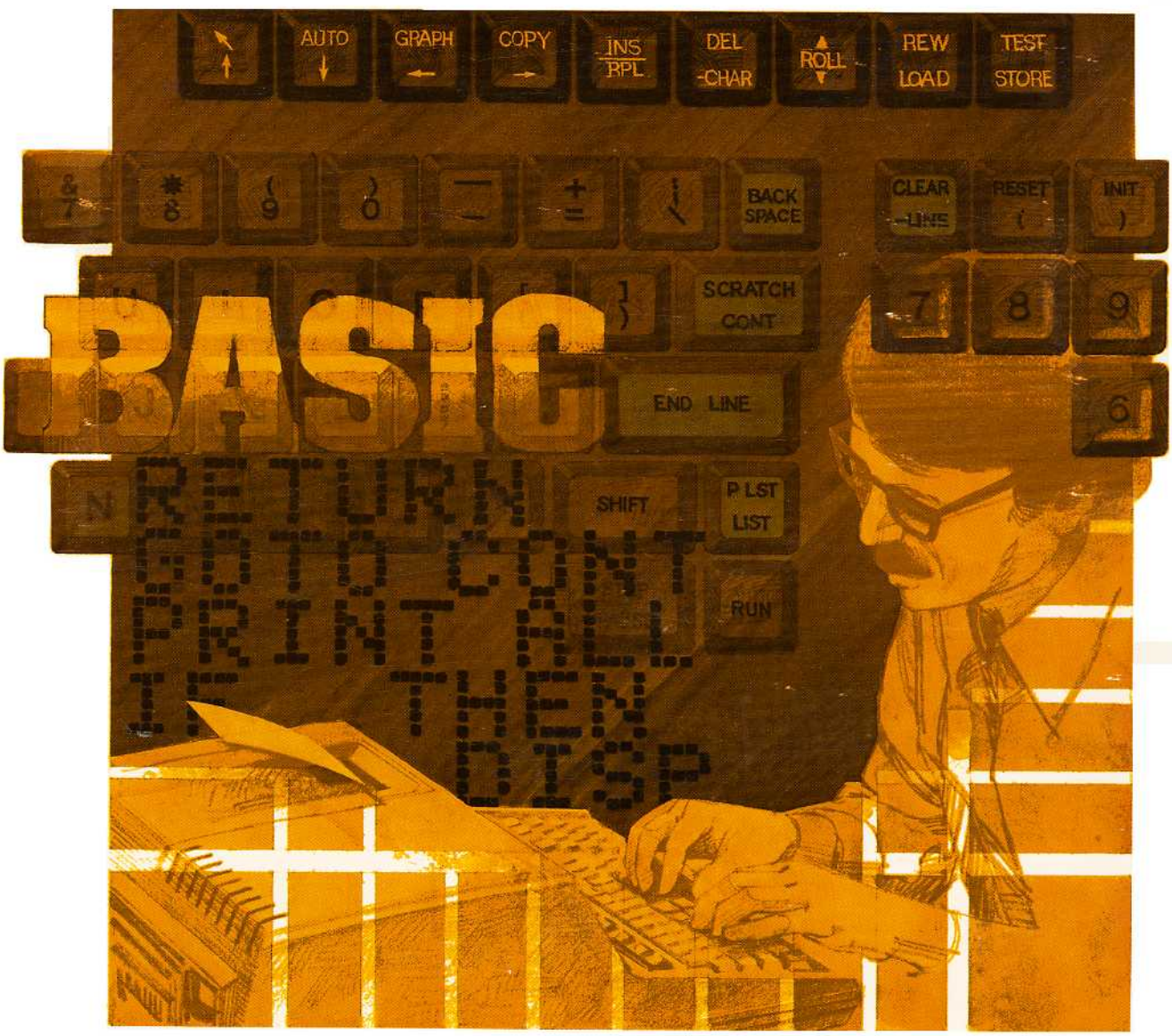
HEWLETT-PACKARD

BASIC Training Pac Supplement

HP-86/87



HP-86/87

BASIC Training Pac Supplement

September 1982

00087-90195

Printed in U.S.A.

Contents

HELP Section

1. Reference: "CH3" printout, steps 3-11.

Note: The symbol ▲ means press the space bar to produce one space.

To type and run the "Hello" program, press these keys in the order shown.

a. To clear memory, switch the HP-86/87 off, then on.

b. Remember, to get ["], hold down [SHIFT] and press the ["] key.

c. Press [1] [0] ▲ [D] [I] [S] [P] ▲ ["] [H] [E] [L] [L] [O] ["].

d. Press [END LINE].

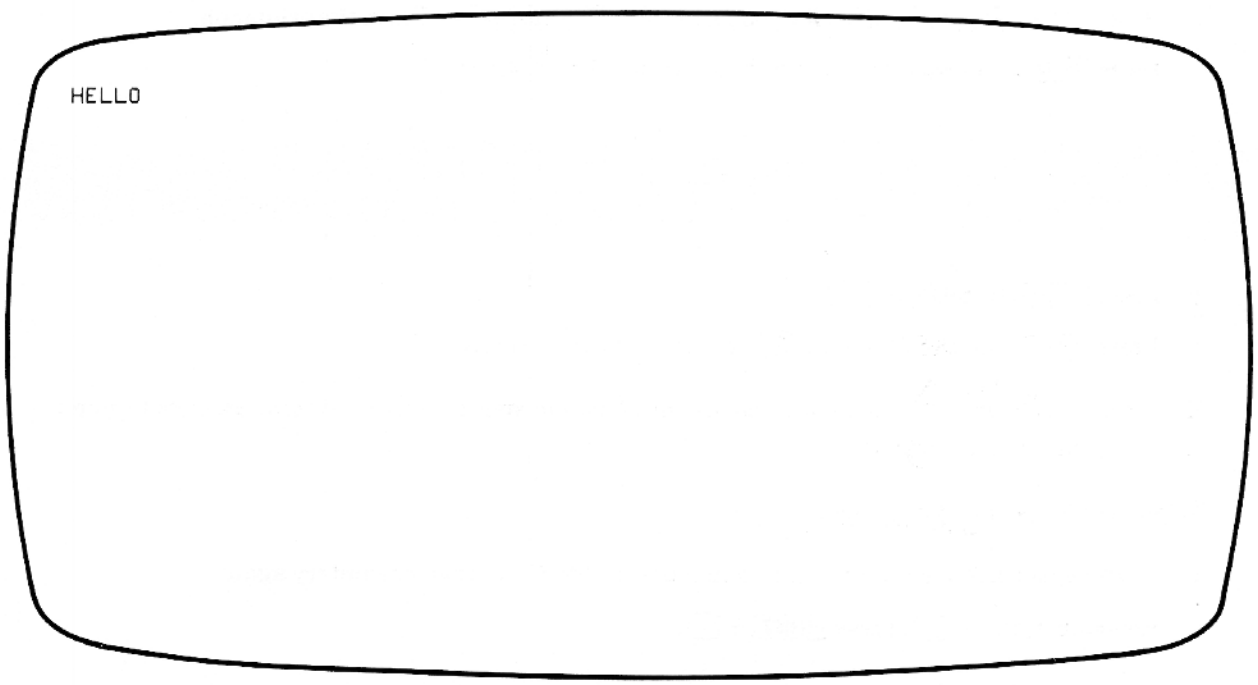
e. Press [2] [0] ▲ [E] [N] [D].

f. Press [END LINE].

g. To clear your screen, press [CLEAR]; that is, hold down [SHIFT] and press the [CLEAR LINE] key.

h. Press [RUN].

You should now see:



on your screen. If not, start at step 1 on your printout and try again.

2. Reference: "CH3" printout, steps 12 and 13.

   To clear your screen and list your program on the screen:

   a. Press [CLEAR] (that is, hold down [SHIFT] and press the [CLEAR LINE] key).

   b. Press [LIST] (that is, press the [PLST LIST] key).

   If your screen shows other differences from the listing shown in step 13 of your printout, go back to step 1 on your printout and start over.

3. Reference: "CH3" printout, steps 14-21.

Note: The symbol ▲ means press the space bar to produce one space.

To instruct the HP-86/87 to say HELLO from the printer, and then to run and list your revised program, follow these steps:

a. Press [CLEAR] (that is, hold down [SHIFT] and press the [CLEAR LINE] key).

b. Remember, to get ["], press [SHIFT] + ["].

c. Press [1] [0] ▲ [P] [R] [I] [N] [T] ▲ ["] [H] [E] [L] [L] [O] ["].

d. Press [END LINE].

e. Press [P] [R] [I] [N] [T] [E] [R] ▲ [I] [S] ▲ [7] [0] [1] [END LINE] and turn the printer on.

f. Press [RUN] and watch the printer. Your printer should show:

```
HELLO
```

g. Press [CLEAR] ([SHIFT] + [CLEAR LINE]).

h. Press [LIST] (that is, press the [PLST LIST] key) and watch the screen.

If your screen shows differences from the listing shown in step 20 of your printout, go back to step 1 on your printout and start over.

4. Reference: "CH4" printout, steps 3-10.

   a. If the keystrokes below don't help, switch the HP-86/87 off, then on, and try again.

   b. Remember, to get ["], press [SHIFT] + ["].

Note: The symbol ▲ means press the space bar to produce one space.

Steps 3 and 4:

Press [2][0]▲[D][I][S][P]▲["][M][Y][N][A][M][E][I][S][H][P][-][8][6]/[8][7][.]["] [END LINE]

If you get an error, clear your screen ([SHIFT] + [CLEAR -LINE]) and try again.

Steps 5 and 6:

Press [3][0]▲▲▲▲▲▲▲▲▲▲[E][N][D].

(generate 10 spaces with the space bar)

Press [END LINE].

Steps 7-9:

Press [LIST] (press the [PLST LIST] key). Wait for listing to display on screen. Then press [RUN].

5. Reference: "CH4" printout, steps 11-18.

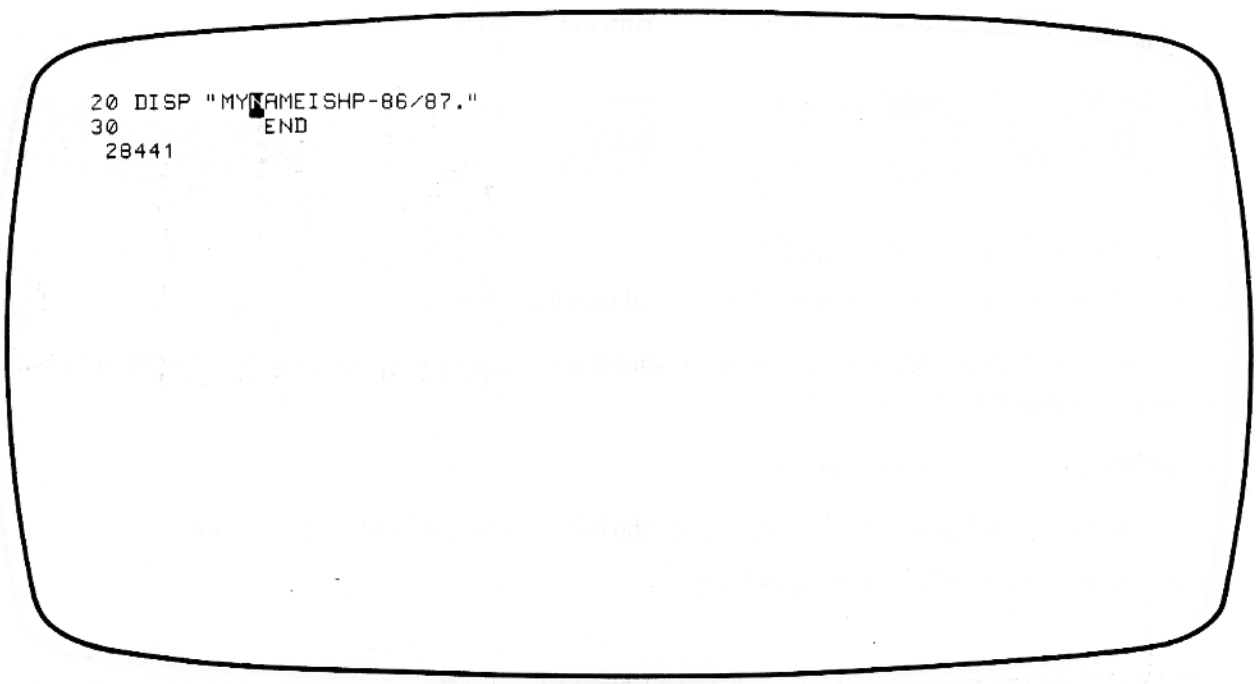
If the keystrokes below don't help, switch the HP-86/87 off, then on and start at step 3 on your printout.

Step 11: To clear your screen, press [CLEAR] ([SHIFT] + [CLEAR -LINE]).

Step 12: Press [LIST] (press the [PLST LIST] key).

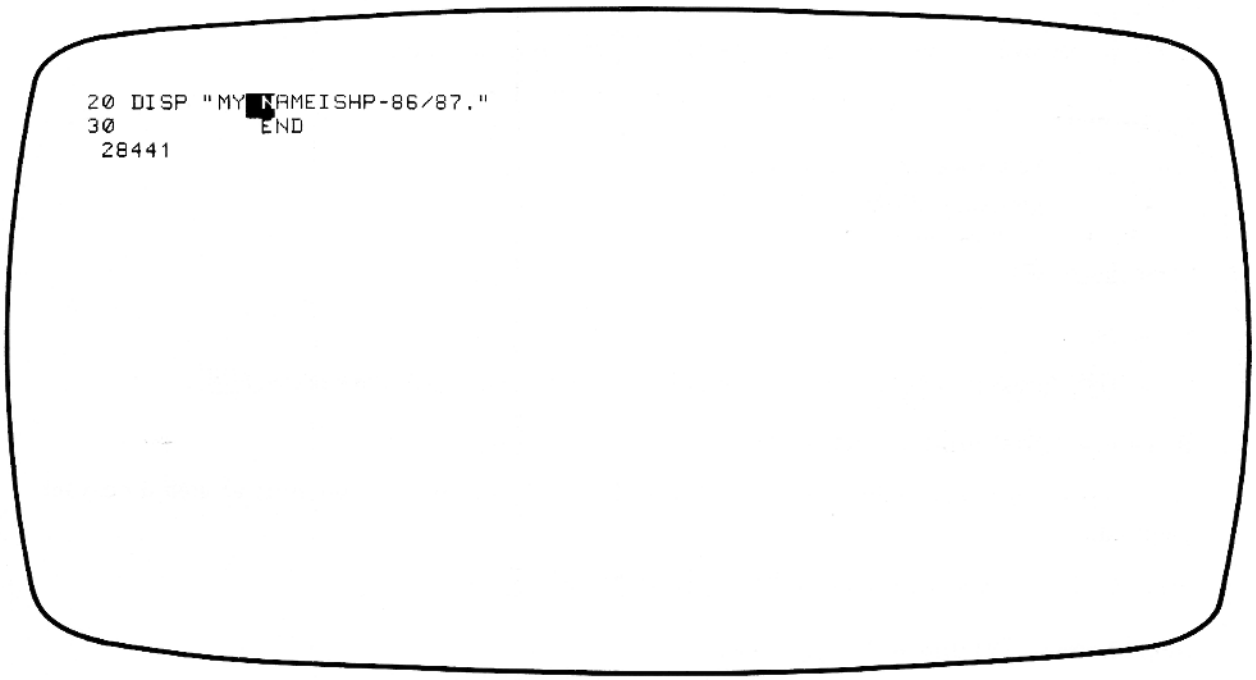
Steps 13-14:

Press [SHIFT] + [↑].
Press [→] 11 times. You should see:



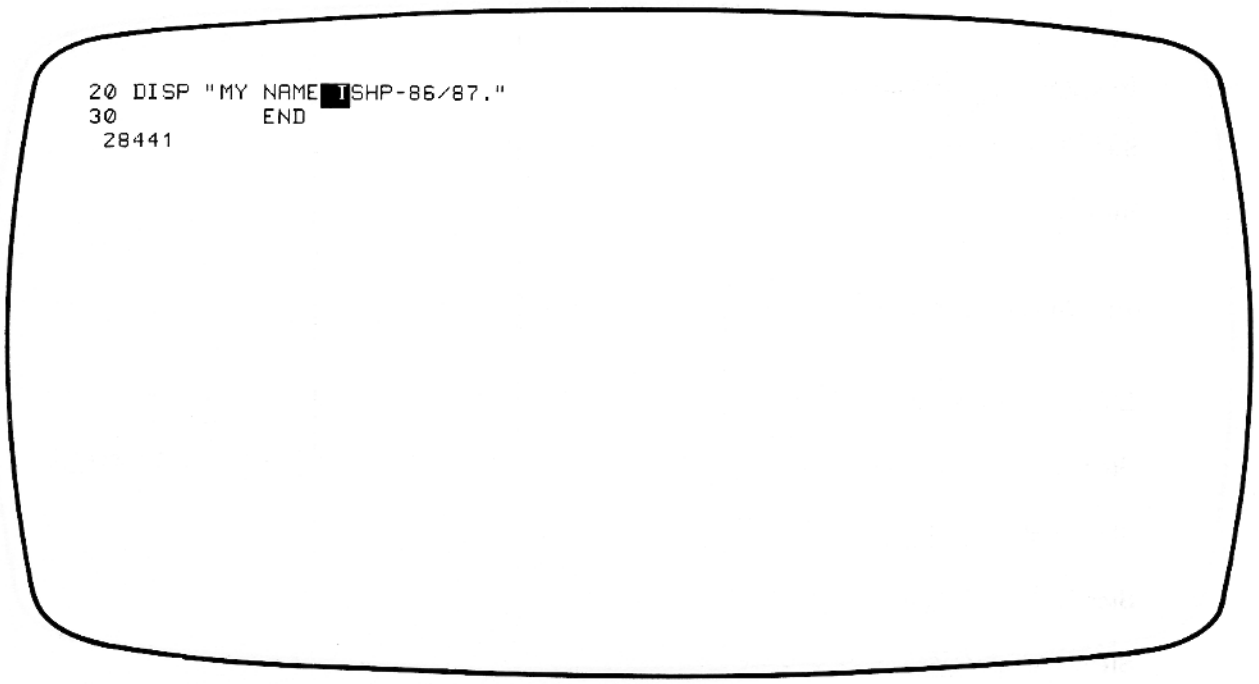
Press (SHIFT) and (I/R ←)

Press the space bar. Now you should see:

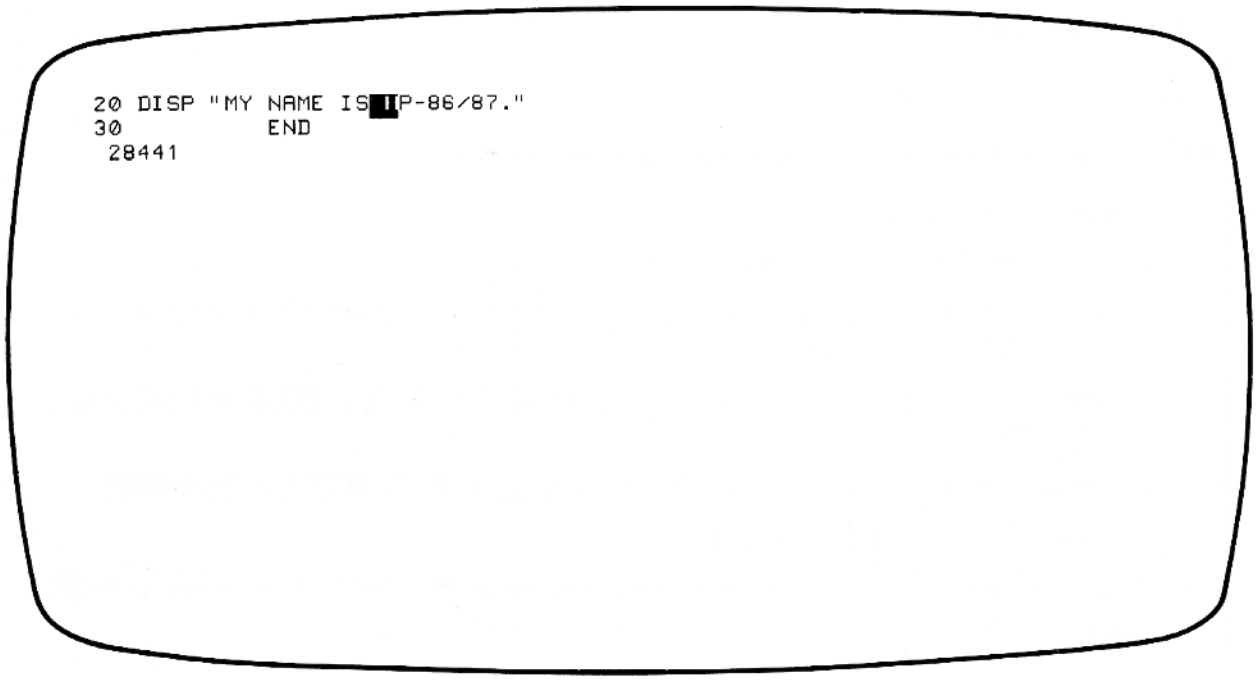


Press (→) 4 times.

