP 'Ball is';IP(O);'yards off lin
    e...'
540 Z=1
550 GOSUB 2060
560 ON ERROR GOTO 570
570 INPUT 'What club do you want?';C
580 IF C<1 OR C>29 THEN GOSUB 2220 ELSE    -Check for legal club selection
     600
590 GOTO 560
600 OFF ERROR
610 IF C>3 THEN 650
620 IF L(1)<5 OR C=14 OR C=23 THEN 680
630 GOSUB 2220
640 GOTO 560
650 IF C<12 THEN 630
660 C=C-6
670 GOTO 620
680 S1=S1+1
690 W=1
700 IF C>13 THEN 1010
710 IF F/3=IP(F/3) THEN 970
720 IF C<4 THEN 740
730 GOTO 750
740 IF L(1)=2 THEN 830
750 IF S1<7 THEN 860
760 D1=IP((30-H)*2.5+187-((30-H)*.25+15
    )*C/2+25*RND)
770 D1=IP(D1*W)
780 IF T=2 THEN 1350
790 O=RND/.8*(2*H+16)*ABS(TAN(D1*.0035)
    )
800 D2=IP(SQR(O^2+ABS(D-D1)^2))
810 IF D-D1<0 THEN 880
820 GOTO 900
830 DISP 'You dubbed it.'
840 D1=35
850 GOTO 790
860 IF D<200 THEN 1480
870 GOTO 760
880 IF D2<20 THEN 900
890 DISP 'Too much club...passed the ho
    le.'
900 B=D
910 D=D2
920 IF D2>27 THEN 1150
930 IF D2>20 THEN 1280
940 IF D2>.5 THEN 1300
```

PROGRAM LISTING

```
 950 L(1)=9
 960 GOTO 1690
 970 IF (72+(H+1)/.85)/18(S2+Q+10*(F-1)/
     18 THEN 720
 980 Q=Q+1
 990 IF S1/2#IP(S1/2) THEN 1100
1000 GOTO 830
1010 ON ERROR GOTO 1020
1020 INPUT 'Percent of full swing (1-99)
     :';W
1030 W=W/100
1040 IF W(=0 OR W)=1 THEN 1020
1050 OFF ERROR
1060 IF L(1)=5 THEN 1470
1070 IF C=14 THEN 750
1080 C=C-10
1090 GOTO 750
1100 IF D(95 THEN 830
1110 DISP 'Ball hit tree...went into rou
     gh'
1120 DISP D-75;'yards from hole.'
1130 D=D-75
1140 GOTO 570
1150 IF O)30 OR J)0 THEN 1330
1160 IF 1)0 THEN 1220
1170 IF (S2+1)/15=IP((S2+1)/15) THEN 123
     0
1180 DISP 'You hooked ';
1190 L(1)=L(3)
1200 IF O)45 THEN 1260 ELSE DISP
1210 GOTO 470
1220 IF (S2+1)/15=IP((S2+1)/15) THEN 118
     0
1230 DISP 'You sliced ';
1240 L(1)=L(2)
1250 GOTO 1200
1260 DISP 'badly.'
1270 GOTO 470
1280 L(1)=5
1290 GOTO 470
1300 L(1)=8
1310 D2=IP(D2*3)
1320 GOTO 1560
1330 L(1)=1
1340 GOTO 470
1350 D1=IP(.85*D1)
1360 GOTO 790
1370 IF L(1))6 THEN 1450
1380 DISP 'Your shot went into the water
     .'
1390 S1=S1+1
1400 DISP 'Penalty stroke assessed.' @ D
     ISP 'Hit from previous location.'
1410 J=J+1
```

PROGRAM LISTING

```
1420 L(1)=1
1430 D=B
1440 GOTO 560
1450 DISP 'Your shot went out of bounds.
     '
1460 GOTO 1390
1470 !
1480 D2=1+3*IP(80/(40-H)*RND)
1490 GOTO 1560
1500 IF RND>N THEN 1540
1510 N=N*.2
1520 DISP 'Shot dubbed...still in trap.'
1530 GOTO 560
1540 N=.8
1550 GOTO 1480
1560 DISP 'On green';D2;'feet from pin.'
1570 ON ERROR GOTO 1580
1580 INPUT 'Putt potency number?';I
1590 OFF ERROR
1600 IF I<=0 THEN 1570
1610 S1=S1+1
1620 IF S1+1-P>H*.072+2 THEN 1690
1630 IF K>2 THEN 1690
1640 K=K+1
1650 IF T=4 THEN 1750
1660 D2=D2-I*(4+2*RND)+1.5
1670 IF D2<-2 THEN 1770
1680 IF D2>2 THEN 1720
1690 DISP 'You holed it!!!'
1700 F=F+1
1710 GOTO 280
1720 DISP 'Putt short.'
1730 D2=IP(D2)
1740 GOTO 1560
1750 D2=D2-I*(4+RND)+1
1760 GOTO 1670
1770 DISP 'Passed by cup.'
1780 D2=-D2
1790 GOTO 1730
1800 READ D,P,L(2),L(3)

1810 IF F-1>18 THEN 1970
1820 T5=T5+P
1830 DISP 'You are on tee of hole';F
1840 DISP 'Distance';D;'yards   PAR';P
1850 DISP 'On right is ';
1860 Z=0
1870 X=2
1880 GOSUB 2060
1890 DISP 'On left is ';
1900 X=3
1910 GOSUB 2060
1920 DISP
1930 GOTO 560
```

-Read in distance for hole, the par and fairway condition

PROGRAM LISTING