Press the space bar. Now you should see:



Press → two times.
Press the space bar. Now you should see the result shown on your printout under step 14, which is:



Now press END LINE and the editing of line 20 is complete.

Steps 15-18: Press, in order: END LINE SHIFT + CLEAR LINE RUN LIST.

6. Reference: "CH4" printout, steps 19-23

Note: The symbol ▲ means press the space bar to produce one space.

If the keystrokes below don't help, switch the HP-86/87 off, then on, and start at step 3.

Steps 19 and 20: Press 1 0 ▲ D I S P ▲ " G R E E T I N G S " END LINE.

Step 21: Press CLEAR (SHIFT + CLEAR LINE).

Step 22: Press RUN.

Step 23: Press LIST.

7. Reference: "CH4" printout, steps 26-29

   Step 26: Press [CLEAR] ([SHIFT] + [CLEAR LINE]).

   Step 27: Press [2] [0] [END LINE].

   Step 28: Press [RUN].

   Step 29: Press [LIST].

8. Reference: Pages 4-3 and 4-4, steps 19-24.

   Note: The symbol ▲ means press the space bar to produce one space.

   a. To write your first program:

      1. Turn the HP-86/87 off, then on.
      2. Press [1] [0] ▲ [P] [R] [I] [N] [T] ▲ ["] [H] [P] [-] [8] [6] [/] [8] [7] ▲ [I] [S] ▲ [M] [Y] ▲ [N] [A] [M] [E] ["] [END LINE].
      3. Press [2] [0] ▲ [P] [R] [I] [N] [T] ▲ ["] [A] [N] [D] ▲ [S] [E] [R] [V] [I] [N] [G] ▲ [Y] [O] [U] ["] [END LINE].
      4. Press [3] [0] ▲ [P] [R] [I] [N] [T] ▲ ["] [I] [S] ▲ [M] [Y] ▲ [G] [A] [M] [E] [.] ["] [END LINE].
      5. Press [4] [0] ▲ [E] [N] [D] [END LINE].

   b. To list your program, whether it's finished or not, press [LIST] to list on the screen, or [PLST] ([SHIFT] + [PLST LIST]) to list on the printer. To run your program, press [RUN].

9. Reference: Page 4-4, step 25.
   Listing for "Name-Game" program.

```
10 PRINT "HP-86/87 IS MY NAME."
20 PRINT "AND SERVING YOU"
30 PRINT "IS MY GAME."
40 END
 28394
```

10. Reference: Page 5-7, step 24.

Here's the way to edit line 100 of the "Iron Jaw" story:

a. Use the display editing keys to position the cursor under the O in WASON (line 100).

b. Press (SHIFT) + (I/R) to get the insert mode. You should see the cursor over the S and the O.

c. Press the space bar. The word WAS should now appear, followed by a space, then ON. Notice what happened to N, at the end of the line. They were the last two characters on the first line of statement 100. Now the comma (,) has been moved down to become the first character of the next line. N was character 79, but when you inserted the cursor and a space at character position 15, all higher numbered characters advanced one position. Character 79 (N) became character 80, and character 80 (,) became character 81.

d. Press (SHIFT) + (I/R) to go back to the replace mode. The O of ON is visible beneath the cursor.

e. Now use the (→) key to move the cursor right two positions to the space between N and L.

f. Press (SHIFT) + (-CHAR). Zap goes the space, leaving ONYON . Also, the comma (,) has moved back to be the last character on the first line again.

g. Move the cursor two more positions to the right (to the O).

h. Press (SHIFT) + (I/R) once again to switch into insert mode. Now the cursor should be over the Y and the O.

i. Press the space bar. Now English begins to emerge from gibberish.

j. Move the cursor two positions to the right. Now the cursor is over the N and the space in front of K.

k. Press (E) to add on E to ON.

l. Move your cursor one more position to the right.

m. Press (SHIFT) + (I/R) to get the replace mode. Your cursor should now be over K.

n. Press (W).

o. Move right two positions.

p. Press (K).

q. Move right three positions.

r. Press (A).

s. Move the cursor right until it is over the U of STUR.

t. Press (A). Don't celebrate yet. You have not finished correcting line 100.

u. Press (END LINE). Now you've finished editing line 100. Good luck with the remaining three lines.

11. Reference: "CH5," first printout, steps 1-6

Step 1: Press [k12] [END LINE], (or type SCRATCH and press [END LINE]), then press [RESET], ([SHIFT] + [RESET]).

Step 3: Here are the keystrokes for statement 21:

Note: The symbol ▲ means press the space bar to produce one space.

a. Make sure your cursor is at the left hand edge of the screen, on the line just below statement 14.

b. Press [2] [1] ▲ [D] [I] [S] [P] ▲ ["] ['] [H] [E] [L] [L] [O] ['] ▲ [A] [N] [D] ["]

c. If there are any characters on this line to the right of the final ", press [-LINE] (that is, press the [CLEAR -LINE] key).

d. Press [END LINE].

Step 4: After you have entered the four statements, you run your program by simply pressing the single [RUN] key.

Step 6:

To load program "CH5" and start at line 1000.

a. Insert your BASIC Training disc into the disc drive.

b. Press [k6] ["] [C] [H] [5] ["] [END LINE]. Remember, ["] means [SHIFT] + [']. If you get an error message, clear you screen ([SHIFT] + [CLEAR -LINE]) and try again. And check your keystrokes.

c. Wait for the disc drive light to go out.

d. Press [R] [U] [N] ▲ [1] [0] [0] [0] [END LINE].

12. Reference: "CH5," second printout, steps 1-4

Step 1: Press [k12] [END LINE], (or type SCRATCH and press [END LINE]), then press [RESET], ([SHIFT] + [RESET]).

Steps 2, 3 and 4:

It's easy to get mixed up when entering statements. Let's follow the misadventures of "Ten Thumbs" McBotch as he tries to enter "MONEY."

Here is his attempt to type in statement 14. Before he presses [END LINE], his screen looks like this:

```
14 PRENTER IS 701
```

Ten Thumbs doesn't focus on the misspelling, so he presses [END LINE], and sees:

```
14 PRENTER IS 701
Error 92 : SYNTAX
```

Now he notices the bad spelling. He presses ← six times, then I, and sees:

```
14 PRINTER IS 701
Error 92 : SYNTAX
```

He presses END LINE and sees:

```
14 PRINTER IS 701
Error 92 : SYNTAX
```

Notice how the HP-86/87 accepts his revised statement (since no beep is heard), even though his cursor was in the middle of line 14 when he pressed END LINE. You may enter a statement into the HP-86/87's memory (that is, you may type a statement and press END LINE) as long as your cursor is under any of the 80 positions of the screen line containing your statement. For instance, if your screen looked like this (note the cursor location):

```
14 PRINTER IS 701
```

and you pressed END LINE, statement 14 would be correctly entered into the HP-86/87's memory.

Now McBotch tries for line 21. He doesn't clear the line before he types in his line 21 effort. Here is how his screen looks after he finishes his typing:

```
14 PRINTER IS 701
21 FOR S=1 TO 40X
```

"No typing mistakes," he proudly observes, and, with a flourish, he presses END LINE. Another beep! His screen shows:

```
14 PRINTER IS 701
21 FOR S=1 TO 40X
Error 84 : EXCESS CHARS
```

"Easily fixed," he says, and presses SHIFT + CHAR → to delete the X, then presses END LINE. Since the HP-86/87 does not beep, Ten Thumbs knows his statement is accepted. Now his screen shows:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
Error 84 : EXCESS CHARS
```

This time Ten Thumbs gets smart. First he presses (-LINE) to clear the line and gets:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
```

Looking at statment 22, he sees the symbol ^. McBotch feels in complete control, since he knows ^ is an exponent symbol, and that the expression 2 ^ (S − 1) means raise 2 to the (S − 1) power. He also knows the ^ symbol is typed by pressing the (RESLT) key or the (SHIFT) + (^ 6) keys. So he types in statement 22 and presses (END LINE). He sees:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
```

"My first error free statement!" he chortles, and types in statement 29, trying for two in a rwo. After pressing (END LINE), he is happy to hear no beep. His screen shows:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE",S,"HAS $,D"
```

"What's hard about this?" Ten Thumbs remarks, and quickly enters the last two statements. He then gazes proudly on his result:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE",S,"HAS $,D"
40 NEXT S
97 END
```

"Now I'll run this little beauty," he confidently forecasts, and presses (RUN). He watches as his printer reveals:

```
SQUARE          1          HAS $,D
SQUARE          2          HAS $,D
SQUARE          3          HAS $,D
SQUARE          4          HAS $,D
SQUARE          5          HAS $,D
SQUARE          6          HAS $,D
  :             :            :
SQUARE          35         HAS $,D
SQUARE          36         HAS $,D
SQUARE          37         HAS $,D
SQUARE          38         HAS $,D
SQUARE          39         HAS $,D
SQUARE          40         HAS $,D
```

He compares his printout with that shown on page H-18. "Something is wrong," muses McBotch, "but what?" He inspects his displayed listing, concentrating on that tricky line 29. "Of course," he shrieks, "I used commas instead of the semicolons I was supposed to use!" So he uses his cursor moving arrows to put his cursor over the first comma in line 29:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE",S,"HAS $,D"
40 NEXT S
97 END
```

After pressing (;), he sees:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE";S,"HAS $,D"
40 NEXT S
97 END
```

He changes the remaining two commas into semicolons in the same way:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE";S;"HAS $;D"
40 NEXT S
97 END
```

and presses (END LINE):

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE";S;"HAS $;D"
40 NEXT S
97 END
```

"Now I've got it," he exults, and presses (RUN) again. Now his printer shows:

```
SQUARE 1 HAS $;D
SQUARE 2 HAS $;D
SQUARE 3 HAS $;D
SQUARE 4 HAS $;D
SQUARE 5 HAS $;D
SQUARE 6 HAS $;D
SQUARE 7 HAS $;D
SQUARE 8 HAS $;D
SQUARE 9 HAS $;D
SQUARE 10 HAS $;D
SQUARE 11 HAS $;D
        :
SQUARE 38 HAS $;D
SQUARE 39 HAS $;D
SQUARE 40 HAS $;D
```

(Expletive deleted), he curses, and takes another look at line 29. This time he checks each character, one by one, against the printed listing given by program "CH5." "Aha! The problem lies in the last three characters!" he correctly concludes. Instead of

```
;D"
```

McBotch should have typed

```
";D
```

So he moves his cursor to these last three characters and makes the changes. Now he sees:

```
14 PRINTER IS 701
21 FOR S=1 TO 40
22 LET D=2^(S-10)
29 PRINT "SQUARE";S;"HAS $";D
40 NEXT S
97 END
```

With confidence born of inexperience, he presses (RUN) and sees:

```
SQUARE 1 HAS $;D
SQUARE 2 HAS $;D
SQUARE 3 HAS $;D
SQUARE 4 HAS $;D
SQUARE 5 HAS $;D
SQUARE 6 HAS $;D
SQUARE 7 HAS $;D
SQUARE 8 HAS $;D
SQUARE 9 HAS $;D
SQUARE 10 HAS $;D
SQUARE 11 HAS $;D
        :
SQUARE 38 HAS $;D
SQUARE 39 HAS $;D
SQUARE 40 HAS $;D
```

Now Ten Thumbs, with an impressive leap of insight, realizes he forgot to press (END LINE) after he typed in his corrections to line 29. His cursor is still positioned where he left it, one position past the end of his corrected line 29, so he need not move his cursor. He presses (END LINE), then (RUN), and watches another disaster emerge:

```
SQUARE 1 HAS $ .001953125
SQUARE 2 HAS $ .00390625
SQUARE 3 HAS $ .0078125
SQUARE 4 HAS $ .015625
SQUARE 5 HAS $ .03125
SQUARE 6 HAS $ .0625
SQUARE 7 HAS $ .125
SQUARE 8 HAS $ .25
SQUARE 9 HAS $ .5
SQUARE 10 HAS $ 1
SQUARE 11 HAS $ 2
        :
SQUARE 36 HAS $ 67108864
SQUARE 37 HAS $ 134217728
SQUARE 38 HAS $ 268435456
SQUARE 39 HAS $ 536870912
SQUARE 40 HAS $ 1073741824
```

Ten Thumbs is beginning to get the message: A computer is very fast, but very simpleminded, and an extremely poor mind reader.

McBotch puts on a new pair of glasses and performs a careful character-by-character comparison of each statement of his listing against the "Money" listing given by program "CH5." He finds one more error in statement 22. He typed `(S-10)` instead of `(S-1)`. Using (-CHAR), he makes this last (he hopes) change, presses (END LINE), and with some humility, presses (RUN).

Out comes victory. His output is identical to that shown below.

13. Reference: "CH5," second printout, step 5

Here's the "MONEY" program output:

```
SQUARE 1 HAS $ 1
SQUARE 2 HAS $ 2
SQUARE 3 HAS $ 4
SQUARE 4 HAS $ 8
SQUARE 5 HAS $ 16
SQUARE 6 HAS $ 32
SQUARE 7 HAS $ 64
SQUARE 8 HAS $ 128
SQUARE 9 HAS $ 256
SQUARE 10 HAS $ 512
SQUARE 11 HAS $ 1024
SQUARE 12 HAS $ 2048
SQUARE 13 HAS $ 4096
SQUARE 14 HAS $ 8192
SQUARE 15 HAS $ 16384
SQUARE 16 HAS $ 32768
SQUARE 17 HAS $ 65536
SQUARE 18 HAS $ 131072
SQUARE 19 HAS $ 262144
SQUARE 20 HAS $ 524288
SQUARE 21 HAS $ 1048576
SQUARE 22 HAS $ 2097152
SQUARE 23 HAS $ 4194304
SQUARE 24 HAS $ 8388608
SQUARE 25 HAS $ 16777216
SQUARE 26 HAS $ 33554432
SQUARE 27 HAS $ 67108864
SQUARE 28 HAS $ 134217728
SQUARE 29 HAS $ 268435456
SQUARE 30 HAS $ 536870912
SQUARE 31 HAS $ 1073741824
SQUARE 32 HAS $ 2147483648
SQUARE 33 HAS $ 4294967296
SQUARE 34 HAS $ 8589934592
SQUARE 35 HAS $ 17179869184
SQUARE 36 HAS $ 34359738368
SQUARE 37 HAS $ 68719476736
SQUARE 38 HAS $ 137438953472
SQUARE 39 HAS $ 274877906944
SQUARE 40 HAS $ 549755813888
```

14. Reference: "CH5," second printout, steps 6 and 7

To get a listing of your "MONEY" program:

a. On the screen:
Press [LIST].

b. On the printer:

Set the printer address by typing `PRINTER IS 701` and pressing the [END LINE] key.

Press [PLST] (that is, press [SHIFT] + [PLST LIST]).

15. Reference: "CH5," second printout, step 10

    To store your "MONEY" program:

    a. Be sure your BASIC Training disc is inserted into the disc drive.

    b. Press (S)(T)(O)(R)(E)▲ (or press the STORE typing aid (k7)), (")(M)(O)(N)(E)(Y)(.)(B)(A)(S)(I)(C)(").

    c. Press (END LINE).

    d. Wait for the disc drive to stop (wait for the red light to go out).

16. Reference: "CH5," second printout, step 12

    To confirm that your "MONEY" program is stored properly on your disc:

    a. Press (k12) ((SHIFT) + (k12),) or type SCRATCH.

    b. Press (END LINE) (this executes the SCRATCH command).

    c. Press (LIST) (that is press the (PLST LIST) key). The number on your screen is the maximum memory now available in your HP-86/87.

    d. Press (L)(O)(A)(D)▲ (or press the LOAD typing aid (k6))(")(M)(O)(N)(E)(Y)(.)(B)(A)(S)(I)(C)(")(END LINE).

    e. Press (LIST) and see the listing of your "MONEY" program.

17. Reference: "CH5," second printout, step 13

    To load program "CH5" and start at line 6000,

    a. Insert your BASIC Training disc into the disc drive.

    b. Press (L)(O)(A)(D)▲ (or press the LOAD typing aid (k6))(")(C)(H)(5)(.)(B)(A)(S)(I)(C)(")(END LINE). If you get an error message, check to see that your BASIC Training disc is properly inserted in the drive. Then, clear your screen by pressing (CLEAR) ((SHIFT) + (CLEAR LINE)) and try again. And check your keystrokes.

    c. Wait for the disc drive light to go out.

    d. Press (R)(U)(N)▲(6)(0)(0)(0)(END LINE).

18. Reference: "CH5," display starting at line 6000

    To roll the screen up, press (ROLL▲) (that is press (SHIFT) + (▲ROLL▼).

    To roll the screen down, press (ROLL▼) (that is, press the (▲ROLL▼) key).

19. Reference: Page 6-2, step 3

    To execute the SCRATCH command:
    Press the (k12) or type SCRATCH and press (END LINE).

    To press (RESET).
    Press (SHIFT) + (RESET).

20. Reference: Page 6-2, step 8

Note: The symbol ▲ means press the space bar to produce one space.

Here's how line 125 of "Calculate" is entered. If you have trouble, clear your screen by pressing [CLEAR] ([SHIFT] + [CLEAR LINE]) and try again. Just press the keys in the order shown. Press [1] [2] [5] ▲ [P] [R] [I] [N] [T] ▲ ["] [T] [H] [E] ▲ [S] [U] [M] ▲ [O] [F] ▲ [T] [H] [E] [S] [E] ▲ [4] ▲ [R] [E] [S] [U] [L] [T] [S] ▲ [I] [S] ["] [;] [(] [4] [+] [2] [)] [+] [(] [4] [-] [2] [)] [+] [(] [4] [*] [2] [)] [+] [(] [4] [/] [2] [)] [END LINE].

21. Reference: Page 6-3, steps 13 and 14

Note: The symbol ▲ means press the space bar to produce one space.

The "Seven Come Eleven" program may be written using these keystrokes. If you have trouble, clear your screen by pressing [CLEAR] ([SHIFT] + [CLEAR LINE]) and try again.

a. Scratch memory: press [SHIFT] + [k12] [END LINE].

b. Press [1] [0] ▲ [D] [I] [S] [P] ▲ ["] [S] [E] [V] [E] [N] ▲ [R] [A] [I] [S] [E] [D] ▲ [T] [O] ▲ [T] [H] [E] ["] [END LINE].

c. Press [2] [0] ▲ [D] [I] [S] [P] ▲ ["] [E] [L] [E] [V] [E] [N] [T] [H] ▲ [P] [O] [W] [E] [R] ▲ [I] [S] ["] [;]
[7] [^] [1] [1] [END LINE].

d. Press [3] [0] ▲ [E] [N] [D] [END LINE].

e. To run your program, press [RUN].

f. To list your program on the printer, press [PLST] (that is, press [SHIFT] + [PLST LIST]).

22. Reference: Page 6-3, step 15

The output of your "Seven Come Eleven" program should look like this:

```
SEVEN RAISED TO THE
ELEVENTH POWER IS 1977326743
```

23. Reference: Page 6-4, steps 21-23

Note: The symbol ▲ means press the space bar to produce one space.

Step 21:

The second statement of the "Binary Brain" program may be entered by pressing:
[2] [0] ▲ [P] [R] [I] [N] [T] ▲ ["] [B] [I] [N] [A] [R] [Y] ▲ [B] [R] [A] [I] [N] ['] [S] ▲ [N] [U] [M] [B] [E] [R] ▲ [I] [S] ["] [;] [6] [*] [2] [*] [6] [*] [3] [*] [9] [END LINE].

Step 22: To run "Binary Brain," press (RUN).

Step 23: To list your program on the printer, press (PLST); that is, press (SHIFT) + (PLST LIST).

24. Reference: Page 6-4, step 25

Here's an acceptable "Binary Brain" printed listing:

```
10 PRINTER IS 701
20 PRINT "BINARY BRAIN'S NUMBER IS";6*2*6*3*9
30 PRINT "FLYING FINGER'S NUMBER IS";8*9*7*4*1
40 END
```

and output:

```
BINARY BRAIN'S NUMBER IS 1944
FLYING FINGER'S NUMBER IS 2016
```

25. Reference: Page 6-5, steps 31 and 32

Note: The symbol ▲ means press the space bar to produce one space.

Step 31:

Here are the keystrokes for lines 30, 40 and 50 of the "Semicolon, Comma" program. Just press the keys as shown.

a. Press (2) (0) ▲ (P) (R) (I) (N) (T) (END LINE).

b. Press 3 0 ▲ P R I N T ▲ " A " ▲ S E M I C O L O N ▲ D O E S ▲ T H I S : " END LINE.

c. Press 4 0 ▲ P R I N T ▲ " 5 " ; " 1 0 " ; " 1 5 " END LINE.

Step 32:

a. Press RUN.

b. Press CLEAR (SHIFT + CLEAR -LINE).

c. Press LIST (PLST LIST).

d. Press the ON LINE key on your printer to take it off line. The ON LINE light should go out.
Press the LF key (line feed), to generate some blank paper.
Press the ON LINE key again, to put the printer back on line.

e. Press PLST (SHIFT + PLST LIST).

f. Press the ON LINE key to take the printer off line.
Press the LF key (line feed), so that the paper clears the top of the printer bar.
Put the printer back ON LINE.

26. Reference: Page 7-2, step 10

To load and run program "CH7," press k6 " C H 7 . B A S I C " END LINE. When the disc drive light goes out, press RUN. If you have trouble, clear your screen by pressing CLEAR (SHIFT + CLEAR -LINE) and start Help No. 26 over again.

27. Reference: "CH7," printout, step 2

Note: The symbol ▲ means press the space bar to produce one space.

To enter and run the "A + 2" program:

a. Press k12, (SHIFT + k12), END LINE to clear the HP-86/87's memory. (This removes any program that may be in the HP-86/87's memory.) If this is not done, your statements may get mixed up with statements of another program, which could result in neither program working.

b. Here are the keystrokes for entering statement 75:
7 5 ▲ D I S P ▲ " A = " ; A END LINE.

c. To run your program, simply press the RUN key.

28. Reference: Page 7-5, step 21

To load "CH7" and start at line 2000, press k6 " C H 7 . B A S I C " END LINE. When the disc drive light goes out, press R U N ▲ 2 0 0 0 END LINE. If you have trouble, clear your screen by pressing CLEAR (SHIFT + CLEAR -LINE) and start Help No. 28 over again.

29. Reference: Page 7-13, step 49

Here is one way to write the "23 Skidoo" program:

```
5 PRINTER IS 701
10 LET A=5
20 LET B=A*8/2+3
30 PRINT "B=";B
40 END
```

When this program is run, it produces:

```
B= 23
```

30. Reference: Page 7-15, step d

Note: This is the last Help message I'll give you on how to load a program from your disc and run it.

To load and run "REVIEW":

a. Insert your BASIC Training disc into the disc drive.

b. Press [k6] ["] [R] [E] [V] [I] [E] [W] [.] [B] [A] [S] [I] [C] ["] [END LINE].

c. When the disc drive light goes out, press [RUN].

d. If you have trouble, clear your screen by pressing [CLEAR] ([SHIFT] + [CLEAR LINE]) and start over at step b above.

31. Reference: Page 7-15, step f

To enter your answer number 4, press [4] [END LINE].

32. Reference: "CH8," final printout, "Savings" program

Note: The symbol ▲ means press the space bar to produce one space.

a. To enter statement 120, press the following keys:

[1] [2] [0] ▲ [D] [I] [S] [P] ▲ ["] [A] [F] [T] [E] [R] ["] [;] [y] [;] ["] [Y] [E] [A] [R] [S] ▲ [Y] [O] [U] [R] ▲ [O] [R] [I] [G] [I] [N] [A] [L] ▲ [$] ["] [;] [B] [;] ["] [W] [I] [L] [L] ▲ [G] [R] [O] [W] ▲ [T] [O] ▲ [$] ["] [;] [END LINE]

[1] [3] [0] ▲ [D] [I] [S] [P] ▲ [I] [N] [T] [(] [B] [*] [(] [1] [+] [I] [/] [1] [0] [0] [)] [^] [Y] [*] [1] [0] [0] [+] [.] [5] [)] [/] [1] [0] [0] [;] ["] [.] ["] [END LINE]

b. Here is a PRINTALL printout of the display produced by "SAVINGS" when the following values are entered:

B = 1000
I = 6.25
Y = 10

```
              SAVINGS

WHAT IS YOUR PRESENT SAVINGS ACCOUNT BALANCE
?
1000

WHAT IS S&L INTEREST (%)
?
6.25

HOW MANY YEARS WILL SAVINGS BE HELD
?
10

AFTER 10 YEARS YOUR ORIGINAL $ 1000 WILL GROW TO $ 1833.54 .
```

33. Reference: Page 8-16, step 52

Listing and output of "SAVINGS" with the INPUT question mark moved to the input message.

```
10 DISP "              SAVINGS"
20 DISP
30 DISP "WHAT IS YOUR PRESENT SAVINGS ACCOUNT BALANCE";
40 INPUT B
50 DISP
60 DISP "WHAT IS S&L INTEREST (%)";
70 INPUT I
80 DISP
90 DISP "HOW MANY YEARS WILL SAVINGS BE HELD";
100 INPUT Y
110 DISP
120 DISP "AFTER";Y;"YEARS YOUR ORIGINAL $";B;"WILL GROW TO $";
130 DISP INT (B*(1+I/100)^Y*100+.5)/100;"."
140 END
```

```
              SAVINGS

WHAT IS YOUR PRESENT SAVINGS ACCOUNT BALANCE?
100

WHAT IS S&L INTEREST (%)?
8.06

HOW MANY YEARS WILL SAVINGS BE HELD?
6

AFTER 6 YEARS YOUR ORIGINAL $ 100 WILL GROW TO $ 159.22 .
```

34. Reference: Page 9-6, step 6

Listing of "Inflation" program and typical PRINTALL printed output

```
10 DISP "                  INFLATION"
20 DISP
30 DISP "WHAT IS YOUR PRESENT SAVINGS ACCOUNT BALANCE";
40 INPUT B
50 DISP
52 DISP "WHAT IS INFLATION RATE (%)";
54 INPUT F
56 DISP
60 DISP "WHAT IS S&L INTEREST (%)";
70 INPUT I
80 DISP
90 DISP "HOW MANY YEARS WILL SAVINGS BE HELD";
100 INPUT Y
110 DISP
120 DISP "AFTER";Y;"YEARS YOUR ORIGINAL $";B;"WILL GROW TO $";INT (B*(1+I/100)^Y
*100+.5)/100;"."
130 DISP
140 DISP "AFTER";Y;"YEARS IT WILL COST $";INT (B*(1+F/100)^Y*100+.5)/100;"TO BUY
 WHAT $";B;
150 DISP "BUYS TODAY."
160 END
```

```
                    INFLATION

WHAT IS YOUR PRESENT SAVINGS ACCOUNT BALANCE?
500

WHAT IS INFLATION RATE (%)?
10.5

WHAT IS S&L INTEREST (%)?
5.25

HOW MANY YEARS WILL SAVINGS BE HELD?
5

AFTER 5 YEARS YOUR ORIGINAL $ 500 WILL GROW TO $ 645.77 .

AFTER 5 YEARS IT WILL COST $ 823.72 TO BUY WHAT $ 500 BUYS TODAY.
```

35. Reference: Page 10-10.

Here is one set of answers to the four programming questions for "Pail."

a. What answers do I want?

I want to find out how long it will take Pail-Face to fill his pool. I want this answer expressed in four ways—in total minutes, hours, days and years. I want this answer for a variety of pool sizes, various useful pail capacities, and different intervals of time needed for one pail filling and emptying round trip.

b. What things do I know?

I know the formulas needed to solve this problem. They're almost identical to the "Mole" program formulas.

c. What methods will I use to find answers using things I know?

I'll choose values for pool dimensions, useful pail capacity, plus the time interval needed for one round trip with the pail. Then I'll solve the formulas.

d. How can BASIC and the HP-86/87 help me find answers?

I'll use `DISP` and `INPUT` statements. First the HP-86/87 will display a request for pool dimension values (for W, L and D values). Pool volume Vol will then be calculated and displayed. Next, the HP-86/87 will display a request for pail volume Pail, and then it will calculate and display Pails, the number of pailfuls needed to fill the pool. A request for MinPail will be displayed, where MinPail is the minutes needed for one pail carrying roundtrip, and finally the total pool filling time expressed four ways—minutes, hours, days and years (Minutes, Hrs, Days and Yrs)—will be calculated and displayed.

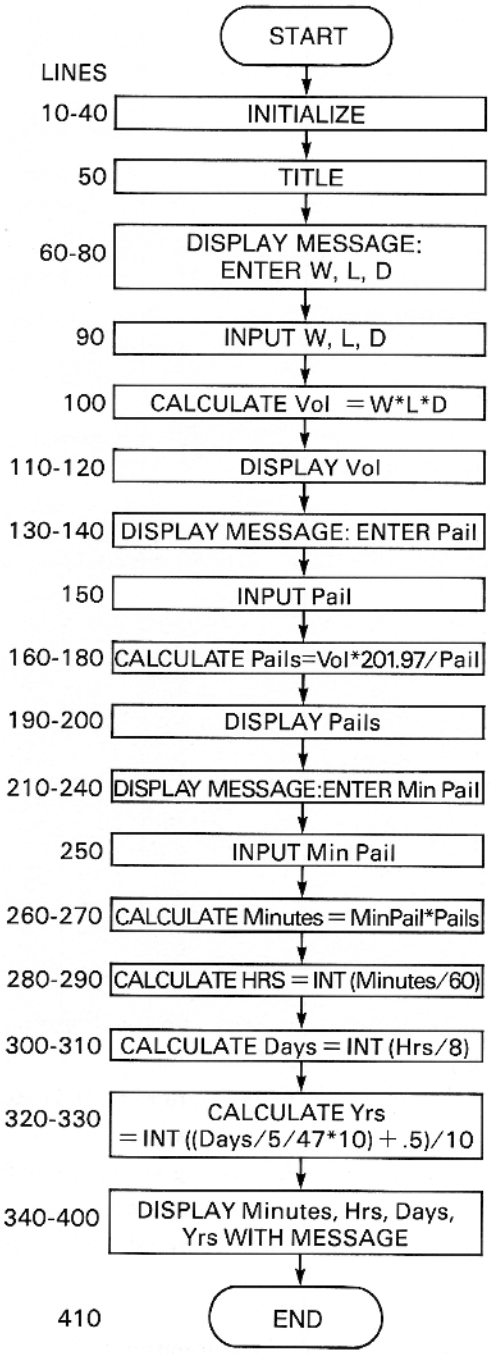
Listing for "Pail" Program

Reference: Page 10-10

```
10 ! "PAIL" PROGRAM
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 DISP "              PAIL"
60 DISP
70 DISP "ENTER WIDTH, LENGTH AND AVERAGE DEPTH OF SWIMMING POOL IN YARDS."
80 DISP "SEPARATE NUMBERS BY COMMAS."
90 INPUT W,L,D
100 LET Vol=W*L*D
110 DISP
120 DISP "POOL VOLUME IS";Vol;"CUBIC YARDS."
130 DISP
140 DISP "ENTER THE EFFECTIVE VOLUME OF PAIL IN GALLONS."
150 INPUT Pail
160 DISP
170 ! Pails=PAILFULS OF WATER NEEDED TO FILL POOL
180 LET Pails=Vol*201.97/Pail
190 DISP
200 DISP "THE NUMBER OF PAILFULS OF WATER REQUIRED TO FILL THE POOL IS";Pails;".
"
210 DISP
220 DISP "ENTER TIME IN MINUTES REQUIRED TO EMPTY ONE PAIL OF WATER INTO POOL."
230 DISP "INCLUDE ENOUGH TIME TO COVER COFFEE BREAKS, CARRYING PAIL TO FAUCET,"
240 DISP "FILLING PAIL, CARRYING PAIL TO POOL, ETC."
250 INPUT MinPail
260 ! Minutes=MINUTES NEEDED TO FILL POOL
270 LET Minutes=MinPail*Pails
280 ! Hrs=HOURS NEEDED TO FILL POOL
290 LET Hrs=INT (Minutes/60)
300 ! Days=DAYS NEEDED TO FILL POOL
310 LET Days=INT (Hrs/8)
320 ! Yrs=YEARS NEEDED TO FILL POOL
330 LET Yrs=INT (Days/5/47*10+.5)/10
340 DISP
350 DISP "TIME NEEDED TO FILL POOL:"
360 DISP
370 DISP "  MINUTES:  ";Minutes
380 DISP "  HOURS:    ";Hrs
390 DISP "  DAYS:     ";Days
400 DISP "  YEARS:    ";Yrs
410 END
```

Flowchart for "Pail" Program

Reference: Page 10-10



Dimensions of pool in yards
- W = Width
- L = Length
- D = Average depth
- Vol = Pool volume in cubic yards
- Pail = Useable volume of pail in gallons
- Pails = Pailfuls of water needed to fill pool

MinPail = Minutes needed for one pail filling and emptying round trip

Time needed to fill pool
- Minutes: In minutes
- Hrs: In Hours
- Days: In days
- Yrs: In years

Here is a listing of the "SECRET" message omitting all GOTO statements.

Reference: Page 11-9

```
1000 PRINT
1010 PRINT TAB (25);"ABRAHAM LINCOLN"
1020 PRINT
1030 PRINT TAB (16);"EXCERPT FROM";
1040 PRINT " THE SECOND INAUGURAL"
1050 PRINT
1060 PRINT "FONDLY DO WE HOPE - FERVENTLY DO";
1070 PRINT " WE PRAY - THAT THIS MIGHTY"
1080 PRINT "SCOURGE OF WAR MAY SPEEDILY PASS AWAY."
1090 PRINT "YET, IF GOD WILLS THAT IT CONTINUE,";
1100 PRINT " UNTIL ALL THE WEALTH PILED"
1110 PRINT "BY THE BOND-MAN'S TWO HUNDRED AND";
1120 PRINT " FIFTY YEARS OF UNREQUITED TOIL"
1130 PRINT "SHALL BE SUNK, AND UNTIL EVERY";
1140 PRINT " DROP OF BLOOD DRAWN WITH THE LASH"
1150 PRINT "SHALL BE PAID BY ANOTHER DRAWN";
1160 PRINT " WITH THE SWORD, AS WAS SAID THREE"
1170 PRINT "THOUSAND YEARS AGO, SO STILL IT";
1180 PRINT " MUST BE SAID 'THE JUDGEMENTS OF"
1190 PRINT "THE LORD ARE TRUE AND";
1200 PRINT " RIGHTEOUS ALTOGETHER.'"
1210 END
```

And here is the message itself:

```
                    ABRAHAM LINCOLN

                EXCERPT FROM THE SECOND INAUGURAL

FONDLY DO WE HOPE - FERVENTLY DO WE PRAY - THAT THIS MIGHTY
SCOURGE OF WAR MAY SPEEDILY PASS AWAY.
YET, IF GOD WILLS THAT IT CONTINUE, UNTIL ALL THE WEALTH PILED
BY THE BOND-MAN'S TWO HUNDRED AND FIFTY YEARS OF UNREQUITED TOIL
SHALL BE SUNK, AND UNTIL EVERY DROP OF BLOOD DRAWN WITH THE LASH
SHALL BE PAID BY ANOTHER DRAWN WITH THE SWORD, AS WAS SAID THREE
THOUSAND YEARS AGO, SO STILL IT MUST BE SAID 'THE JUDGEMENTS OF
THE LORD ARE TRUE AND RIGHTEOUS ALTOGETHER.'
```

Here is an easy way to rewrite "SECRET" so all GOTO statements are removed. First, run the program, "SECRET." Now follow the instructions on the printed listing given by statement 60, that is, get statement 400 displayed on the screen (L I S T ▲ 4 0 0 END LINE). Renumber it 1000. Remember to press END LINE. You might write 1000 on a piece of paper so you will know what the next available number is (1010). Next, follow the trail indicated by statement 410. After listing statement 300 on the screen, renumber it as 1010. Write the number 1010 down. Continue in the same way. When you're done; that is, when you've renumbered 520 as 1200 and 560 as 1210, then renumber 560 as 1220. Execute PLIST 1000 to print a listing. Check your work by executing RUN 1000.

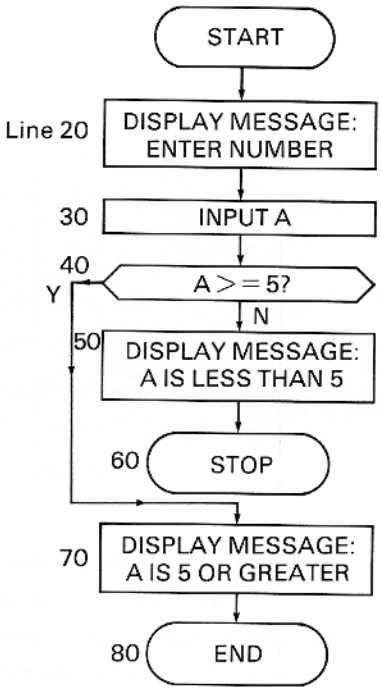
An easy way to print this message without rewriting the program is to execute the command RUN 60 (R U N ▲ 6 0 END LINE). List "SECRET" and inspect statement 50. Notice how it prevents the execution of statement 60 when "SECRET" is run by pressing RUN.

Here is a listing for "Less Than Five" using STOP.

Reference: Page 12-5

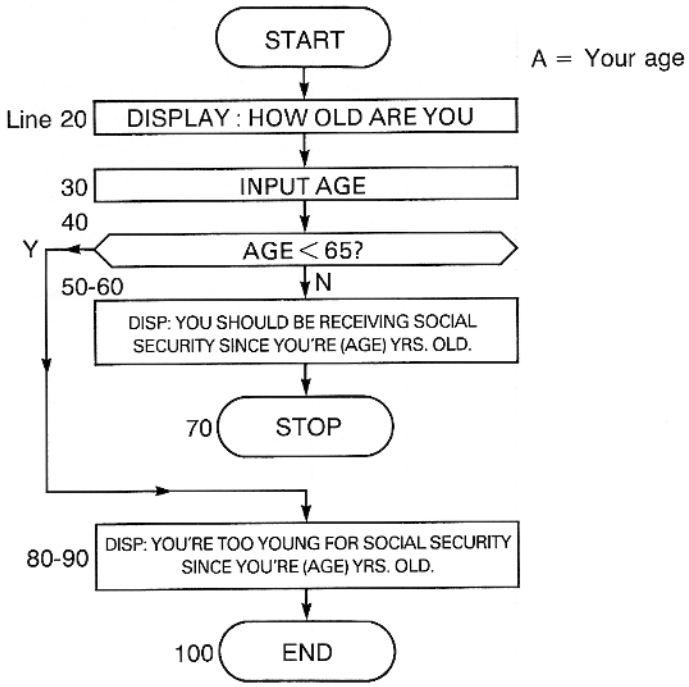
```
10 ! LESS THAN FIVE
20 DISP "ENTER ANY NUMBER."
30 INPUT A
40 IF A>= 5 THEN 70
50 DISP "YOUR NUMBER IS LESS THAN FIVE."
60 STOP
70 DISP "YOUR NUMBER IS 5 OR GREATER."
80 END
```

And here is its flowchart.



Flowchart for "Social Security, Anyone?"

Reference: Page 12-5



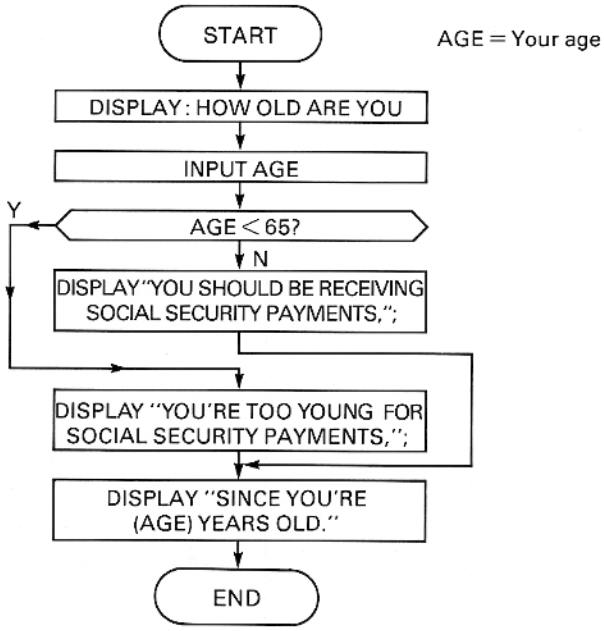
"Social Security, Anyone?" shortened by one step

Reference: Page 12-5

Listing for shortened "Social Security, Anyone?"