```
1940 DATA 361,4,4,2,389,4,3,3,206,3,4,2,
     500,5,7,2,408,4,2,4,359,4,6,4
1950 DATA 424,4,4,2,388,4,4,4,196,3,7,2,
     400,4,7,2,560,5,7,2,132,3,2,2
1960 DATA 357,4,4,4,294,4,2,4,475,5,2,3,
     375,4,4,2,180,3,6,2,550,5,6,6
1970 DISP 'Total score for';F-1;'holes w
     as';S2
1980 IF S2-T5 THEN 2010
1990 DISP 'Par game !'
2000 GOTO 2500
2010 IF S2-T5>0 THEN 2040
2020 DISP -(S2-T5);'under par!!!'
2030 GOTO 2500
2040 DISP S2-T5;'over par.'
2050 GOTO 2500
2060 IF Z=1 AND L(X)#7 THEN DISP '    ...
      in ';
2070 ON L(X) GOTO 2100,2120,2140,2160,21
     80,2200,2080
2080 DISP 'out of bounds.'
2090 RETURN
2100 DISP 'fairway.'
2110 RETURN
2120 DISP 'rough.'
2130 RETURN
2140 DISP 'trees.'
2150 RETURN
2160 DISP 'adjacent fairway.'
2170 RETURN
2180 DISP 'trap.'
2190 RETURN
2200 DISP 'water.'
2210 RETURN
2220 ON ERROR GOTO 2230

2230 INPUT 'Need a list of clubs?','Y';Q
     $
2240 Q$=UPRC$(Q$[1,1])
2250 IF Q$#'Y' THEN 2490
2260 DISP '    WOODS *** Full swing only'
2270 DISP '    1)  DRIVER'
2280 DISP '    2)  2 WOOD'
2290 DISP '    3)  3 WOOD'
2300 DISP '    IRONS *** Full swing only'
2310 DISP '    12)  2 IRON'
2320 DISP '    13)  3 IRON'
2330 DISP '    14)  4 IRON'
2340 DISP '    15)  5 IRON'
2350 DISP '    16)  6 IRON'
2360 DISP '    17)  7 IRON'
2370 DISP '    18)  8 IRON'
2380 DISP '    19)  9 IRON'
2390 DISP ' IRONS *** Partial swing only
```

-Data for course

-Check for use of preposition in subroutine

-Subroutine to display club choices

PROGRAM LISTING

```
2400 DISP '    22)  2 iron'
2410 DISP '    23)  3 iron'
2420 DISP '    24)  4 iron'
2430 DISP '    25)  5 iron'
2440 DISP '    26)  6 iron'
2450 DISP '    27)  7 iron'
2460 DISP '    28)  8 iron'
2470 DISP '    29)  9 iron'
2480 OFF ERROR
2490 RETURN
2500 ON ERROR GOTO 2510
2510 INPUT 'Would you like to play again
     ?','Y';Q$
2520 Q$=UPRC$(Q$[1,1])
2530 IF Q$='Y' THEN 100
2540 IF Q$#'N' THEN 2510
2550 OFF ERROR
2560 DELAY 1
2570 DISP TAB(9);'END OF GAME' @ DISP
2580 STOP
```

PROGRAM DESCRIPTION

HAMURABI

This game allows a player to control a country's economy through the buying and selling of land. The more efficiently he uses the land, the better a governor he is.

The game begins with a report of the economy and population. The player then has the opportunity to buy and sell land, allocate food, and plant a number of acres for harvest. The player must deal with plagues, starvation, rats, and the rise and fall of the land market.

If the player rules unsuccessfully, he will be warned and asked if he wishes to continue. He may resign at any time by selling all his land.

The object of this game is to determine how it works and find the best set of circumstances for a growing economy.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		HAMURABI	
1	Report of economy	I beg to report that last year:	
		x people starved . . .	
		x people came to the city:	
1a	A plague has a probability of		
	1:10. If there was one:	The plague killed ½ the people	
1b	Population	Population now xx	
1c	Harvest	We harvested xx bushels . . .	
		at xx bushels per acre	
1d	Rats	Rats destroyed xx bushels	
1e	Total food	xxx bushels in storehouses	
1f	Land	The city owns xxx acres . . .	
	Land will sell at . . .	Worth xx bushels/acre	
2	Decisions must be made	Hamurabi . . .	
		Buy how many acres?	xx [RTN]
	If insufficient bushels to		
	buy land then display stats		
		Sell how many acres?	xx [RTN]
	If insufficient acres to sell		
	then display stats		
	If all land sold goto step 5		
		How many bushels for food?	xx [RTN]
	If insufficient bushels in		
	store then display stats		
		How many acres to plant?	xx [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	If insufficient acres to plant		
	then display statistics		
	If many people died then:	Things aren't looking too well!	
	and goto step 4 else step 1		
3	STATS = Review economic status	Think again ... you only have:	
		xx people ...	
		xxx acres and ...	
		xxx bushels in storehouses	
4	Option to continue govening	You have ruled for x years	
	under current conditions:	Do you want to continue?	Y [RTN] or N [RTN]
	If 'Y' then continue		
	If 'N' then step 5		
5	Options to begin again:	Start over?	Y [RTN] or N [RTN]
	If 'Y' then step 1		
	If 'N' then end	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A1	Population	B4	Total harvest
A2	Immigrants	C1	Acres
A3	Deaths	J	Temporary storage for comparison
B1	Bushels in storehouses	I	Input variable
B2	Bushels destroyed by rats	Z	Number of years
B3	Harvest per acre	C2	Value of an acre
		Q$	User interaction

NOTES AND REFERENCES

Reference: "HAMURABI", HP-2000 Basic, Hewlett-Packard

PROGRAM LISTING