```
10 ! SOCIAL SECURITY, ANYONE?
20 DISP "HOW OLD ARE YOU";
30 INPUT AGE
40 IF AGE<65 THEN 80
50 DISP "YOU SHOULD BE RECEIVING SOCIAL SECURITY PAYMENTS, ";
60 GOTO 90
80 DISP "YOU'RE TOO YOUNG FOR SOCIAL SECURITY PAYMENTS, ";
90 DISP "SINCE YOU'RE";AGE;"YEARS OLD."
100 END
```

Flowchart for shortened "Social Security, Anyone?"



One way to answer the programming questions for "Size"

Reference: Page 12-7

a. What answers do I want?

I want the user of this program to find out if he or she is about average height, less than average height, or greater than average height. A man's height should be compared with the average height for men, and a woman's height should be compared with the average height for women.

b. What things do I know?

I know the average height range for men is 68, 69, 70, and 71 inches, and the average height range for women is 63, 64, 65, and 66 inches.

c. What methods will I use to find answers using things I know? (How would I solve the problem with paper and pencil?)

Knowing the person's sex and height, I'll compare that height with the average range for that person's sex. Then I'll know if the person's height is below, about equal to, or above average.

d. How can BASIC and the HP-86/87 help me find answers?

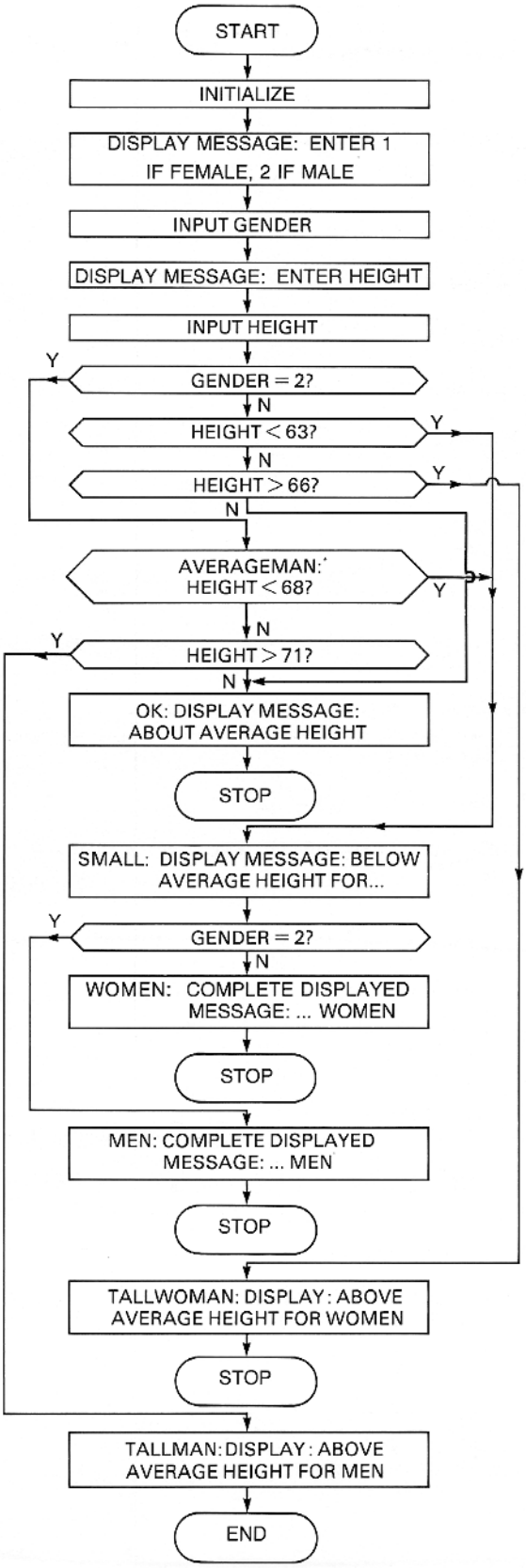
The user will input a 1 for female, 2 for male. Based on this input, the program will branch. Each branch will compare the entered height against the average range for the user's sex and an appropriate message will be displayed: below, about equal, or above average height.

Flow chart for "Size"

Reference: Page 12-7

GENDER = 1, if female
2, if male

HEIGHT = Height in inches



Listing and output for "Size." Listing copied from screen.

Reference: Page 12-7

```
10 ! SIZE
20 NORMAL
30 CLEAR
40 DISP
50 DISP
60 DISP "IF YOU ARE FEMALE, ENTER 1; IF MALE, ENTER 2."
70 INPUT GENDER
80 DISP
90 DISP "WHAT IS YOUR HEIGHT IN INCHES";
100 INPUT HEIGHT
110 IF GENDER=2 THEN AVERAGEMAN
120 DISP
130 IF HEIGHT<63 THEN SMALL
140 IF HEIGHT>66 THEN TALLWOMAN
150 GOTO OK
160  AVERAGEMAN: DISP
170 IF HEIGHT<68 THEN SMALL
180 IF HEIGHT>71 THEN TALLMAN
190  OK: DISP "YOU'RE NOT TOO TALL AND YOU'RE NOT TOO SHORT. YOU'RE JUST RIGHT."

200 STOP
210  SMALL: DISP "GOOD NEWS! WHEN YOU FALL, YOU HIT THE GROUND MORE SOFTLY THAN
";
220 IF GENDER=2 THEN MEN
230 DISP "MOST OTHER WOMEN."
240 STOP
250  MEN: DISP "MOST OTHER MEN."
260 STOP
270  TALLWOMAN: DISP "MOST WOMEN LOOK UP TO YOU."
280 STOP
290  TALLMAN: DISP "MOST MEN LOOK UP TO YOU."
300 END
```

```
IF YOU ARE FEMALE, ENTER 1; IF MALE, ENTER 2.
?
1

WHAT IS YOUR HEIGHT IN INCHES?
64

YOU'RE NOT TOO TALL AND YOU'RE NOT TOO SHORT. YOU'RE JUST RIGHT.
```

Listing for "Sports Quiz"

Reference: Page 12-10

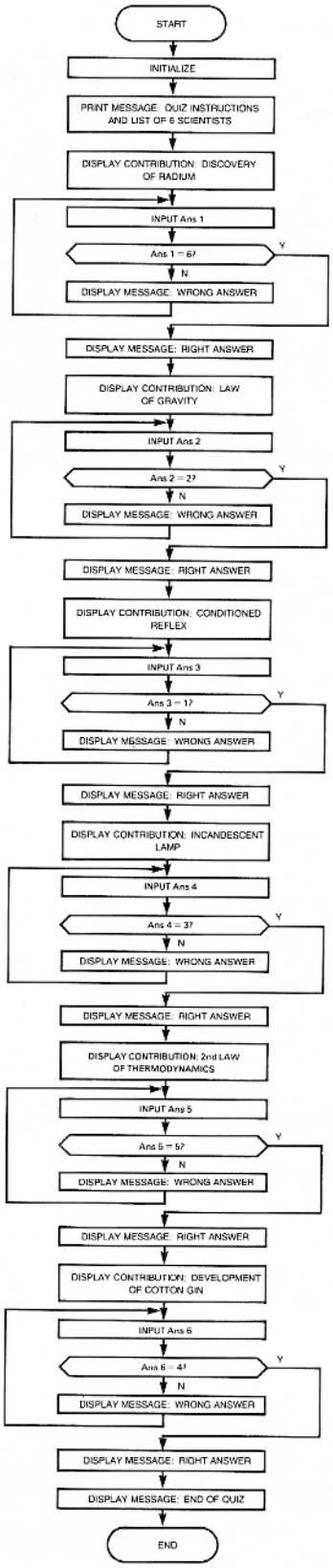
```
10 ! "SPORTS QUIZ"
20 NORMAL
30 CLEAR
40 PRINTER IS 701,80
50 PRINT
60 PRINT "                    SPORTS QUIZ"
70 PRINT
80 PRINT "THIS PROGRAM WILL TEST YOUR KNOWLEDGE OF THE SPORTS WORLD."
90 PRINT
100 PRINT "A SERIES OF SPORTS EVENTS WILL BE DISPLAYED ON THE SCREEN,"
110 PRINT "ONE AT A TIME.  AFTER EACH EVENT RESPOND BY TYPING THE NUMBER"
120 PRINT "OF THE ATHLETE, FROM THE LIST BELOW, THAT CORRESPONDS TO THE EVENT."
130 PRINT "AFTER TYPING THE NUMBER, PRESS THE (END LINE) KEY."
140 PRINT
150 PRINT
160 PRINT "THE ATHLETES:"
170 PRINT
180 PRINT "#1  MARK SPITZ"
190 PRINT "#2  DON LARSEN"
200 PRINT "#3  NADIA COMANECI"
210 PRINT "#4  MICKEY WRIGHT"
220 PRINT "#5  BOB BEAMON"
230 PRINT "#6  ROGER BANNISTER"
240 DISP
250 DISP
260 DISP "FOUR MINUTE BARRIER IS BROKEN IN THE MILE RUN."
270 INPUT Ans1
280 DISP
290 IF Ans1=6 THEN 320
300 DISP "SORRY, TRY AGAIN."
310 GOTO 270
320 DISP "CORRECT!"
330 DISP
340 DISP "FIRST PERFECT GAME IN WORLD SERIES HISTORY."
350 INPUT Ans2
360 DISP
370 IF Ans2=2 THEN 400
380 DISP "NOPE, TRY ANOTHER."
390 GOTO 350
400 DISP "VERY GOOD. YOU KNOW YOUR BASEBALL."
410 DISP
420 DISP "FIRST OLYMPIAN TO WIN 7 GOLD MEDALS IN A SINGLE OLYMPICS."
430 INPUT Ans3
440 DISP
450 IF Ans3=1 THEN 480
460 DISP "WRONG ANSWER, BUT KEEP TRYING."
470 GOTO 430
480 DISP "THAT'S RIGHT. IN MUNICH, 1972."
490 DISP
500 DISP "FIRST PERFECT SCORE IN OLYMPIC GYMNASTICS."
510 INPUT Ans4
520 DISP
530 IF Ans4=3 THEN 560
540 DISP "WRONG. TRY AGAIN."
550 GOTO 510
560 DISP "RIGHT. NOW TO TRACK AND FIELD."
570 DISP
580 DISP "WORLD RECORD SHATTERED IN LONG JUMP."
    :
```

```
590 INPUT Ans5
600 DISP
610 IF Ans5=5 THEN 640
620 DISP "SORRY, TRY ANOTHER."
630 GOTO 590
640 DISP "RIGHT AGAIN. ONE MORE QUESTION."
650 DISP
660 DISP "MOST GOLF VICTORIES IN A SINGLE YEAR."
670 INPUT Ans6
680 DISP
690 IF Ans6=4 THEN 720
700 DISP "WRONG ANSWER, BUT TRY AGAIN."
710 GOTO 670
720 DISP "YOU ARE CORRECT."
730 DISP
740 DISP "THAT CONCLUDES OUR SPORTS QUIZ."
750 END
```

Flowchart for "Science Quiz"

Reference: Page 12-10

1 =Ivan Pavlov
2 = Isaac Newton
3 = Thomas Edison
4 = Eli Whitney
5 = Lord Kelvin
6 = Marie Curie
Ans 1 = Answer to 1st contribution
Ans 2 = Answer to 2nd contribution
Ans 3 = Answer to 3rd contribution
Ans 4 = Answer to 4th contribution
Ans 5 = Answer to 5th contribution
Ans 6 = Answer to 6th contribution



Listing for "Science Quiz"

Reference: Page 12-10

```
10 ! SCIENCE QUIZ
20 NORMAL
30 CLEAR
40 PRINTER IS 701,80
50 PRINT "THIS PROGRAM WILL TEST YOUR KNOWLEDGE OF THE WORLD OF SCIENCE."
60 PRINT "A SERIES OF SCIENTIFIC CONTRIBUTIONS WILL BE DISPLAYED ON THE SCREEN,"

70 PRINT "ONE AT A TIME. AFTER EACH CONTRIBUTION RESPOND BY TYPING THE NUMBER"
80 PRINT "OF THE SCIENTIST, FROM THE LIST BELOW, THAT CORRESPONDS TO THE CONTRIB
UTION."
90 PRINT "AFTER TYPING THE NUMBER, PRESS THE (END LINE) KEY."
100 PRINT
110 PRINT
120 PRINT "THE SCIENTISTS:"
130 PRINT
140 PRINT "#1  IVAN PAVLOV"
150 PRINT "#2  SIR ISAAC NEWTON"
160 PRINT "#3  THOMAS A. EDISON"
170 PRINT "#4  ELI WHITNEY"
180 PRINT "#5  LORD KELVIN"
190 PRINT "#6  MARIE CURIE"
200 DISP
210 DISP
220 DISP "DISCOVERY OF RADIUM, LEADING TO RADIOTHERAPY."
230 INPUT Ans1
240 IF Ans1=6 THEN 270
250 DISP "SORRY, TRY AGAIN."
260 GOTO 230
270 DISP "CORRECT!"
280 DISP
290 DISP "LAW OF GRAVITY IN 1687."
300 INPUT Ans2
310 DISP
320 IF Ans2=2 THEN 350
330 DISP "NOPE, TRY ANOTHER."
340 GOTO 300
350 DISP "VERY GOOD."
360 DISP
370 DISP "CONDITIONED REFLEX, 1910."
380 INPUT Ans3
390 DISP
400 IF Ans3=1 THEN 430
410 DISP "WRONG ANSWER, BUT KEEP TRYING."
420 GOTO 380
430 DISP "THAT'S RIGHT."
440 DISP
450 DISP "INCANDESCENT LAMP, 1879."
460 INPUT Ans4
470 DISP
480 IF Ans4=3 THEN 510
490 DISP "WRONG. TRY AGAIN."
500 GOTO 460
510 DISP "RIGHT."
520 DISP
530 DISP "2ND LAW OF THERMODYNAMICS IN 1851."
540 INPUT Ans5
550 DISP
560 IF Ans5=5 THEN 590
570 DISP "SORRY, TRY ANOTHER."
580 GOTO 540
```

```
590 DISP "RIGHT AGAIN."
600 DISP
610 DISP "DEVELOPMENT OF THE COTTON GIN, 1793."
620 INPUT Ans6
630 DISP
640 IF Ans6=4 THEN 670
650 DISP "WRONG ANSWER, BUT TRY AGAIN."
660 GOTO 620
670 DISP "YOU ARE CORRECT."
680 DISP
690 DISP "THAT CONCLUDES OUR SCIENCE QUIZ."
700 END
```

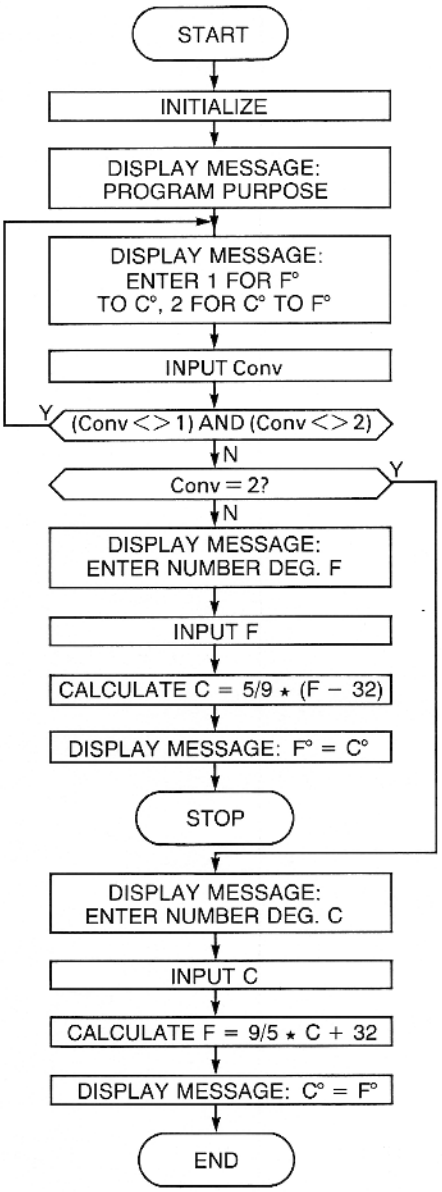
Listing for "Temperature Conversion"

Reference: Page 12-13

```
10 ! TEMPERATURE CONVERSION
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 DISP
60 DISP
70 DISP "THIS PROGRAM CONVERTS BETWEEN CELSIUS AND FAHRENHEIT DEGREES."
80 DISP
90 DISP
100 DISP "YOU HAVE TWO CHOICES:"
110 DISP
120 DISP "1. CONVERT FROM F TO C DEGREES."
130 DISP "2. CONVERT FROM C TO F DEGREES."
140 DISP
150 DISP "TYPE THE NUMBER OF YOUR CHOICE AND PRESS THE (END LINE) KEY."
160 INPUT Conv
170 DISP
180 IF Conv<> 1 AND Conv<> 2 THEN 100
190 IF Conv=2 THEN 270
200 DISP "TYPE THE NUMBER OF DEGREES F AND PRESS (END LINE)."
210 INPUT F
220 C=INT (5/9*(F-32)*10+.5)/10
230 DISP
240 DISP
250 DISP F;"DEG F =";C;"DEG C."
260 STOP
270 DISP "TYPE THE NUMBER OFF DEGREES C AND PRESS (END LINE)."
280 INPUT C
290 F=INT ((9/5*C+32)*10+.5)/10
300 DISP
310 DISP C;"DEG C =";F;"DEG F."
320 END
```

Flowchart for "Temperature Conversion"

Reference: Page 12-13



Conv = Conversion choice number:

1: F° → C°
2: C° → F°

F = Number degrees Fahrenheit

C = Number degrees Celsius

Here are my answers to the programming questions for "Sweepstakes."

Reference: Page 12-19

a. What answers do I want?

I want to know if I'll win the HP-86/87 Sweepstakes, and if so, how much money will I win.

b. What things do I know?

I know a group of key expressions and definitions.

c. What methods will I use to find answers using things I know?

Using expression 1, `LUCKYNUM=INT (9999*RND+1)`, I'll calculate the lucky number `LUCKYNUM`. Then, after the certificate number `CERT` is entered, I'll test it in two ways. First, if it's smaller than 1 or larger than 9999, I'll reject it. Second, if it's not an integer, I'll reject it. If `CERT` passes both tests, I'll check to see if `CERT` is a winner.

I'll first find out if I'm a $1,000,000 winner by checking to see if `CERT` is within 50 of `LUCKYNUM`. If it is, I'll enter another certificate number and try to win again.

If I miss the big one, I'll use the same certificate number `CERT` to find out if I'm a $250,000 winner by checking to see if `CERT` is within 400 of `LUCKYNUM`. If it is, great! I can then enter another certificate number `CERT` and try again.

If I miss $250,000, I'll try in a similar way for $10,000 and $100. If I miss $100, I'm a loser. I then enter another certificate number and try again.

d. How can BASIC and the HP-86/87 help me find answers?

I'll make heavy use of `DISP` statements, both for displayed messages and to produce blank lines for a clearer and more pleasing display.

To calculate and put into the program a value for the lucky number `LUCKYNUM`, I'll use expression 1, `LUCKYNUM=INT (9999*RND+1)`, as an assignment statement.

I'll then use an INPUT statement with an appropriate message that will accept an entered value for `CERT`, the certificate number.

Next, I'll subject `CERT` to two tests. I'll use

`IF CERT<1 OR CERT>9999 THEN` *line number*

and

`IF CERT#INT (CERT) THEN` *line number*

to see if `CERT` is a bad number. If the answer to all these comparisons is NO, `CERT` is a good number. I'll then check to see if `CERT` is a winner. If the answer to any one comparison is YES, the number is bad, and I'll have the program calculate a new lucky number and ask for another certificate number.

To see if `CERT` wins for me $1,000,000, I need to see if `CERT` is within 50 of `LUCKYNUM`. How do I do this?

The range of numbers within 50 of LUCKYNUM is the group of numbers between LUCKYNUM-50 and LUCKYNUM+50. If CERT were greater than LUCKYNUM-50 and less than LUCKYNUM+50, I'd win the big one. In BASIC:

```
IF (CERT>(LUCKYNUM-50)) AND (CERT<(LUCKYNUM+50))
THEN line number
```

This line number will be a DISP statement that displays a message informing me of my good news. After that welcome message is displayed, I'll go to the line that calculates LUCKYNUM, a new CERT will be input, and the process will repeat. If the answer to either CERT>(LUCKYNUM-50)? or CERT<(LUCKYNUM+50)? is NO, I'll have failed to win $1,000,000. I'll then see if CERT is a $250,000 winner. To win $250,000, CERT must be within 400 of LUCKYNUM, so I'll use another IF...AND...THEN:

```
IF (CERT>(LUCKYNUM-400)) AND (CERT<(LUCKYNUM+400))
THEN line number
```

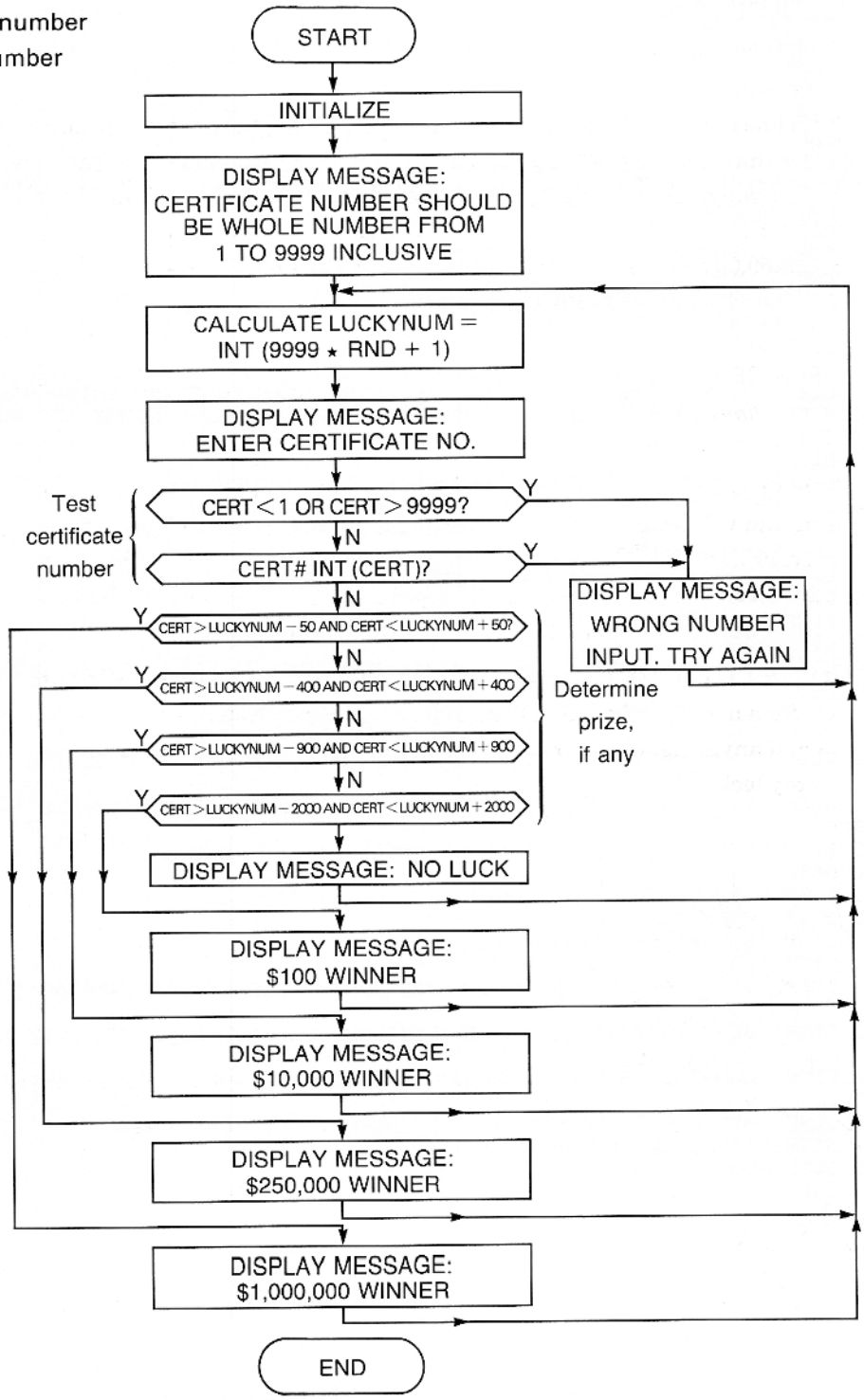
If both CERT>(LUCKYNUM-400) and CERT<(LUCKYNUM+400) are true, then the program will branch to the line number giving a "You won" kind of message. After that message, another LUCKYNUM will be calculated and another CERT will be entered, and the various tests will be run again. If CERT is not within 400 of LUCKYNUM, then a similar IF...AND... THEN will be used to see if CERT is a $10,000 number and so on.

If CERT is not a $100 winner, I'll display a "You lost" message, calculate a new LUCKYNUM, and ask for a new CERT. Then I'll once again use my IF...AND...THEN statements to see what I won, if anything. Whether I won or not, the whole cycle will repeat, and so on, for as long as I wish to try my luck.

Flowchart for "Sweepstakes"

Reference: Page 12-19

LUCKYNUM = Lucky number
CERT = Certificate number



Listing for "Sweepstakes"

Reference: Page 12-19.

```
10 ! "SWEEPSTAKES"
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 RANDOMIZE
60 DISP "                    THE HP-86/87 SWEEPSTAKES"
70 DISP
80 DISP "YOU WILL BE ASKED TO ENTER YOUR SWEEPSTAKES CERTIFICATE NUMBER."
90 DISP "THIS NUMBER SHOULD BE A WHOLE NUMBER FROM 1 TO 9999 INCLUSIVE."
100 LUCKYNUM=INT (9999*RND +1)
110 DISP
120 DISP "WHAT IS YOUR CERTIFICATE NUMBER";
130 INPUT CERT
140 IF CERT<1 OR CERT>9999 THEN 170
150 IF CERT#INT (CERT) THEN 170
160 GOTO 210
170 DISP
180 DISP "SORRY, THE SWEEPSTAKES RULES REQUIRE THAT YOU ENTER A WHOLE NUMBER"
190 DISP "LARGER THAN ZERO AND SMALLER THAN 10,000. PLEASE TRY AGAIN."
200 GOTO 100
210 DISP
220 IF CERT>LUCKYNUM-50 AND CERT<LUCKYNUM+50 THEN 410
230 IF CERT>LUCKYNUM-400 AND CERT<LUCKYNUM+400 THEN 330
240 IF CERT>LUCKYNUM-900 AND CERT<LUCKYNUM+900 THEN 300
250 IF CERT>LUCKYNUM-2000 AND CERT<LUCKYNUM+2000 THEN 280
260 DISP "SORRY, NO LUCK THIS TIME, BUT PLEASE TRY AGAIN."
270 GOTO 100
280 DISP "YOU'RE A $100.00 WINNER. CONGRATULATIONS!"
290 GOTO 100
300 DISP "OUR AGENT WILL CALL ON YOU TOMORROW TO PRESENT YOU WITH A CHECK FOR $1
0,000.00."
310 DISP "WE ARE HAPPY TO GIVE YOU THIS GOOD NEWS."
320 GOTO 100
330 DISP "I AM MOST PLEASED TO CONFIRM MY TELEPHONE CALL GIVING YOU THE NEWS THA
T"
340 DISP "YOU HAVE JUST WON $250,000.00! I LOOK FORWARD TO MEETING YOU TOMORROW
AND"
350 DISP "PRESENTING YOU WITH OUR CHECK IN THAT AMOUNT."
360 DISP
370 DISP "(SIGNED: JOHN D. GOLD"
380 DISP "PRESIDENT"
390 DISP "SWEEPSTAKES INTERNATIONAL)"
400 GOTO 100
410 DISP "I'M HAPPY TO DO THIS FAVOR FOR MY OLD FRIEND JACK GOLD AND EXCITED TO"

420 DISP "BE A PART OF YOUR GOOD FORTUNE. AS JACK TOLD YOU OVER THE PHONE, YOU'V
E"
430 DISP "WON THE JACKPOT - $1,000,000.00 - IN THE HP-86/87 SWEEPSTAKES. AS SOON
 AS"
440 DISP "AIR FORCE ONE CAN GET ME THERE, I'LL PERSONALLY HAND YOU YOUR CHECK."
450 DISP "I'LL SEE YOU SHORTLY."
460 GOTO 100
470 END
```

Listing for "One-Hundred"

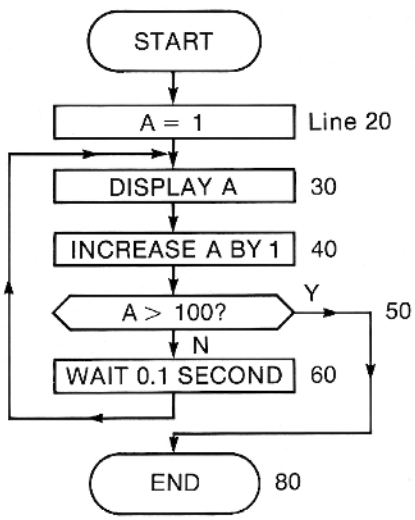
Reference: Page 13-3

```
10 ! ONE-HUNDRED
20 A=1
30 DISP A
40 A=A+1
50 IF A>100 THEN 80
60 WAIT 100
70 GOTO 30
80 END
```

Flowchart for "One-Hundred"

Reference: Page 13-3

A = Positive integer



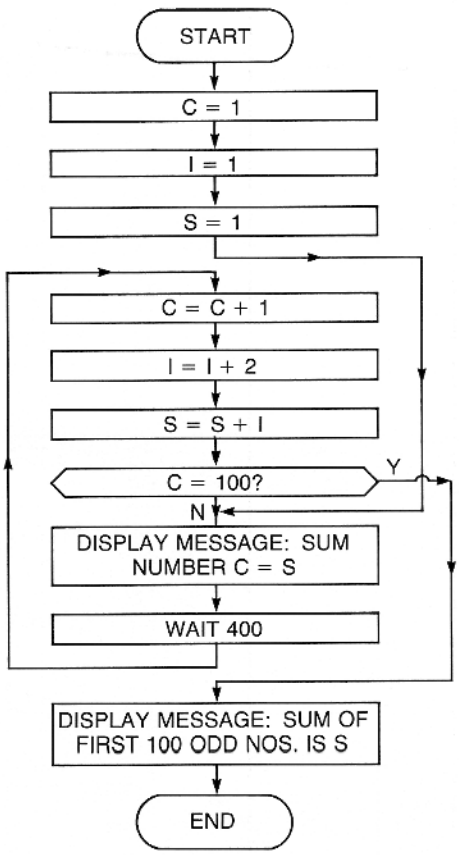
Listing for "100 Odd Sums"

Reference: Page 13-11

```
10 ! 100 ODD SUMS
20 C=1
30 I=1
40 S=1
50 GOTO 100
60 C=C+1
70 I=I+2
80 S=S+I
90 IF C=100 THEN 130
100 DISP "SUM NUMBER";C;"=";S;"."
110 WAIT 400
120 GOTO 60
130 DISP "THE SUM OF THE FIRST 100 ODD INTEGERS IS ";S;"."
140 END
```

Flowchart for "100 Odd Sums"

Reference: Page 13-11



C = Counter
I = Positive odd integer
S = Sum of odd integers

Output for "100 Odd Sums"

Reference: Page 13-11

```
SUM NUMBER 1 = 1 .
SUM NUMBER 2 = 4 .
SUM NUMBER 3 = 9 .
SUM NUMBER 4 = 16 .
SUM NUMBER 5 = 25 .
SUM NUMBER 6 = 36 .
SUM NUMBER 7 = 49 .
      :          :
SUM NUMBER 96 = 9216 .
SUM NUMBER 97 = 9409 .
SUM NUMBER 98 = 9604 .
SUM NUMBER 99 = 9801 .
THE SUM OF THE FIRST 100 ODD INTEGERS IS  10000 .
```

Listing and Partial Output for "Every Ten Times"

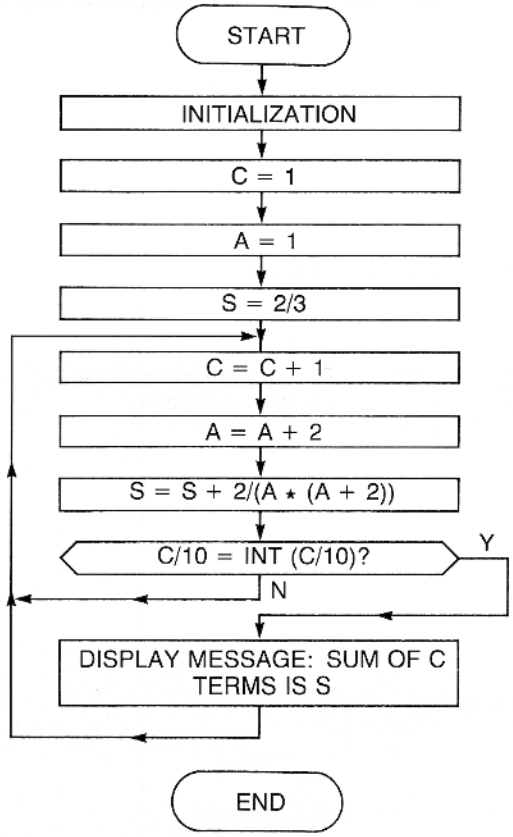
Reference: Page 13-12

```
10 ! EVERY TEN TIMES
20 CRT IS 1
30 NORMAL
40 CLEAR
50 C=1
60 A=1
70 S=2/3
80 C=C+1
90 A=A+2
100 S=S+2/(A*(A+2))
110 IF C/10=INT (C/10) THEN 130
120 GOTO 80
130 DISP "SUM OF";C;"TERMS IS"
140 DISP S
150 GOTO 80
160 END
```

```
SUM OF 10 TERMS IS
 .95238095238
SUM OF 20 TERMS IS
 .975609756097
SUM OF 30 TERMS IS
 .983606557376
SUM OF 40 TERMS IS
 .987654320986
SUM OF 50 TERMS IS
 .9900990099
SUM OF 60 TERMS IS
 .991735537189
```

Flowchart for "Every Ten Times"

Reference: Page 13-12



C = Counter
A = Denominator factor
S = Sum of terms

Table of Loop Counter Values vs. Loop Variable Values for "MONEY"

Reference: Page 14-7

												General Expressions
Loop Counter S	1	2	3	4	5	6	7	8	9	10	...	S
Square Number S	1	2	3	4	5	6	7	8	9	10	...	S
Number of Dollars D	1	2	4	8	16	32	64	128	256	512	...	D = 2 ^ (S − 1)

Listing and Output for "Roots"

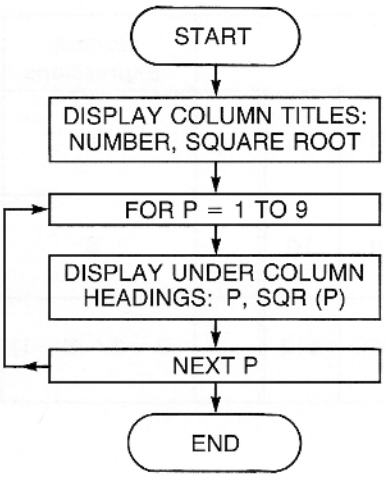
Reference: Page 15-4

```
25 ! ROOTS
50 DISP "NUMBER  SQUARE ROOT"
75 FOR P=1 TO 9
100 DISP P;"    ";SQR (P)
125 NEXT P
150 END
```

```
NUMBER  SQUARE ROOT
 1      1
 2      1.41421356237
 3      1.73205080757
 4      2
 5      2.2360679775
 6      2.44948974278
 7      2.64575131106
 8      2.82842712475
 9      3
```

Flowchart for "Roots"

Reference: Page 15-4



P = For—Next
loop counter

Listing and Output for "Sine"

Reference: Page 15-5

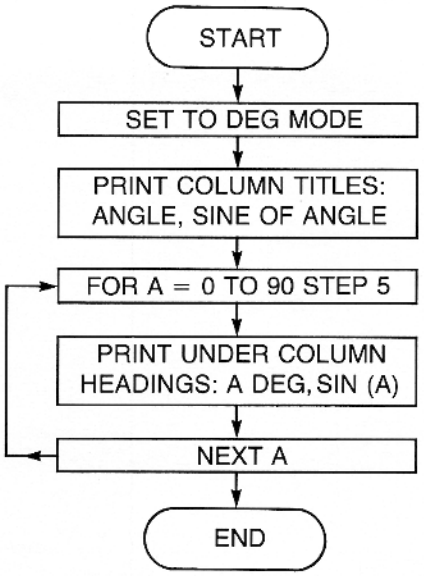
```
100 ! SINE
120 DEG
140 PRINT "  ANGLE      SINE OF ANGLE"
150 PRINT
160 FOR A=0 TO 90 STEP 5
180 PRINT A;"DEG    ";SIN (A)
200 NEXT A
220 END
```