```
 10 ! HAMURABI
 20 !
 30 ! Revision  11/01/82
 40 !
 50 INTEGER A1,A2,A3,B1,B2,B3,B4,C1,J,Z
 60 DELAY 5
 70 DISP TAB(12);'HAMURABI'
 80 A1=100 @ A2=5 @ A3=0
 90 B1=2800 @ B2=200 @ B3=3 @ B4=3000
100 C1=1000 @ J=1 @ Z=0
110 ! REPORT TO HAMURABI.
120 IF Z=1 THEN 140
130 IF A3>(A1+A3)*.45 THEN DISP "Things
     aren't looking too well!" @ GOSUB
    1010
140 BEEP 600,.15 @ BEEP 400,.1 @ BEEP 6
    00,.15
150 Z=Z+1
160 IF Z/5=IP(Z/5) THEN GOSUB 1010

170 DISP 'I beg to report that last yea
    r:'
180 DISP A3;'people starved and ...'
190 DISP A2;'people came to the city.'
200 IF J>0 THEN 230

210 A1=A1-IP(A1/2)
220 DISP 'The plague killed 1/2 the peo
    ple'
230 DISP 'Population now';A1;'.'
240 DISP 'We harvested';B4;'bushels...'
250 DISP 'at';B3;'bushels per acre.'
260 DISP 'Rats destroyed';B2;'bushels.'
270 DISP B1;'bushels in storehouses.'
280 DISP 'The city owns';C1;'acres...'
290 C2=17+IP(6*RND)

300 DISP 'worth';C2;'bushels/acre.'
310 BEEP 600
320 DISP 'Hamurabi ...'
330 ! BUY LAND?
340 ON ERROR GOTO 350
350 INPUT 'Buy how many acres?',CHR$(95
    );I
360 I=IP(ABS(I))
370 OFF ERROR
380 IF I=0 THEN 450
390 J=I*C2
400 IF J<=B1 THEN 430
410 GOSUB 940
420 GOTO 340
430 B1=B1-J
440 C1=C1+I
450 ! SELL LAND?
```

-Set up initial conditions

-If more than 45% of the people died, option to restart

-If 5 years have passed give option to quit or continue

-Plagues have a probability of 1 in 10

-Status report

-Acreage ranges in value from 17 to 22 bushels per acre

-Option to buy land

PROGRAM LISTING

```
460 ON ERROR GOTO 470
470 INPUT 'Sell how many acres?',CHR$(9
    5);I
480 I=IP(ABS(I))
490 OFF ERROR
500 IF I=0 THEN 570
510 IF I<C1 THEN 550
520 IF I=C1 THEN 1090                    -Check to see if selling all
                                          land - option to restart
530 GOSUB 940
540 GOTO 460
550 C1=C1-I
560 B1=B1+C2*I
570 ! BUY FOOD?                          -Food allocation
580 ON ERROR GOTO 590
590 INPUT 'How many bushels for food?',
    CHR$(95);I
600 I=IP(ABS(I))
610 OFF ERROR
620 IF I<=B1 THEN 650
630 GOSUB 940
640 GOTO 580
650 B1=B1-I
660 A3=A1-IP(I/20)
670 A2=0
680 IF A3>=0 THEN 710
690 A2=-A3/2
700 A3=0
710 ! PLANT ACRES?
720 ON ERROR GOTO 730
730 INPUT 'How many acres to plant?',CH
    R$(95);I
740 I=IP(ABS(I))
750 OFF ERROR
760 IF I>C1 THEN 790
770 J=IP(I/2)
780 IF J<=B1 THEN 810
790 GOSUB 940
800 GOTO 720                             -Plagues have a probability of
                                          1 in 10
810 IF I>10*A1 THEN 790
820 B1=B1-J
830 ! HARVEST,RATS,POPULATION.
840 B3=IP(5*RND)+1
850 B4=B3*I
860 B2=IP((B1+B4)*.07*RND)
870 B1=B1-B2+B4
880 J=IP(10*RND)
890 A2=IP(A2+(5-B3)*B1/600+1)
900 IF A2<=50 THEN 920
910 A2=50
920 A1=A1+A2-A3
930 GOTO 110
940 ! ERROR ROUTINE.
```

PROGRAM LISTING

```
 950 BEEP 100,.2
 960 DISP 'Think again ... you only have
     :'
 970 DISP A1;'people ...'
 980 DISP C1;'acres and ...'
 990 DISP B1;'bushels in storehouses.'
1000 RETURN
1010 DISP 'You have ruled for';Z;'years.
     '
1020 ON ERROR GOTO 1030
1030 INPUT 'Do you want to continue?';Q$
1040 Q$=UPRC$(Q$[1,1])
1050 OFF ERROR
1060 IF Q$='N' THEN POP @ GOTO 1090
1070 IF Q$#'Y' THEN 1020
1080 RETURN
1090 ON ERROR GOTO 1100
1100 INPUT 'Start over?'; Q$ @ Q$=UPRC$(
     Q$[1,1])
1110 OFF ERROR
1120 IF Q$='Y' THEN 80
1130 IF Q$#'N' THEN 1090
1140 DELAY 1
1150 DISP TAB(10);'END OF GAME' @ DISP
1160 STOP
```

PROGRAM DESCRIPTION

REVERSE

The object of the game is to order a series of numbers from lowest to highest by reversing the order of a subset of numbers. For example, a series of five numbers, such as 25314, might be ordered by the following moves:

```
    Start : 25314
Reverse 2 : 52314  (reverses the two leftmost numbers)
Reverse 5 : 41325
Reverse 4 : 23145
Reverse 2 : 32145
Reverse 3 : 12345  and the numbers are in order.
```

In this example, it took 5 reversals to win. For n numbers, a solution can always be found in n+1 moves or less. The program allows the player to make any number of moves necessary.

The game allows a series of 3 to 9 numbers to be used in the game.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		*** REVERSE ***	
1	Choose size of game (3-9)	How many numbers to reverse? 5	3-9 [RTN]
2	Display n numbers in random	n_1 n_2 n_3 . . . n_n #	2-n [RTN]
	order. Allow player to reverse		
	the order of the first r		
	numbers. *		
3	Display n numbers in their	n_1 n_2 n_3 . . . n_n #	r [RTN]
	new order. Allow player to		
	reverse the order of the first		
	r numbers. *		
4	Repeat step 3 until the		
	numbers are in their correct		
	order from smallest to		
	largest.		
5	Display correct order	n_1 n_2 n_3 . . . n_n	
6	Display number of reversals		
	taken: if ≤n:	WOW! You won in m moves!	
	if = n+1:	You won in m moves!	
	if >n+1:	It took you m moves	
7	Option to play again	Run again, or End? R	E [RTN] or [RTN]
	If 'R' then step 1 else	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
R	Number of digits to reverse	I	Index
N	Number of digits	K	Index for mixing numbers
A( )	Numbers in mixed order	J	Looping
T	Temp. storage of numbers	R$	Number of digits to reverse
V	Number of reversals made	N$	Number of digits
Q$	Option to run again		

PROGRAM LISTING

```
 10 ! REVERSE - game to reverse
 20 ! the order of a set of
 30 ! numbers.
 40 ! Revision  11/01/81
 50 OPTION BASE 1
 60 DELAY 1
 70 RANDOMIZE
 80 DIM R$[3],Q$[3],N$[3]
 90 INTEGER A(9),N,I,T,V,R,K,J
100 DISP '       * * * REVERSE * * *' @
    WAIT 2
110 ON ERROR GOTO 120
120 INPUT 'How many numbers to reverse?
    ','5'; N$ @ N=VAL(N$)
130 IF N$[1,1]='0' THEN 590
140 IF N<=2 OR N>9 THEN 120
150 OFF ERROR
160 FOR I=1 TO N
170 A(I)=I
180 NEXT I
190 FOR I=N TO 2 STEP -1
200 K=IP(I*RND+1)

210 T=A(I)
220 A(I)=A(K)
230 A(K)=T
240 NEXT I
250 FOR I=1 TO N

260 IF A(I)$I THEN 290
270 NEXT I
280 GOTO 190
290 V=1
300 GOSUB 670
310 ON ERROR GOTO 300
320 INPUT '#',CHR$(95); R$ @ IF R$[1,1]
    =CHR$(95) THEN 590
330 R=VAL(R$)
340 IF R<=1 OR R>N THEN 300
350 OFF ERROR
360 V=V+1
370 FOR I=1 TO IP(R/2)

380 T=A(I)
390 A(I)=A(R-I+1)
400 A(R-I+1)=T
410 NEXT I
420 FOR I=1 TO N
430 IF A(I)$I THEN 300

440 NEXT I
450 DISP TAB(IP((32-N*3)/2));
460 FOR I=1 TO N
470 DISP A(I);
```

-Set up random index for mixing the order of the numbers

-Check to be sure that numbers are not still in order

-Enter the number of digits that you want to reverse

-Adjust the order of the numbers

-Check to see if solution has been found

-Display solution

PROGRAM LISTING

```
480 NEXT I
490 FOR J=1 TO 3
500 FOR I=1 TO 5
510 BEEP 502.857+I*(220/7),.02
520 NEXT I
530 NEXT J
540 WAIT 1 @ DISP
550 IF V-1<=N THEN 580
560 IF V-2=N THEN DISP '   You won in';
    V-1;'moves !' ELSE DISP '   It took
     you';V-1;'moves.'
570 GOTO 590
580 DISP 'W O W !  You won in';V-1;'mov
    es!'
590 WAIT 2
600 ON ERROR GOTO 610
610 DISP CHR$(210);'un again, or ';CHR$
    (197);
620 INPUT 'nd? ','R'; Q$ @ Q$=UPRC$(Q$[
    1,1])
630 ON POS('RE',Q$)+1 GOTO 590,110,640
640 DISP '           END OF GAME' @ WAIT
     2 @ DISP
650 OFF ERROR
660 STOP
670 ! DISPLAY NUMBERS.
680 DISP TAB(IP((32-N*3)/2)-1);
690 FOR I=1 TO N
700 DISP A(I);
710 NEXT I
720 RETURN
```

PROGRAM DESCRIPTION

SLOT MACHINE

This program simulates the actions of a slot machine. The user may choose from five machines (coin denominations) to play. The coin denominations are (1) $ .05, (2) $ .10, (3) $ .25, (4) $ .50, (5) $1.00. The higher the coin, the higher the payoff (or loss). Each machine uses the following payoff system:

&	cherry	&&&	Pays 8 times
o	orange	&&X	Pays 5 times
*	lemon	ooo	Pays 10 times
σ	plum	***	Pays 15 times
♣	bell	σσσ	Pays 20 times
=	bar	♣♣♣	Pays 50 times
		===	Pays 100 times
		&XX	Pays 2 times
		AA not=	Pays 3 times
		AA=	Pays 5 times

Note: 'X' represents any symbol except &. 'A' represents any symbol except & and is the same symbol for both occurrences of 'A' in a line.

At the end of each play, the user has the opportunity to "pull" again, or change machines, or stop.

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		SLOT MACHINE	
1	Select coin size	Choose slot machine (1-5): ▮	1-5 [RTN]
2	Assign the corresponding	You'll be using the .bb machine	
	coin value to the machine.		
3	Pull the handle	▒ ▒ ▒	
4	View roll, winnings or losses,	X X X win .nn $n.nn	
	and current cash total in hand	or	
	(X represents a symbol)	X X X lose $n.nn	
5	Option to pull again:	Pull or Stop? P	[RTN] or S [RTN]
	If 'P' then goto step 3		
	If 'S' then goto step 6		
6	Option to change machines:	Try another machine? Y	[RTN] or N [RTN]
	If 'Y' then step 1		
	If 'N' then step 7		
7	View total winnings or losses:	Total losses = n.nn	
		or	
		You broke even	
		or	
		Total winnings = n.nn	
8	End	END OF GAME	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
X( )	Represents each symbol for a roll	K	" ▒ " decimal code
I	Index and looping	B	Coin denomination
P	Payoff value	S	Total cash in hand
J	Tab values	Q$	User interaction

NOTES AND REFERENCES

Note: A players current cash total is displayed in the right most corner of the display window. If he changes machines this total stays with him.

PROGRAM LISTING