```
ANGLE      SINE OF ANGLE

0 DEG     0
5 DEG     8.71557427477E-2
10 DEG     .173648177667
15 DEG     .258819045103
20 DEG     .342020143326
25 DEG     .422618261741
30 DEG     .5
35 DEG     .573576436351
40 DEG     .642787609687
45 DEG     .707106781187
50 DEG     .766044443119
55 DEG     .819152044289
60 DEG     .866025403784
65 DEG     .906307787037
70 DEG     .939692620786
75 DEG     .965925826289
80 DEG     .984807753012
85 DEG     .996194698092
90 DEG     1
```

Flowchart for "Sine"

Reference: Page 15-5



A = FOR—NEXT loop counter

Listing for "Common Log"

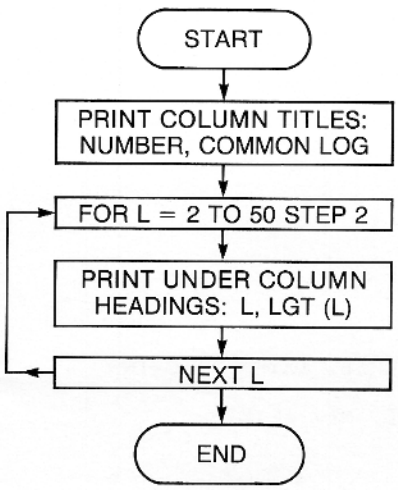
Reference: Page 15-5

```
10 ! COMMON LOG
20 PRINT "NUMBER    COMMON LOG"
30 PRINT
40 FOR L=2 TO 50 STEP 2
50 PRINT L;"    ";LGT (L)
60 NEXT L
70 END
```

Flowchart for "Common Log"

Reference: Page 15-5

L = FOR—NEXT loop counter
= Number



Output for "Common Log"

```
NUMBER    COMMON LOG

 2       .301029995664
 4       .602059991328
 6       .778151250384
 8       .903089986992
 10       1
 12       1.07918124605
 14       1.14612803568
 16       1.20411998266
 18       1.2552725051
 20       1.30102999566
 22       1.34242268082
 24       1.38021124171
 26       1.41497334797
 28       1.44715803134
 30       1.47712125472
 32       1.50514997832
 34       1.53147891704
 36       1.55630250077
 38       1.57978359662
 40       1.60205999133
 42       1.6232492904
 44       1.64345267649
 46       1.66275783168
 48       1.68124123738
 50       1.69897000434
```

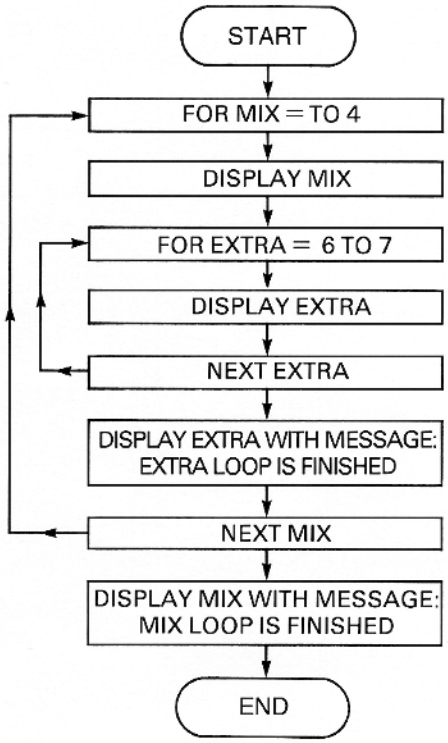
Listing for "Mix-up Unmixed"

Reference: Page 16-5

```
10 ! MIX UP
20  FOR MIX=1 TO 4
30  DISP "MIX=";MIX
40   FOR EXTRA=6 TO 7
50   DISP "EXTRA =";EXTRA
60   NEXT EXTRA
70  DISP "EXTRA LOOP IS FINISHED. EXTRA =";EXTRA
80  NEXT MIX
90 DISP "MIX LOOP IS FINISHED. MIX =";MIX
100 END
```

Flowchart for "Mix-up Unmixed"

Reference: Page 16-5



MIX = Outer loop counter
EXTRA = Inner loop counter

PRINTALL Output for "Mix-up Unmixed"

Reference: Page 16-5

```
MIX= 1
EXTRA = 6
EXTRA = 7
EXTRA LOOP IS FINISHED. EXTRA = 8
MIX= 2
EXTRA = 6
EXTRA = 7
EXTRA LOOP IS FINISHED. EXTRA = 8
MIX= 3
EXTRA = 6
EXTRA = 7
EXTRA LOOP IS FINISHED. EXTRA = 8
MIX= 4
EXTRA = 6
EXTRA = 7
EXTRA LOOP IS FINISHED. EXTRA = 8
MIX LOOP IS FINISHED. MIX = 5
```

PRINTALL Output for "Son of Roots"

Reference: Page 16-5

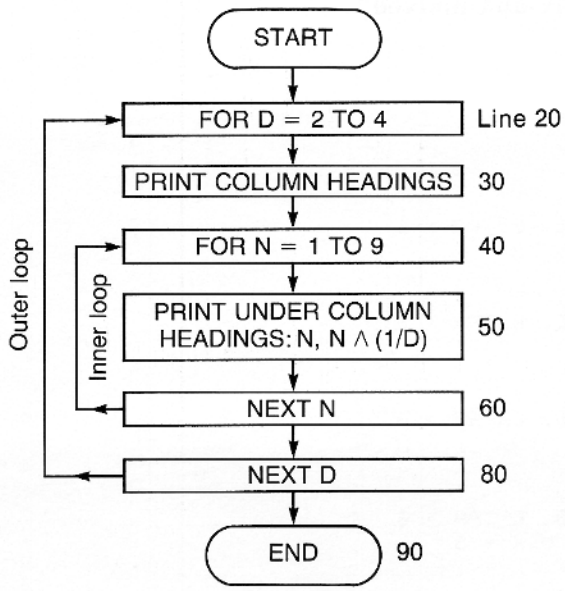
```
NUMBER    ROOT 2
 1        1
 2        1.41421356237
 3        1.73205080757
 4        2
 5        2.2360679775
 6        2.44948974278
 7        2.64575131106
 8        2.82842712475
 9        3

NUMBER    ROOT 3
 1        1
 2        1.25992104989
 3        1.44224957031
 4        1.58740105197
 5        1.70997594668
 6        1.81712059283
 7        1.91293118277
 8        2
 9        2.08008382305

NUMBER    ROOT 4
 1        1
 2        1.189207115
 3        1.31607401295
 4        1.41421356237
 5        1.49534878122
 6        1.56508458007
 7        1.6265765617
 8        1.68179283051
 9        1.73205080757
```

Flowchart for "Son of Roots"

Reference: Page 16-5



Listing for "Multiplication Test"

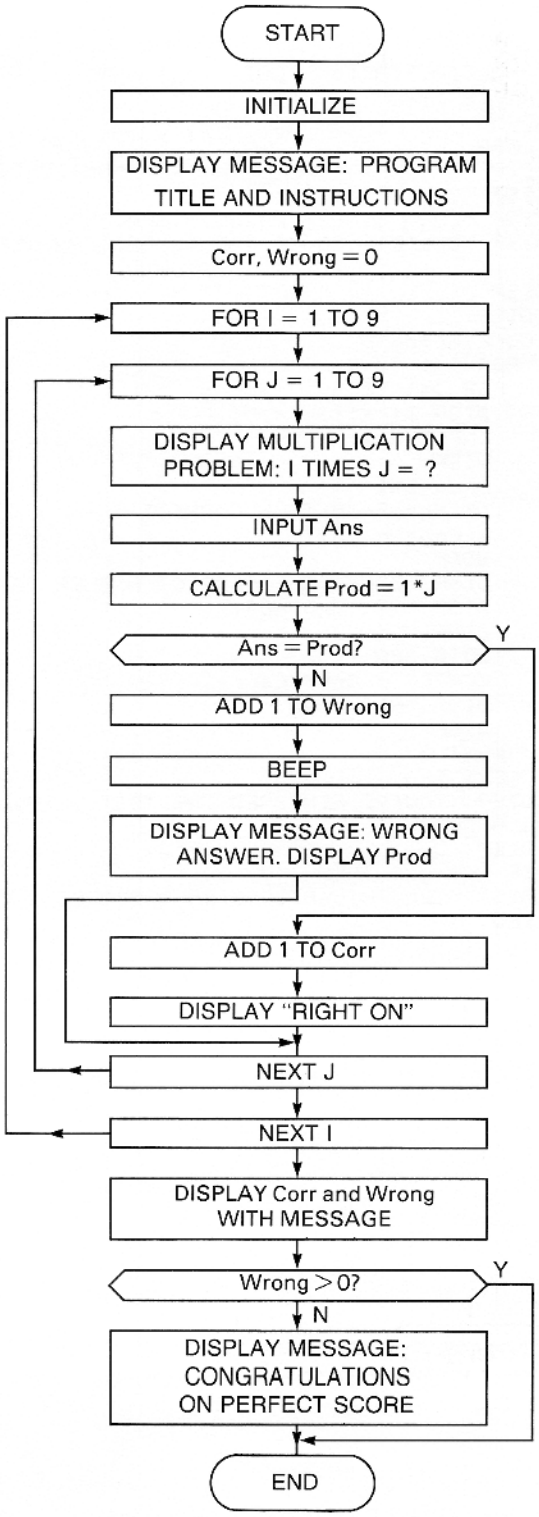
Reference: Page 16-6

```
10 ! MULTIPLICATION TEST
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 DISP "                          MULTIPLICATION TEST"
60 DISP
70 DISP "EACH TIME A PROBLEM APPEARS, SUCH AS '7 X 8 = ?,' TYPE A NUMBER YOU"
80 DISP "BELIEVE TO BE THE CORRECT ANSWER AND PRESS THE (END LINE) KEY. GOOD LUC
K!"
90 DISP
100 DISP
110 Corr,Wrong=0
120  FOR I=1 TO 9
130   FOR J=1 TO 9
140    DISP I;"X";J;"= ";
150    INPUT Ans
160    DISP
170    Prod=I*J
180    IF Ans=Prod THEN CORRECT
190    Wrong=Wrong+1
200    BEEP
210    DISP "OOPS! THE CORRECT ANSWER IS";Prod;"."
220    DISP "TRY ANOTHER."
230    DISP
240    GOTO 280
250    CORRECT: Corr=Corr+1
260    DISP "RIGHT ON!"
270    DISP
280   NEXT J
290  NEXT I
300 DISP "YOU'VE FINISHED THE TEST.. YOU HAD";Corr;"CORRECT ANSWERS AND";Wrong;"
WRONG ANSWERS."
310 DISP
320 IF Wrong>0 THEN 340
330 DISP "CONGRATULATIONS ON A PERFECT SCORE!"
340 END
```

Flowchart for "Multiplication Test"

Reference: Page 16-6

Corr = Number of correct answers
Wrong = Number of wrong answers
I = Outer loop counter
= First of 2 numbers of multiplication problem
J = Inner loop counter
= Second of 2 numbers of multiplication problem
Ans = User's answer to problem
Prod = Correct answer



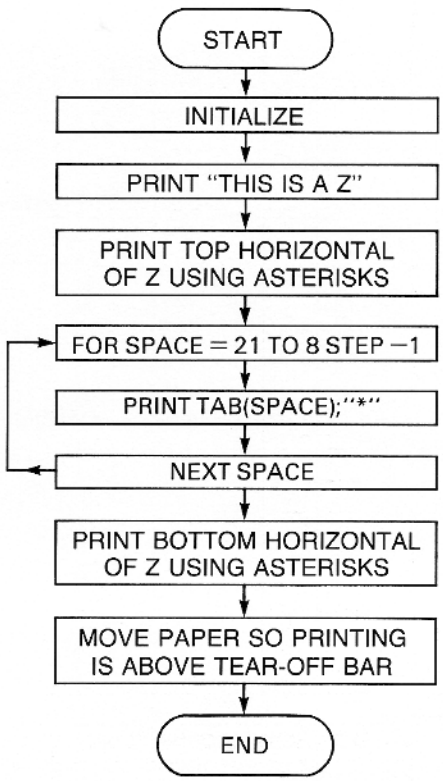
Listing for "Easy Z"

Reference: Page 17-6

```
10 ! EASY Z
20 PRINTER IS 701,80
30 PRINT TAB (9);"THIS IS A Z"
40 PRINT
50 PRINT TAB (7);"****************"
60 FOR SPACE=21 TO 8 STEP -1
70 PRINT TAB (SPACE);"*"
80 NEXT SPACE
90 PRINT TAB (7);"****************"
100 FOR LINE=1 TO 31
110 PRINT
120 NEXT LINE
130 END
```

Flowchart for "Easy Z"

Reference: Page 17-6



SPACE = Loop counter
LINE = LINE FEEDS

Listing for "Center"

Reference: Page 17-7

```
10 ! CENTER
20 PRINTER IS 701,80
30 PRINT
40 PRINT TAB (30);"HEWLETT-PACKARD"
50 PRINT TAB (27);"GETTING DOWN TO BASIC"
60 PRINT TAB (25);"CONTINUATION OF CHAPTER 3"
70 FOR LINE=1 TO 31
80 PRINT
90 NEXT LINE
100 END
```

Listing and Output for "Biggest and Smallest"

Reference: Page 17-9

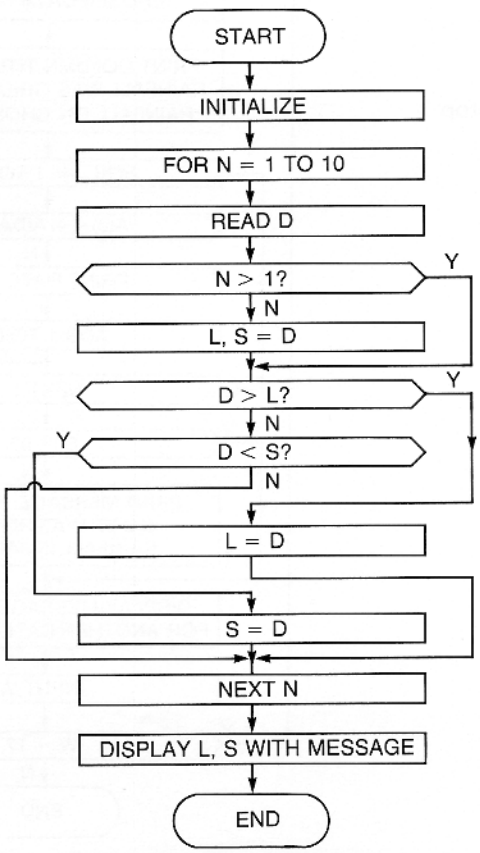
```
10 ! BIGGEST AND SMALLEST
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50  FOR N=1 TO 10
60   READ D
70   IF N>1 THEN 90
80   L,S=D
90   IF D>L THEN 120
100  IF D<S THEN 140
110  GOTO 150
120  L=D
130  GOTO 150
140  S=D
150 NEXT N
160 DISP "LARGEST NUMBER:";L
170 DISP "SMALLEST NUMBER:";S
180 DATA 5,14,362,43,201,-61,77,-843,2,-471
190 END
```

```
LARGEST NUMBER: 362
SMALLEST NUMBER:-843
```

Flowchart for "Biggest and Smallest"

Reference: Page 17-9

N = Loop counter
D = Number in DATA statement
L = Largest of numbers read from DATA list
S = Smallest of numbers read from DATA list



Flowchart for "Wet"

Reference: Page 18-11

A = Array holding each day's rainfall in January

N = First loop counter
= Date in January

R = Rainfall for January N^{th}

C = Counter. Counts number of days rainfall is greater than rainfall on user chosen date

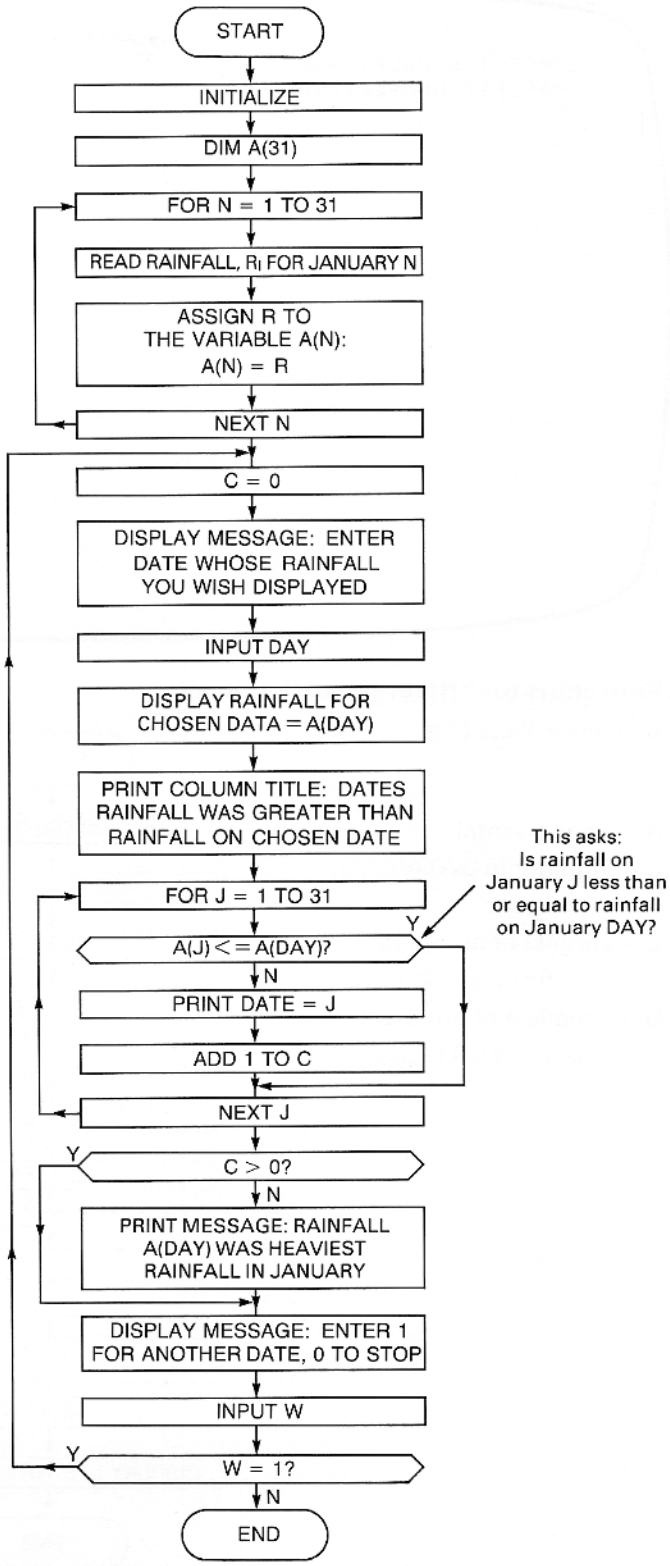
DAY = User chosen date in January whose rainfall user wants displayed

J = Second loop counter
= Date in January

W = Response to "continue or stop" question

W = 1: Continue

W = 0: Stop



Listing for "Wet"

Reference: Page 18-11

```
10 ! WET
20 OPTION BASE 1
30 CRT IS 1,80
40 PRINTER IS 701,80
50 NORMAL
60 CLEAR
70 DIM A(31)
80 FOR N=1 TO 31
90 READ R
100 A(N)=R
110 NEXT N
120 C=0
130 DISP "FOR WHICH DATE IN JANUARY, 1981 DO YOU WISH THE RAINFALL AMOUNT"
140 DISP "(ONE OR TWO DIGITS, PLEASE)";
150 INPUT DAY
160 DISP
170 DISP "THE RAINFALL IN DRENCH, OREGON FOR JANUARY";DAY;", 1981 WAS";A(DAY);"I
N."
180 WAIT 1000
190 DISP
200 DATA 3,7,1,19,37,6,1,5,2,4,7,9,8,3,0,2,6,4,3,6,10,12,13,11,14,16,14,13,8,7,4

210 PRINT TAB (8);"JANUARY 1981 DATES"
220 PRINT TAB (8);"RAINFALL WAS OVER"
230 PRINT TAB (13);A(DAY);"IN."
240 PRINT
250  FOR J=1 TO 31
260  IF A(J)<= A(DAY) THEN 290
270  PRINT TAB (14);J
280  C=C+1
290  NEXT J
300 IF C>0 THEN 330
310 PRINT TAB (2);A(DAY);"IN. WAS HEAVIEST RAINFALL"
320 PRINT TAB (7);"DURING JANUARY, 1981."
330 PRINT
340 DISP "IF YOU WANT ANOTHER DATE, ENTER 1. IF YOU WANT TO STOP, ENTER 0."
350 INPUT W
360 PRINT
370 DISP
380 IF W=1 THEN 120
390 END
```

```
FOR WHICH DATE IN JANUARY, 1981 DO YOU WISH THE RAINFALL AMOUNT
(ONE OR TWO DIGITS, PLEASE)?
28

THE RAINFALL IN DRENCH, OREGON FOR JANUARY 28 , 1981 WAS 13 IN.
```

```
JANUARY 1981 DATES
RAINFALL WAS OVER
      13 IN.