```
 10 ! SLOT - SLOT MACHINE.
 20 !
 30 ! Revision  11/01/82
 40 !
 50 DIM Q$[4]
 60 INTEGER X(3),I,P,J,K
 70 SHORT B,S
 80 DELAY 2
 90 RANDOMIZE
100 DISP '            SLOT MACHINE'
110 S=0
120 ON ERROR GOTO 130
130 INPUT 'Choose slot machine (1-5):',
    CHR$(95);Q$
140 Q$=UPRC$(Q$[1,1])
150 B=VAL(Q$)
160 DISP "You'll be using the ";
170 IF B=1 THEN B=.05 @ GOTO 230

180 IF B=2 THEN B=.1 @ GOTO 230
190 IF B=3 THEN B=.25 @ GOTO 230
200 IF B=4 THEN B=.5 @ GOTO 230
210 IF B=5 THEN B=1 @ GOTO 230
220 GOTO 120
230 DISP USING 240 ; B
240 IMAGE d.dd,' machine'
250 OFF ERROR
260 !
270 DELAY 0
280 K=31

290 FOR I=1 TO 5
300 DISP TAB(2);CHR$(K);
310 DISP TAB(5);CHR$(K);
320 DISP TAB(8);CHR$(K);
330 BEEP 141.429,.03
340 BEEP 172.857,.03
350 BEEP 204.286,.03
360 BEEP 235.714,.03
370 DISP CHR$(27)&'E';

380 NEXT I
390 J=2
400 DISP
410 DELAY 5
420 FOR I=1 TO 3
430 DISP TAB(J); @ BEEP 439.999
440 X(I)=IP(8*RND+1)

450 IF X(I)>5 THEN 650
460 IF X(I)>2 THEN 540
470 IF X(I)=2 THEN 510
480 DISP CHR$(18);
490 X(I)=1
```

-Establish coin size for machine selected

-Decimal code for upside down card

-Clear display and move to column 1

-Set up tab values

-Select a random number to correspond with a symbol

-cherry symbol

PROGRAM LISTING

```
500 GOTO 760
510 DISP 'O';                          -Orange
520 X(I)=2
530 GOTO 760
540 IF X(I)>3 THEN 580
550 DISP '*';                          -Lemon
560 X(I)=3
570 GOTO 760
580 IF X(I)=5 THEN 620
590 DISP CHR$(9);                      -Plum
600 X(I)=4
610 GOTO 760
620 DISP CHR$(7);                      -Bell
630 X(I)=5
640 GOTO 760
650 IF X(I)>7 THEN 730                 -Weighting process for a bar.

660 IF X(I)=7 THEN 710                  Chance of 3 bars is 1/2560
670 IF I=1 AND RND(.8 THEN 510
680 DISP '=';
690 X(I)=6
700 GOTO 760
710 IF I=2 THEN 480
720 GOTO 550
730 IF I=1 THEN 620
740 IF I=3 AND X(1)<6 THEN 680
750 GOTO 590
760 J=J+3
770 NEXT I
780 DISP TAB(12);
790 P=0                                -Set up value of payoff
800 IF X(1)#1 THEN 820                 -Check for first symbol being a

810 P=2                                 cherry
820 IF X(1)=X(2) THEN 880
830 IF P>0 THEN 1070
840 S=S-B
850 BEEP 220,.2 @ DISP USING 860 ; 'LOS
    E',S
860 IMAGE k,6x,'  $',mdddd.dd
870 GOTO 1110
880 IF X(2)=X(3) THEN 930
890 IF X(1)=1 THEN 910
900 IF X(3)<6 THEN P=3 @ GOTO 1070
910 P=5
920 GOTO 1070
930 IF X(1)=1 THEN P=8 @ GOTO 1070
940 IF X(1)>2 THEN 970
950 P=10
960 GOTO 1070
970 IF X(1)>3 THEN 1000
980 P=15
990 GOTO 1070
1000 IF X(1)>4 THEN 1030
```

PROGRAM LISTING

```
1010 P=20
1020 GOTO 1070
1030 IF X(1)>S THEN 1060
1040 P=50
1050 GOTO 1070
1060 P=100
1070 W=P*B
1080 S=S+W-B
1090 BEEP 660,.2 @ DISP USING 1100 ; 'WI
     N ',W,S
1100 IMAGE K,ddD.DD,'  $',MDdDD.DD
1110 DELAY 2
1120 ON ERROR GOTO 1140
1130 DISP TAB(10);
1140 DISP ' ';CHR$(208);'ull or ';CHR$(2
     11);
1150 INPUT 'top? ','P';Q$
1160 Q$=UPRC$(Q$[1,1])
1170 IF Q$='P' THEN 250
1180 IF Q$='S' THEN 1190 ELSE 1110
1190 ON ERROR GOTO 1200
1200 DISP TAB(6);
1210 INPUT 'Try another machine?','Y';Q$
1220 Q$=UPRC$(Q$[1,1])
1230 IF Q$='Y' THEN 120
1240 IF Q$='N' THEN 1250 ELSE 1190
1250 DELAY 3
1260 DISP TAB(7);
1270 IF S<0 THEN DISP USING 'K,DDD.DD' ;
      'Total losses =',ABS(S)
1280 IF S=0 THEN DISP 'You broke even.'
1290 IF S>0 THEN DISP USING 'K,DDD.DD' ;
      'Total winnings =',S
1300 DISP TAB(11);
1310 DISP 'END OF GAME'
1320 DISP
1330 STOP
```

PROGRAM DESCRIPTION

BREAK-OUT

In this game the player uses a paddle and ball to knock out as many bricks as possible. The playing field is displayed on a video screen using the HP 82163A Video Interface. The paddle is controlled by using the "up" and "down" arrow keys. There are six layers of bricks. Bricks that are farther back are worth more points. There are five balls for each game. Bonuses are given for any leftover balls when all the bricks are gone.

There are two versions of the game, "Break-out" and "Break-thru". The difference between them is that in Break-out, the ball bounces back when it knocks out a brick, and in Break-thru the ball continues on its original path. In addition, there is an auto-play mode available, which will show the game running without user interface.

To play the game, assign the printer to the HP 82163A Video Interface and set the display device to the LCD. Run the program. The first prompt will be for the version of the game you wish to play: Break-out or Break-thru. Press "O" for Break-out, or "T" for Break-thru. Then press "A" for Auto-play mode, or "N" for regular play. Press any key to serve the ball, and use the "up" and "down" arrows to move the paddle. If you miss with the first ball and require a second, the serve will occur after you press any key.

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
T(,)	Target array: 0 if hit, 1 if not	M	Mode flag
X	Next X coordinate	XO	Current X coordinate
Y	Next Y coordinate	YO	Current Y coordinate
I,J	Temporary counters	B	Current ball
S	Score	H	High score
U	Horizontal vector	V	Vertical vector
P	Paddle position	T	Count of targets left
A	Auto mode	K$	Current key hit
R	Ricochet	E$	Escape
L$	Linefeed	B$	Backspace
S$	Inverse space	C$	Clear, cursor off
A$	Cursor address	AO$	Address column 0
A3$	Address column 3	A5$	Address column 5
U$	Paddle up	D$	Paddle down

NOTES AND REFERENCES

Thanks to Ed Groth for the original program, Jack Applin IV and Raan Young of CVD for their contributions.

PROGRAM LISTING

```
 10 ! Breakout- video game
 20 ! Requires 82163A
 30 !
 40 ! Revision 11/01/82
 50 !
 60 OPTION BASE 1
 70 DIM E$[1],L$[1],B$[1],S$[1],C$[4],A
    $[2],A0$[3],A3$[3],A5$[3],K$[1],U$[
    6],D$[6]
 80 INTEGER T(6,13),X,Y,X0,Y0,I,J,S,B,M
    ,U,V,P,T,H
 90 E$=CHR$(27) @ L$=CHR$(10) @ B$=CHR$
    (8) @ S$=CHR$(160)
100 C$=E$&'E'&E$&'('                    -Clear, cursor off
110 A$=E$&'%' @ A0$=A$&CHR$(0) @ A3$=A$
    &CHR$(3) @ A5$=A$&CHR$(5)
120 U$=S$&L$&L$&L$&B$&' '
130 D$=' '&L$&L$&L$&B$&S$
140 H=0
150 RANDOMIZE
160 PRINTER IS ":TV" @ PWIDTH INF @ PRI
    NT C$
170 PRINT 'Do you wish to play Break-Ou
    t   or Break-Thru?   Break-';
180 K$=UPRC$(KEY$)
190 IF K$="O" THEN M=-1 @ PRINT "Out" @
     GOTO 220
200 IF K$="T" THEN M=1 @ PRINT "Thru" @
     GOTO 220
210 GOTO 180
220 PRINT
230 PRINT "Auto or Normal mode? ";
240 K$=UPRC$(KEY$)
250 IF K$="A" THEN A=1 @ PRINT "Auto" @
     GOTO 300
260 IF K$="N" THEN A=0 @ PRINT "Normal"
     @ GOTO 280
270 GOTO 240
280 PRINT
290 PRINT 'Use up/down arrows to move
        paddle; any key to serve ball'
300 ! Construct playing board
310 WAIT 5 @ PRINT C$
320 ! Build frame
330 FOR I=0 TO 31 @ PRINT A$;CHR$(I);CH
    R$(0);S$; @ NEXT I
340 FOR I=1 TO 13 @ PRINT A$;CHR$(31);C
    HR$(I);S$; @ NEXT I
350 FOR I=31 TO 0 STEP -1 @ PRINT A$;CH
    R$(I);CHR$(14);S$; @ NEXT I
360 ! Build targets
370 FOR J=1 TO 6 @ FOR I=1 TO 13
380 T(J,I)=1
390 PRINT A$;CHR$(2*J+16);CHR$(I*(1-MOD
    (J,2))+(13-MOD(J,2)*(I-1))*MOD(J,2)
    );S$;
```

PROGRAM LISTING

```
400 NEXT I @ NEXT J
410 ! Initialize ball, score, paddle, t
    arget count
420 B,S=0 @ P=7 @ T=78
430 ! Serve ball: build paddle, get bal
    l & coordinates, show score, wait f
    or key
440 PRINT A0$;CHR$(P-1);S$;L$;B$;S$;L$;
    B$;S$;
450 R=0 @ B=B+1 @ IF B>5 THEN 960
460 IF A THEN GOTO 480
470 IF KEY$='' THEN GOTO 470
480 GOSUB 1150
490 X0=14 @ Y0=INT(26*RND)+2 @ U=-1 @ V
    =INT(5*RND)-2
500 ! Find next point; let paddle move