        4
        5
        25
        26
        27
```

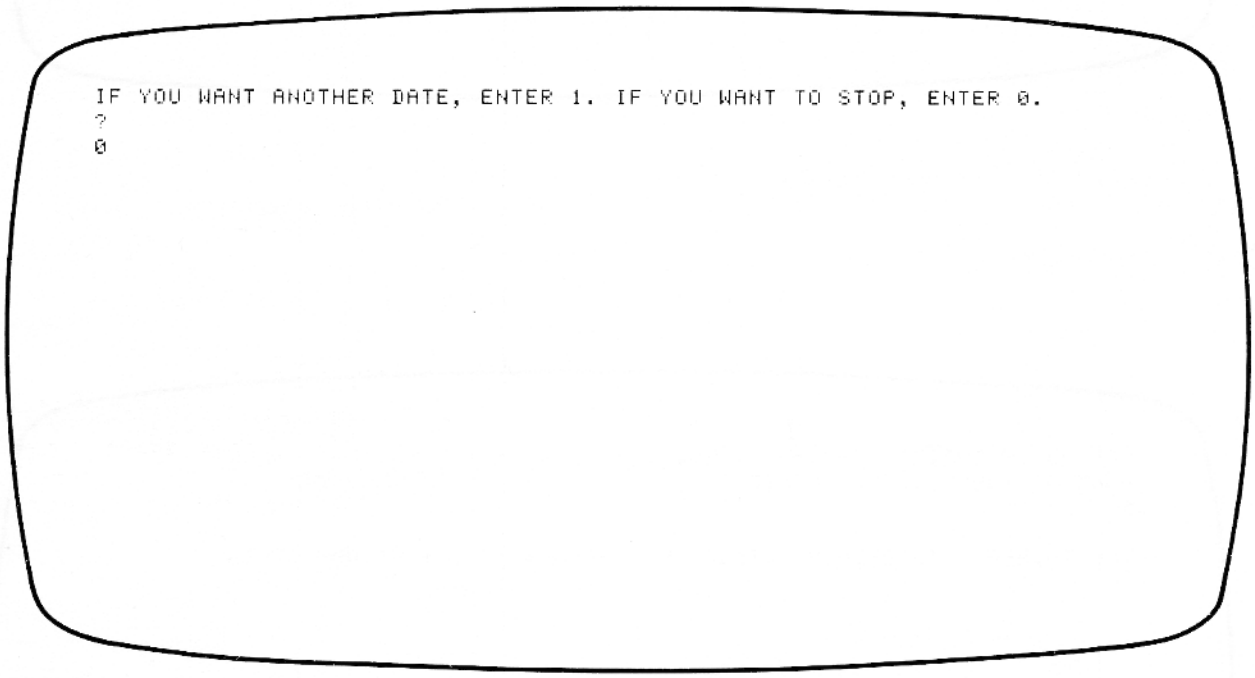
```
IF YOU WANT ANOTHER DATE, ENTER 1. IF YOU WANT TO STOP, ENTER 0.
?
1
```

```
FOR WHICH DATE IN JANUARY, 1981 DO YOU WISH THE RAINFALL AMOUNT
(ONE OR TWO DIGITS, PLEASE)?
5

THE RAINFALL IN DRENCH, OREGON FOR JANUARY 5 , 1981 WAS 37 IN.
```

```
     JANUARY 1981 DATES
     RAINFALL WAS OVER
           37 IN.

37 IN. WAS HEAVIEST RAINFALL
    DURING JANUARY, 1981.
```



Use of TRACE ALL Command on "Sort." The listing shows how statements 30 and 40 were changed to remarks.

Reference: Page 18-17

```
10 ! SORT
20 OPTION BASE 1
30 ! CRT IS 1,80
40 ! NORMAL
50 CLEAR
60 DIM N(3)
70 FOR J=1 TO 3
80 DISP "ENTER ANY NUMBER."
90 INPUT N(J)
100 NEXT J
110 FOR K=1 TO 2
120 FOR L=1 TO 2
130 IF N(L)>N(L+1) THEN 170
140 T=N(L)
150 N(L)=N(L+1)
160 N(L+1)=T
170 NEXT L
180 NEXT K
190 DISP
200 FOR D=1 TO 3
210 DISP N(D)
220 NEXT D
230 END
```

The CRT IS 701 and TRACE ALL commands were executed, then (RUN) was pressed. If PRINTALL had been executed instead of CRT IS 701, the number entries, -5, 0, and 7 would have been printed also.

```
 Trace line 10 to 20
 Trace line 20 to 30
 Trace line 30 to 40
 Trace line 40 to 50
 Trace line 50 to 60
 Trace line 60 to 70
 Trace line 70 J=1
 Trace line 70 to 80
ENTER ANY NUMBER.
 Trace line 80 to 90
?  <-- -5 entered here
 Trace line 90 N(1)=-5
 Trace line 90 to 100
 Trace line 100 J=2
 Trace line 100 to 80
ENTER ANY NUMBER.
 Trace line 80 to 90
?  <-- 0 entered here
 Trace line 90 N(2)=0
 Trace line 90 to 100
 Trace line 100 J=3
 Trace line 100 to 80
ENTER ANY NUMBER.
 Trace line 80 to 90
```

```
?                                                    7 entered here
Trace line 90 N(3)=7
Trace line 90 to 100
Trace line 100 J=4
Trace line 100 to 110
Trace line 110 K=1
Trace line 110 to 120
Trace line 120 L=1
Trace line 120 to 130
Trace line 130 to 140
Trace line 140 T=-5
Trace line 140 to 150
Trace line 150 N(1)=0
Trace line 150 to 160
Trace line 160 N(2)=-5
Trace line 160 to 170
Trace line 170 L=2
Trace line 170 to 130
Trace line 130 to 140
Trace line 140 T=-5
Trace line 140 to 150
Trace line 150 N(2)=7
Trace line 150 to 160
Trace line 160 N(3)=-5
Trace line 160 to 170
Trace line 170 L=3
Trace line 170 to 180
Trace line 180 K=2
Trace line 180 to 120
Trace line 120 L=1
Trace line 120 to 130
Trace line 130 to 140
Trace line 140 T=0
Trace line 140 to 150
Trace line 150 N(1)=7
Trace line 150 to 160
Trace line 160 N(2)=0
Trace line 160 to 170
Trace line 170 L=2
Trace line 170 to 130
Trace line 130 to 170
Trace line 170 L=3
Trace line 170 to 180
Trace line 180 K=3
Trace line 180 to 190
                                                     Line 190 DISP executed here
Trace line 190 to 200
Trace line 200 D=1
Trace line 200 to 210
7                                                    Output
Trace line 210 to 220
Trace line 220 D=2
Trace line 220 to 210
0                                                    Output
Trace line 210 to 220
Trace line 220 D=3
Trace line 220 to 210
5                                                    Output
Trace line 210 to 220
Trace line 220 D=4
Trace line 220 to 230
```

Use of TRACE ALL statement in "Sort" to trace element swapping statements only. Statements 135 and 165 were added. To produce the trace, the only command executed was RUN.

Reference: Page 18-17

```
10 ! SORT
20 OPTION BASE 1
30 CRT IS 1,80
40 NORMAL
50 CLEAR
60 DIM N(3)
70 FOR J=1 TO 3
80 DISP "ENTER ANY NUMBER."
90 INPUT N(J)
100 NEXT J
110 FOR K=1 TO 2
120 FOR L=1 TO 2
130 IF N(L)>N(L+1) THEN 170
135 TRACE ALL
140 T=N(L)
150 N(L)=N(L+1)
160 N(L+1)=T
165 NORMAL
170 NEXT L
180 NEXT K
190 DISP
200 FOR D=1 TO 3
210 DISP N(D)
220 NEXT D
230 END
```

```
Trace line 100 to 140
Trace line 140 T=-5
Trace line 140 to 150
Trace line 150 N(1)=0
Trace line 150 to 160
Trace line 160 N(2)=-5
Trace line 160 to 165
Trace line 170 to 140
Trace line 140 T=-5
Trace line 140 to 150
Trace line 150 N(2)=7
Trace line 150 to 160
Trace line 160 N(3)=-5
Trace line 160 to 165
Trace line 180 to 140
Trace line 140 T=0
Trace line 140 to 150
Trace line 150 N(1)=7
Trace line 150 to 160
Trace line 160 N(2)=0
Trace line 160 to 165
```

Listing for Augmented "Wet"

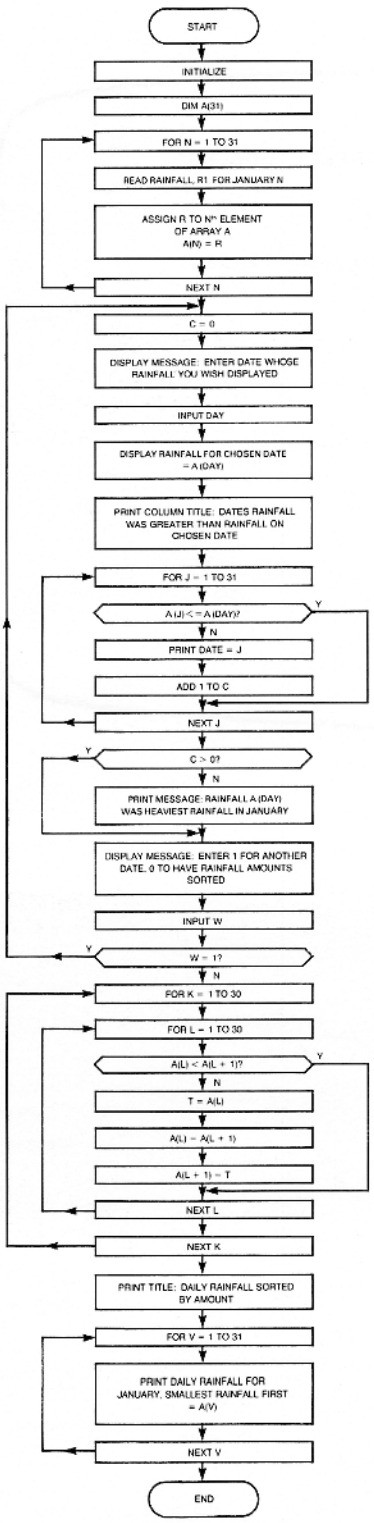
Reference: Page 18-18

```
10 ! WET-AUGMENTED
20 OPTION BASE 1
30 CRT IS 1,80
40 PRINTER IS 701,80
50 NORMAL
60 CLEAR
70 DIM A(31)
80    FOR N=1 TO 31
90    READ R
100   A(N)=R
110   NEXT N
120 C=0
130 DISP "FOR WHICH DATE IN JANUARY, 1981 DO YOU WISH THE RAINFALL AMOUNT"
140 DISP "(ONE OR TWO DIGITS, PLEASE)";
150 INPUT DAY
160 DISP
170 DISP "THE RAINFALL IN DRENCH, OREGON FOR JANUARY";DAY;", 1981 WAS";A(DAY);"I
N."
180 WAIT 3000
190 DISP
200 DATA 3,7,1,19,37,6,1,5,2,4,7,9,8,3,0,2,6,4,3,6,10,12,13,11,14,16,14,13,8,7,4

210 PRINT TAB (8);"JANUARY 1981 DATES"
220 PRINT TAB (8);"RAINFALL WAS OVER"
230 PRINT TAB (13);A(DAY);"IN."
240 PRINT
250    FOR J=1 TO 31
260    IF A(J)<= A(DAY) THEN 290
270    PRINT TAB (14);J
280    C=C+1
290    NEXT J
300 IF C>0 THEN 330
310 PRINT TAB (2);A(DAY);"IN. WAS HEAVIEST RAINFALL"
320 PRINT TAB (7);"DURING JANUARY, 1981."
330 PRINT
340 DISP "IF YOU WANT ANOTHER DATE, ENTER 1. IF YOU WANT TO HAVE RAINFALL AMOUNT
S"
350 DISP "SORTED, ENTER 0."
360 INPUT W
370 PRINT
380 DISP
390 IF W=1 THEN 120
400 CLEAR
410 DISP "Please wait about 20 seconds for the data to be sorted."
420    FOR K=1 TO 30
430       FOR L=1 TO 30
440       IF A(L)<A(L+1) THEN 480
450       T=A(L)
460       A(L)=A(L+1)
470       A(L+1)=T
480       NEXT L
490    NEXT K
495 CLEAR
500 PRINT "JANUARY 1981 DAILY RAINFALL FOR DRENCH, OREGON"
510 PRINT TAB (5);"SORTED BY AMOUNT."
520 PRINT
530    FOR V=1 TO 31
540    PRINT TAB (13);A(V)
550    NEXT V
560 END
```

Flowchart for Augmented "Wet"

Reference: Page 18-18



A = Array holding each day's rainfall in January

N = First loop counter
= Date in January

R = Rainfall for January Nth

C = Counter. Counts number of days rainfall is greater than rainfall on user chosen date

D = User chosen date in January whose rainfall user wants displayed

J = Second loop counter
= Date in January

W = Response to "another date or sort" question

W = 1: Date

W = 0:

K = Third loop counter. Used in sort routine

L = Fourth loop counter. Used in sort routine
= Date in January

T = Used for temporary assignment in sort routine

V = Fifth loop counter

PRINTALL Output for Augmented "Wet"

Reference: Page 18-18

```
FOR WHICH DATE IN JANUARY, 1981 DO YOU WISH THE RAINFALL AMOUNT
(ONE OR TWO DIGITS, PLEASE)?
22

THE RAINFALL IN DRENCH, OREGON FOR JANUARY 22 , 1981 WAS 12 IN.
```

```
JANUARY 1981 DATES
RAINFALL WAS OVER
      12 IN.

        4
        5
       23
       25
       26
       27
       28
```

```
IF YOU WANT ANOTHER DATE, ENTER 1. IF YOU WANT TO HAVE RAINFALL AMOUNTS
SORTED, ENTER 0.
?


0
```

Please wait about 20 seconds for the data to be sorted.

JANUARY 1981 DAILY RAINFALL FOR DRENCH, OREGON
SORTED BY AMOUNT.

0
1
1
2
2
3
3
3
4
4
4
5
6
6
6
7
7
7
8
8
9
10
11
12
13
13
14
14
16
19
37

Listing and Typical Output for "Rand 1" run three times
Reference: Page 19-4

```
10 ! RAND 1
20 RANDOMIZE
30 FOR N=1 TO 5
40 DISP INT (13*RND +1)
50 NEXT N
60 DISP
70 DISP
80 END
```

```
 12
 10
 13
 2
 5

 1
 12
 9
 3
 12

 2
 10
 8
 2
 1
```

Listing and Typical Output for "Rand 2"

Reference: Page 19-4

```
10 ! RAND 2
20 RANDOMIZE
30 FOR N=1 TO 5
40 DISP INT (100*RND +1)
50 NEXT N
60 DISP
70 DISP
80 END
```

```
 60
 71
 12
 68
 84

 89
 29
 90
 1
 15

 93
 80
 23
 79
 64

 69
 86
 82
 46
 26
```

Listing and Typical Output for "Rand 3" run four times

Reference: Page 19-4

```
10 ! RAND 3
20 RANDOMIZE
30 FOR N=1 TO 8
40 DISP INT (31*RND -10)
50 NEXT N
60 DISP
70 DISP
80 END
```

```
 20
-10
 6
 2
 9
 1
 14
 1

 4
-4
 4
 19
 19
 20
-4
 4

-4
-4
-10
 15
 15
 7
-7
 13

-4
-2
-10
-2
 6
 5
 6
 9
```

Listing for "Rand 4"

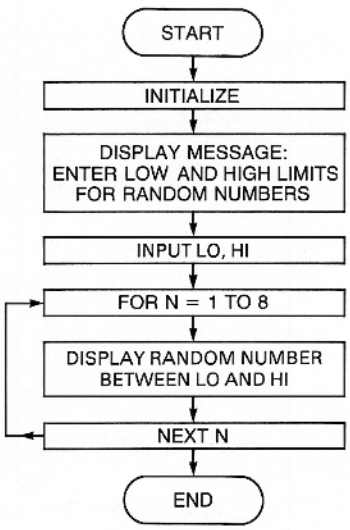
Reference: Page 19-4

```
10 ! RAND4
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 RANDOMIZE
60 DISP "WHAT IS THE RANGE OF RANDOM NUMBERS YOU WISH TO GENERATE?"
70 DISP "TYPE THE LOW END OF THE RANGE AND THE HIGH END OF THE RANGE,"
80 DISP "SEPARATED BY A COMMA, THEN PRESS (END LINE)."
90 INPUT LOW,HI
100 DISP
110 FOR N=1 TO 8
120 DISP INT ((HI+1-LOW)*RND +LOW);
130 NEXT N
140 DISP
150 DISP
160 END
```

Flowchart for "Rand 4"

Reference: Page 19-4

LO = Low end of random number range
HI = High end of random number range
N = Loop counter



Two Outputs for "Rand 4"

Reference: Page 19-4

```
WHAT IS THE RANGE OF RANDOM NUMBERS YOU WISH TO GENERATE?
TYPE THE LOW END OF THE RANGE AND THE HIGH END OF THE RANGE,
SEPARATED BY A COMMA, THEN PRESS (END LINE).
?
-5000,5000

-3096  1497 -3268 -529 -4545  2924  4255  71
```

```
WHAT IS THE RANGE OF RANDOM NUMBERS YOU WISH TO GENERATE?
TYPE THE LOW END OF THE RANGE AND THE HIGH END OF THE RANGE,
SEPARATED BY A COMMA, THEN PRESS (END LINE).
?
0,9

 8  5  3  2  9  1  0  4
```

Listing for "High Roller"

Reference: Page 19-4

```
10 ! "HIGH ROLLER"
20 CRT IS 1,80
30 PRINTER IS 701,80
40 NORMAL
50 CLEAR
60 RANDOMIZE
70 PRINT
80 PRINT TAB (26);"HIGH ROLLER"
90 PRINT
100 PRINT "THE OBJECT OF THIS GAME IS TO ROLL A HIGHER TOTAL THAN THE HP-86/87."