510 X=X0+U @ Y=Y0-V
520 GOSUB 1100
530 ! Did ball hit walls?
540 IF Y<2 OR Y>27 THEN V=-V @ GOTO 780
550 ! Did ball hit end?
560 IF X>30 THEN U=-U @ R=1 @ GOTO 780
570 ! Did ball miss paddle?
580 IF X<0 THEN GOTO 870
590 ! Did ball hit paddle?
600 IF X=0 AND (Y0\2)=P-1 AND Y0\2<=P+1
     OR Y\2)=P-1 AND Y\2<=P+1) THEN GOT
    O 810
610 ! Ball in target range?
620 IF X>=18 AND X<=28 AND MOD(X,2)=0 T
    HEN GOTO 670
630 ! Move ball, go find next point
640 GOSUB 1050
650 GOTO 510
660 ! Did ball hit target?
670 I=(X-18)\2+1 @ J=Y\2
680 IF NOT T(I,J) THEN GOTO 640
690 ! Process target hit
700 BEEP X*40+500,.05 @ T(I,J)=0 @ S=S+
    (I+3)*2 @ T=T-1
710 U=U*M
720 IF NOT R AND M<0 THEN U=-ABS(U)
730 GOSUB 1050
740 GOSUB 1150
750 IF T=0 THEN 910
760 GOTO 510
770 ! Process wall or end hit
780 X=X0 @ Y=Y0 @ BEEP 400,.05
790 GOTO 640
800 ! Process paddle hit
810 IF V=0 THEN V=V+INT(3*RND)-1
820 U=-U @ Y=Y0 @ X=X0 @ BEEP 600,.05
830 IF Y0\2<=P-1 AND V<0 OR Y0\2>=P+1 A
    ND V>0 THEN V=-2*SGN(V)
```

PROGRAM LISTING

```
 840 V=V+INT(3*RND)-1 @ IF ABS(V)>2 THEN
      V=2*SGN(V)
 850 GOTO 640
 860 ! Process paddle miss
 870 BEEP 20,.5
 880 PRINT A0$;CHR$(Y0\2);' ';
 890 GOTO 440
 900 ! Cleared board: show bonus, score,
      restart
 910 PRINT A$;CHR$(X0);CHR$(Y0\2);' ';
 920 PRINT A3$;CHR$(3);'Bonus for';6-B;'
     balls left:';150*(6-B);
 930 S=S+150*(6-B)
 940 GOSUB 1150
 950 ! Out of balls if entered here
 960 IF S>H THEN H=S
 970 PRINT A3$;CHR$(5);'High:';H;
 980 PRINT A3$;CHR$(7);'Again? ';
 990 K$=UPRC$(KEY$) @ IF A THEN K$="Y"
1000 IF K$='Y' THEN PRINT 'Yes' @ GOTO 3
     10
1010 IF K$#'N' THEN GOTO 990
1020 PRINT C$
1030 END
1040 ! Move ball subroutine
1050 PRINT A$;CHR$(X0);CHR$(Y0\2);' ';
1060 PRINT A$;CHR$(X);CHR$(Y\2);'*';
1070 X0=X @ Y0=Y
1080 RETURN
1090 ! Move paddle subroutine
1100 K$=UPRC$(KEY$)
1110 IF (K$='' OR A AND Y0\2<P) AND P>2
      THEN P=P-1 @ PRINT A0$;CHR$(P-1);U
     $;
1120 IF (K$='' OR A AND Y0\2>P) AND P<1
     2 THEN PRINT A0$;CHR$(P-1);D$; @ P=
     P+1
1130 RETURN
1140 ! Show score subroutine
1150 PRINT A5$;CHR$(15);'Ball:';B;'   Sc
     ore:';S;
1160 RETURN
```

Sent Aug. 83
by Michael Markov

TABLE OF CONTENTS

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 determine 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
```

PROGRAM LISTING

```
370 DISP 'and kills you !!' @ GOTO 2100
380 IF V5>10 THEN 400                          -Find appropriate routine for
                                                verb (command)
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                        -Check for valid direction
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         -Routine for 'go' command
    "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    -Attack monster
    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$[1,9] THEN DISP "It is
    n't here." @ GOTO 100
610 IF C5=12 THEN 950
620 IF C5#5 AND C5#9 AND C5#16 AND C5#2    -Check for rooms with monsters
    0 THEN 750
630 IF RND>.3 THEN K=1                     -Probabilities for battle
640 IF RND>.85 THEN M=1
650 IF A5=1 THEN DISP 'again, ';
660 DISP A1$;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$
```

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:"
     @ Z=0
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$)=0 THEN DISP V$;'
      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 '
     Huh?' @ GOTO 100
```

-Check to see what player is carrying

-Pick up something

-Unlock or open something

PROGRAM LISTING

```
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
     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
      "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$[4,5]#"12" OR E=1 THEN DISP "N
     othing happened." @ GOTO 100
1570 OFF ERROR
     READ # 1,59 ; O$[1,15]
```

-Leave something

-Change clock

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                                  -Say something
1960 IF C5=3 AND N=9 AND D1=0 THEN DISP
     "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          -Throw something
1990 IF N#26 THEN DISP "I don't understa
     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                      -Study something
2070 IF N=37 AND B(N,0)=-1 THEN DISP "It
      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 *                 -End of game
2160 DISP TAB(9);"END OF GAME" @ DISP @
     STOP                                      -Initialize game
2170 DISP TAB(11);'ADVENTURE'
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$[1*5-4,I*5] @
      NEXT I
2260 M1$="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: $X = X_0 + V_0 t + \frac{1}{2} a t^2$

$V = V_0 + at$

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

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
		*** ROCKET LANDER ***
1	Select planet	Choose planet (1-9):
	If '1' then	*** MERCURY ***
	If '2' then	*** MARS ***
	If '3' then	*** PLUTO ***
	If '4' then	*** VENUS ***
	If '5' then	*** EARTH ***
	If '6' then	*** URANUS ***
	If '7' then	*** NEPTUNE ***
	If '8' then	*** SATURN ***
	If '9' then	*** JUPITER ***
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
3a	Display current conditions and	
	enter burn:	n n n
3b	Goto step 3a until:	
	1) out of fuel	
	2) crashed, or	
	3) landed safely	
4	If out of fuel, display the	
	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,mddddddd.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
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
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
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
640 T=T+1
650 GOTO 880
660 IF T8>0 AND T8<=T9 THEN 850
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

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
810 X=X+V+A/2
820 V=V+A
830 GOTO 770
840 T=T+T9
```

-Check for valid fuel burn

-Avoid division by zero

-Avoid division by zero

-Test altitude for safe landing

-Test fuel supply

-Test altitude after fuel supply is depleted

-Test fuel supply

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])

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
```

-Check for valid initial velocity

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)

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
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
240 DISP 'You will be using 4 notes...'
     @ WAIT 2 @ DISP
250 J=2

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

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...'
```

-Set up the frequencies for 8 notes

-Introduction - play tune from 'Close Encounters'

-Select game difficulty

-J sets up the tab positions for the bell

-Bell positions are 5 spaces apart

PROGRAM LISTING

```
450 J=1

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

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
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
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
    OFF ERROR
```

-J sets up the tab positions for the bell

-Random number between 1 and the number of notes in play

-Check for valid key input

-Check for correct response to sequence

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 DESCRI

BLACKJACK

In this card game, the HP-75 is the dealer and up to eig
Each player begins with $200 and may bet as much or all
(in $10 increments) up to $200. Before the cards are dea
enters his bet (to leave the game, enter 0 for a bet). Th
deals the cards from a pack of 104 (double deck). If the
a natural, he immediately collects the bets of all players
naturals. (A natural is an ace and any face card or ten,
21 in two cards. A natural is also called Blackjack). If
any other player both have naturals, the bet is a standoff
keeps his money. If the dealer does not have a natural, the
with each player deciding what he will do with his hand. Ac
counted as 1 or 11, as the player wishes. All face cards co
(K, Q and J) and any other cards count as their pip value (n
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 (/),
A hit means he would like another card to add to the ones he a
double (down-for-double) means he wants only one more card and
his original bet. This is usually done if the player has a to
ready in his hand, taking a chance on getting a ten or face car
A split means the player has two identical cards, i.e., two qu
fives, etc., and would like to separate them into two different
ing two more cards to go with the two like cards. The bet that
originally made is then placed also on the second hand. In othe
is now playing for twice as much money. The player must play ea
separately, finishing his 1st hand before going to the 2nd. A s
the player is satisfied with the cards he has.

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

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

VARIABLE NAM

NAME	DESCRIPTION	NAME	
N	Number of players	F2	Usec rou: rout 'bust
H(,)	Players hands and totals		
J,I	Looping, index	P	Playe
K1	Counts cards used in deck	Z	Loop t
D( )	Dealers hand and total	Z1	Random in shu
T1,T2	Index used for manipulating and evaluating hands	F	Flag to players
B( )	Bets	F5	Flag to out of r
T( )	Total cash	F( )	Indicate split hi
N( )	Flags for naturals	B	Total ga split ha
C	Number of cards in a hand	A1	Indicates
A	Value of cards in a hand	D$	Shuffled
F1	Indicates split - used to convert parts of programs to subroutines	T$	Cards used
D1$	Used to shuffle deck - temporary storage	Q$	User inter options on

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,
    11,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$&T$
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(I)=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
```

PROGRAM LISTING

```
470 GOTO 580
480 FOR I=1 TO N
490 IF T(I)=0 THEN 570

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)
    @ T(I)=0 @ GOTO 570
550 IF B(I)<10 OR B(I)>200 OR B(I)/10#I
    P(B(I)/10) THEN 500
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

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

720 IF K1=104 THEN GOSUB 2670
730 K1=K1+1
740 H(I,1)=NUM(D$[K1,K1])
750 IF K1=104 THEN GOSUB 2670

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
890 IF D(T2)=64 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.
```

-Check to see if player is out of money

-Quit if bet is zero

-Check for valid bet between $10 and $200

-Check for all players wanting to quit

-Check to see if player is out of money

-Deal card
-Check to see if we need to reshuffle the deck

-Check for ace for blackjack

PROGRAM LISTING

```
930 IF T(I)=0 THEN 990                          -Check to see if a player is
                                                 out of money
940 T1=1 @ T2=2                                 -Check for ace for blackjack
950 IF H(I,T1)#65 THEN 980
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$)+1 GOTO 1240,1260,        -Options- Stand, Hit, Double or
     1420,1530,1660                              Split(/)
1260 ! STAND.
1270 H(I,12)=0
1280 A1=0
1290 FOR J=1 TO C                               -Check for regular numeric card
1300 IF H(I,J)>=50 AND H(I,J)<=57 THEN A
     =VAL(CHR$(H(I,J))) @ GOTO 1340             -Check for face card
1310 IF H(I,J)#65 THEN A=10 @ GOTO 1340         -Check for previous aces
1320 IF A1=1 THEN A=1 @ GOTO 1340
```

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])
```

```
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';'           -Player ended up with even
      $';T(I) @ GOTO 2590                   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);'    -Players with less than $10
       $';T(I)                              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   -Check to see if a player is
     s out.'                                out of money
2590 NEXT I
```

```
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 letters 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 continue on to the next word.

After the player has tried five words, he will be asked if he wants to continue 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$(205);'e
    dium, or ';CHR$(200);
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
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
    ,250,260,270,280
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
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
```

-Easy=5 letters, Medium=6, Hard=7

-Time limits are 15,20 and 25 seconds

-Move data pointer to random data stmt for each word set

-There are 10 words per data statement

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

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
```

-Check to see if word is scrambled

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
```