110 PRINT "WHEN YOU SEE 'YOUR ROLL,' PRESS (CONT) TO ROLL THE DICE. THE VALUE"
120 PRINT "ON EACH DIE WILL BE DISPLAYED, ALONG WITH THE TOTAL. THE HP-86/87 WIL
L"
130 PRINT "THEN ROLL AND DISPLAY A TOTAL. IT'S EASY TO STOP THE GAME. ALL YOU"
140 PRINT "DO IS STOP PRESSING (CONT)."
150   FOR P=1 TO 32
160   PRINT
170   NEXT P
180 S1,S2=0
190 DISP
200  ROLL: DISP "YOUR ROLL:"
210 PAUSE
220 A=INT (6*RND +1)
230 B=INT (6*RND +1)
240 DISP TAB (10);A;"+";B;"=";A+B
250 DISP
260 T1=A+B
270 A=INT (6*RND +1)
280 B=INT (6*RND +1)
290 DISP "HP-86/87'S ROLL:"
300 DISP TAB (10);A;"+";B;"=";A+B
310 DISP
320 T2=A+B
330 IF T1>T2 THEN PLAYER
340 IF T1<T2 THEN HP
350 DISP "TIE - ROLL AGAIN."
360 DISP
370 GOTO ROLL
380  PLAYER: S1=S1+1
390  DISP "YOU WIN!"
400  DISP
410 GOTO DISPLAY
420  HP: S2=S2+1
430 DISP "THE HP-86/87 WINS."
440  DISPLAY: DISP
450  DISP "WIN TOTAL:"
460  DISP
470  DISP TAB (10);"YOU HAVE";S1;"WINS."
480  DISP TAB (10);"THE HP-86/87 HAS";S2;"WINS."
490  DISP
500 GOTO ROLL
510 END
```

Flowchart for "High Roller"

Reference: Page 19-4

S1 = Number of times you win

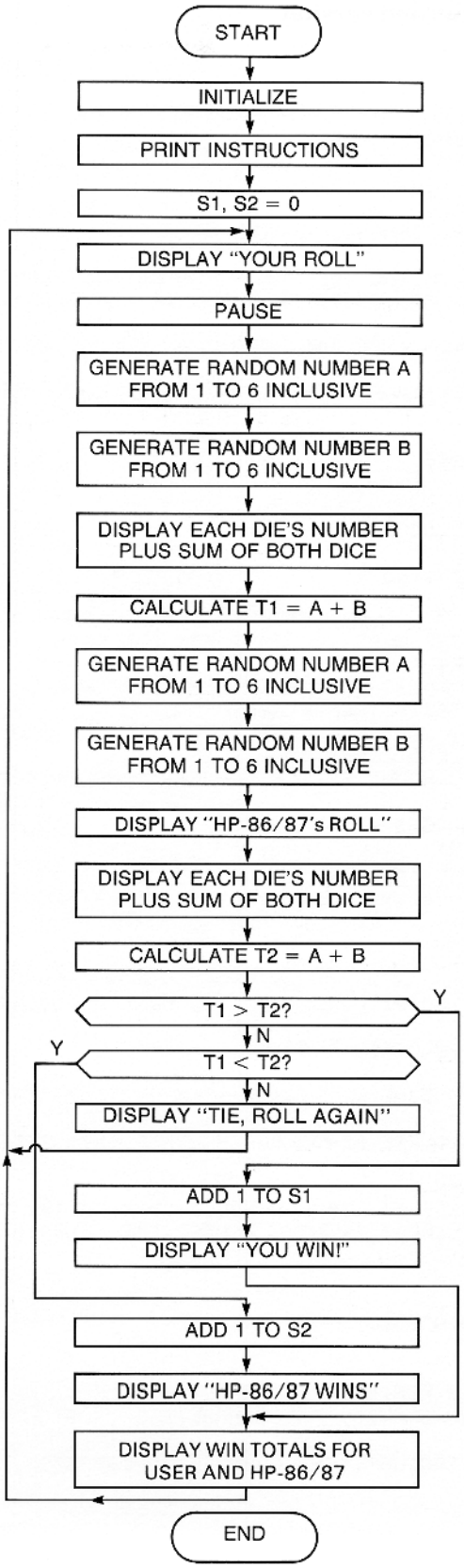
S2 = Number of times HP-86/87 wins

A = Throw of one die

B = Throw of second die

T1 = Total of user's 2 dice

T2 = Total of HP-86/87's 2 dice



Listing for "Spell," second version

Reference: Page 19-13

```
10 ! "SPELL" - SECOND VERSION
20 CLEAR
30 CRT IS 1,80
40 NORMAL
50 DISP "ON YOUR COMMAND, A WORD WILL FLASH ON THE SCREEN. IT WILL APPEAR IN THE
"
60 DISP "CENTER OF THE TOP ROW FOR A LITTLE OVER 0.1 SEC. WHEN IT DISAPPEARS,"
70 DISP "YOU WILL BE ASKED TO SPELL IT CORRECTLY. DON'T FORGET TO PRESS (END LIN
E)"
80 DISP "AFTER YOU TYPE THE WORD."
90 DISP
100 DISP "SHARPEN YOUR EYEBALLS AND GOOD LUCK! PRESS (CONT) WHEN READY."
110 DISP
120 N=0
130   WORDS: PAUSE
140 CLEAR
150 N=N+1
160 IF N=11 THEN DONE
170 READ TEST$
180 DISP TAB (31);TEST$
190 WAIT 100
200 CLEAR
210 DISP "TYPE THE WORD YOU JUST SAW AND PRESS (END LINE)."
220 INPUT ANSWER$
230 DISP
240 IF ANSWER$=TEST$ THEN RIGHT
250 DISP "SORRY, YOU'RE A LITTLE OFF. IF YOU WOULD LIKE TO TRY AGAIN, PRESS Y,"
260 DISP "THEN (END LINE)."
270 DISP "IF YOU WOULD RATHER SEE THE CORRECT SPELLING, THEN TRY THE NEXT WORD,"

280 DISP "PRESS (END LINE) ONLY."
290 INPUT ANSWER$
300 IF ANSWER$="" THEN 330
310 CLEAR
320 GOTO 180
330 DISP
340 DISP TEST$
350 DISP
360 DISP "WHEN YOU'RE READY FOR THE NEXT WORD, PRESS (CONT)."
370 GOTO WORDS
380   RIGHT: DISP "GOOD SHOW!"
390 WAIT 300
400 IF N=10 THEN DONE
410 DISP
420 DISP "WHEN YOU'RE READY FOR ANOTHER ONE, PRESS (CONT)."
430 GOTO WORDS
440   DONE: CLEAR
450 DISP "YOU MADE IT. IF YOU WOULD LIKE TO TRY THE TEST AGAIN, PRESS Y, THEN"
460 DISP "(END LINE). IF NOT, PRESS (END LINE) ONLY."
470 INPUT R$
480 CLEAR
490 DISP
500 IF R$="" THEN 530
510 RESTORE
520 GOTO 100
530 DISP "THANKS FOR YOUR EFFORT"
540 DATA "CARBOHYDRATE","PSYCHOANALYSIS","UBIQUITOUS","RHODODENDRON","RHYTHM","L
ANGUOROUS"
550 DATA "SAGITTARIUS","CHROMOSOME","GALAPAGOS","BEGUILEMENT"
560 END
```

Listing for "Spell," Third Version

Reference: Page 19-14

```
10 ! "SPELL" - THIRD VERSION
20 CLEAR
30 CRT IS 1,80
40 NORMAL
50 DISP "IF YOU'D LIKE INSTRUCTIONS, PRESS Y, THEN (END LINE). IF NOT PRESS"
60 DISP "(END LINE) ONLY."
70 INPUT ANSWER$
80 CLEAR
90 IF ANSWER$="" THEN 160
100 DISP "ON YOUR COMMAND, A WORD WILL FLASH ON THE SCREEN. IT WILL APPEAR IN TH
E"
110 DISP "CENTER OF THE TOP ROW FOR A LITTLE OVER 0.1 SEC. WHEN IT DISAPPEARS,"
120 DISP "YOU WILL BE ASKED TO SPELL IT CORRECTLY. DON'T FORGET TO PRESS (END LI
NE)"
130 DISP "AFTER YOU TYPE THE WORD."
140 DISP "FULL CREDIT WILL BE GIVEN FOR EACH CORRECT ANSWER."
150 DISP
160 DISP "SHARPEN YOUR EYEBALLS AND GOOD LUCK! PRESS (CONT) WHEN READY."
170 DISP
180 N,SCORE=0
190   WORDS: PAUSE
200 CLEAR
210 N=N+1
220 IF N=11 THEN DONE
230 READ TEST$
240 DISP TAB (31);TEST$
250 WAIT 100
260 CLEAR
270 DISP "TYPE THE WORD YOU JUST SAW AND PRESS (END LINE)."
280 INPUT ANSWER$
290 DISP
300 IF ANSWER$=TEST$ THEN RIGHT
310 DISP "SORRY, YOU'RE A LITTLE OFF. IF YOU WOULD LIKE TO TRY AGAIN, PRESS Y,"
320 DISP "THEN (END LINE)."
330 DISP "IF YOU WOULD RATHER SEE THE CORRECT SPELLING, THEN TRY THE NEXT WORD,"

340 DISP "PRESS (END LINE) ONLY."
350 INPUT ANSWER$
360 IF ANSWER$="" THEN 390
370 CLEAR
380 GOTO 240
390 DISP
400 DISP TEST$
410 DISP
420 DISP "WHEN YOU'RE READY FOR THE NEXT WORD, PRESS (CONT)."
430 GOTO WORDS
440   RIGHT: DISP "GOOD SHOW!"
450 SCORE=SCORE+10
460 WAIT 300
470 IF N=10 THEN DONE
480 DISP
490 DISP "WHEN YOU'RE READY FOR ANOTHER ONE, PRESS (CONT)."
500 GOTO WORDS
510   DONE: CLEAR
520 DISP "YOU MADE IT. YOUR SCORE IS";SCORE;"%. IF YOU WOULD LIKE TO TRY THE TES
T AGAIN"
530 DISP "PRESS Y, THEN (END LINE). IF NOT, PRESS (END LINE) ONLY."
540 INPUT R$
550 CLEAR
560 DISP
```

```
570 IF R$="" THEN 600
580 RESTORE
590 GOTO 160
600 DISP "THANKS FOR YOUR EFFORT"
610 DATA "CARBOHYDRATE","PSYCHOANALYSIS","UBIQUITOUS","RHODODENDRON","RHYTHM","L
ANGUOROUS"
620 DATA "SAGITTARIUS","CHROMOSOME","GALAPAGOS","BEGUILEMENT"
630 END
```

Listing for "Spell," Final Version

Reference: Page 19-14

```
10 ! "SPELL" - FINAL VERSION
20 CLEAR
30 CRT IS 1,80
40 NORMAL
50 DISP "IF YOU'D LIKE INSTRUCTIONS, PRESS Y, THEN (END LINE). IF NOT PRESS"
60 DISP "(END LINE) ONLY."
70 INPUT ANSWER$
80 CLEAR
90 IF ANSWER$="" THEN 160
100 DISP "ON YOUR COMMAND, A WORD WILL FLASH ON THE SCREEN. IT WILL APPEAR IN TH
E"
110 DISP "CENTER OF THE TOP ROW FOR A LITTLE OVER 0.1 SEC. WHEN IT DISAPPEARS,"
120 DISP "YOU WILL BE ASKED TO SPELL IT CORRECTLY AND PRESS (END LINE)."
130 DISP "TO GET FULL CREDIT, SPELL IT CORRECTLY ON YOUR FIRST TRY. IF YOU SPELL
 IT"
140 DISP "CORRECTLY ON A LATER TRY, YOU'LL ONLY GET HALF CREDIT."
150 DISP
160 DISP "SHARPEN YOUR EYEBALLS AND GOOD LUCK! PRESS (CONT) WHEN READY."
170 DISP
180 N,SCORE,POINTS=0
190   WORDS: PAUSE
200 CLEAR
210 N=N+1
220 IF N=11 THEN DONE
230 READ TEST$
240 POINTS=10
250 DISP TAB (31);TEST$
260 W=100
270 WAIT W
280 CLEAR
290 DISP "TYPE THE WORD YOU JUST SAW AND PRESS (END LINE)."
300 INPUT ANSWER$
310 DISP
320 IF ANSWER$=TEST$ THEN RIGHT
330 DISP "SORRY, YOU'RE A LITTLE OFF. IF YOU WOULD LIKE TO TRY AGAIN, PRESS Y,"
340 DISP "THEN (END LINE)."
350 DISP "IF YOU WOULD RATHER SEE THE CORRECT SPELLING, THEN TRY THE NEXT WORD,"

360 DISP "PRESS (END LINE) ONLY."
370 INPUT ANSWER$
380 IF ANSWER$="" THEN 470
390 POINTS=5
400 CLEAR
410 DISP TAB (31);TEST$
420 W=W+100
430 WAIT W
440 CLEAR
```

```
450 DISP "TYPE THAT WORD AGAIN AND PRESS (END LINE)."
460 GOTO 300
470 DISP
480 DISP TEST$
490 DISP
500 DISP "WHEN YOU'RE READY FOR THE NEXT WORD, PRESS (CONT)."
510 GOTO WORDS
520   RIGHT: DISP "GOOD SHOW!"
530 SCORE=SCORE+POINTS
540 WAIT 300
550 IF N=10 THEN DONE
560 DISP
570 DISP "WHEN YOU'RE READY FOR ANOTHER ONE, PRESS (CONT)."
580 GOTO WORDS
590   DONE: CLEAR
600 DISP "YOU MADE IT. YOUR SCORE IS";SCORE;"%. IF YOU WOULD LIKE TO TRY THE TES
T AGAIN"
610 DISP "PRESS Y, THEN (END LINE). IF NOT, PRESS (END LINE) ONLY."
620 INPUT R$
630 CLEAR
640 DISP
650 IF R$="" THEN 680
660 RESTORE
670 GOTO 160
680 DISP "THANKS FOR YOUR EFFORT"
690 DATA "CARBOHYDRATE","PSYCHOANALYSIS","UBIQUITOUS","RHODODENDRON","RHYTHM","L
ANGUOROUS"
700 DATA "SAGITTARIUS","CHROMOSOME","GALAPAGOS","BEGUILEMENT"
710 END
```

Listing for "THROW"

Reference: Page 19-16

```
10 ! TO THROW OR NOT TO THROW--"THROW"
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 DIM Y$[6],T$[6],S$[6],C$[18],F$[6]
60 Y$=" YOUR "
70 T$="TOTAL "
80 S$="SCORE:"
90 C$=" THE HP-86/87'S "
100 F$="FINAL "
110 RANDOMIZE
120 DISP "DO YOU WISH SOME INSTRUCTIONS (ENTER Y OR N)"
130 INPUT I$
140 IF I$="N" THEN 380
150 CLEAR
160 DISP TAB (19);"TO THROW OR NOT TO THROW"
170 DISP
180 DISP "YOU AND THE HP-86/87 ROLL 3 DICE. HIGH SCORE WINS. THE SCORING IS:"
190 DISP
200 DISP "         NO PAIR:      SUM OF DICE"
210 DISP "        ONE PAIR:      3 TIMES SUM OF DICE"
220 DISP "    3 OF A KIND:       18 TIMES SUM OF DICE"
230 DISP
240 DISP "IF YOU DO NOT ROLL A PAIR OR 3 OF A KIND, YOUR TURN ENDS AND THE HP-86
/87 ROLLS."
250 DISP "IF YOU ROLL A PAIR, YOU MAY END YOUR TURN OR TRY FOR 3 OF A KIND BY"
```

```
260 DISP "ROLLING THE ODD DIE."
270 DISP
280 DISP "THE HP-86/87'S TURN ALWAYS CONSISTS OF A SINGLE ROLL. IT NEVER TRIES T
O"
290 DISP "UPGRADE A PAIR INTO 3 OF A KIND."
300 DISP
310 DISP "FOR MORE RULES, PRESS (CONT)."
320 PAUSE
330 CLEAR
340 DISP
350 DISP "YOU AND THE HP-86/87 EACH HAVE 12 ROLLS. ROLLING ONE DIE OR THREE DICE
"
360 DISP "USES UP A ROLL. SO WHEN YOU FINISH YOUR 12TH ROLL, THE HP-87 COULD HAV
E"
370 DISP "SEVERAL ROLLS LEFT."
380 R1,R2,T1,T2,F=0
390 DISP
400 DISP "TO ROLL YOUR DICE, PRESS (CONT)."
410 PAUSE
420 CLEAR
430 DISP
440 A=INT (6*RND +1)
450 B=INT (6*RND +1)
460 C=INT (6*RND +1)
470 R1=R1+1
480 IF R1=12 THEN 500
490 GOTO 510
500 F=1
510 WAIT 500
520 DISP TAB (5);A;TAB (10);B;TAB (15);C
530 IF A=B AND B=C THEN 730
540 IF A#B AND B#C AND A#C THEN 750
550 IF F=1 THEN 770
560 DISP
570 DISP "IF YOU WISH TO TRY FOR 3 OF A KIND, ENTER Y. IF NOT, ENTER N."
580 INPUT A$
590 IF A$="Y" THEN 630
600 IF A$="N" THEN 770
610 DISP "I DIDN'T UNDERSTAND YOU. PLEASE TRY AGAIN."
620 GOTO 560
630 X=INT (6*RND +1)
640 IF A=B THEN 670
650 IF A=C THEN 690
660 IF B=C THEN 710
670 C=X
680 GOTO 470
690 B=X
700 GOTO 470
710 A=X
720 GOTO 470
730 T1=T1+18*(A+B+C)
740 GOTO 780
750 T1=T1+A+B+C
760 GOTO 780
770 T1=T1+3*(A+B+C)
780 DISP
790 IF F=1 THEN 820
800 DISP Y$;T$;S$;TAB (27);T1
810 GOTO 860
820 DISP Y$;F$;S$;TAB (27);T1
830 IF R2=11 THEN 860
840 DISP
850 DISP "YOU'VE USED UP YOUR 12 THROWS. SIT BACK AND WATCH THE HP-86/87 FINISH
ITS THROWS."
```

```
860 WAIT 1500
870 DISP
880 DISP "THE HP-86/87'S ROLL:"
890 A=INT (6*RND +1)
900 B=INT (6*RND +1)
910 C=INT (6*RND +1)
920 R2=R2+1
930 WAIT 500
940 DISP
950 DISP TAB (5);A;TAB (10);B;TAB (15);C
960 IF A=B AND B=C THEN 1000
970 IF A#B AND B#C AND A#C THEN 1020
980 T2=T2+3*(A+B+C)
990 GOTO 1030
1000 T2=T2+18*(A+B+C)
1010 GOTO 1030
1020 T2=T2+A+B+C
1030 IF R2=12 THEN 1150
1040 DISP
1050 DISP C$;T$;S$;TAB (27);T2
1060 IF F=1 THEN 1080
1070 GOTO 390
1080 DISP
1090 DISP Y$;F$;S$;TAB (27);T1
1100 DISP
1110 DISP TAB (6);"*    *    *    *    *"
1120 DISP
1130 WAIT 1500
1140 GOTO 860
1150 DISP
1160 DISP "THE GAME IS OVER."
1170 DISP
1180 DISP Y$;F$;S$;TAB (27);T1
1190 DISP C$;F$;S$;TAB (27);T2
1200 DISP
1210 DISP "IF YOU'D LIKE ANOTHER GAME, PRESS (CONT)."
1220 PAUSE
1230 GOTO 380
1240 END
```

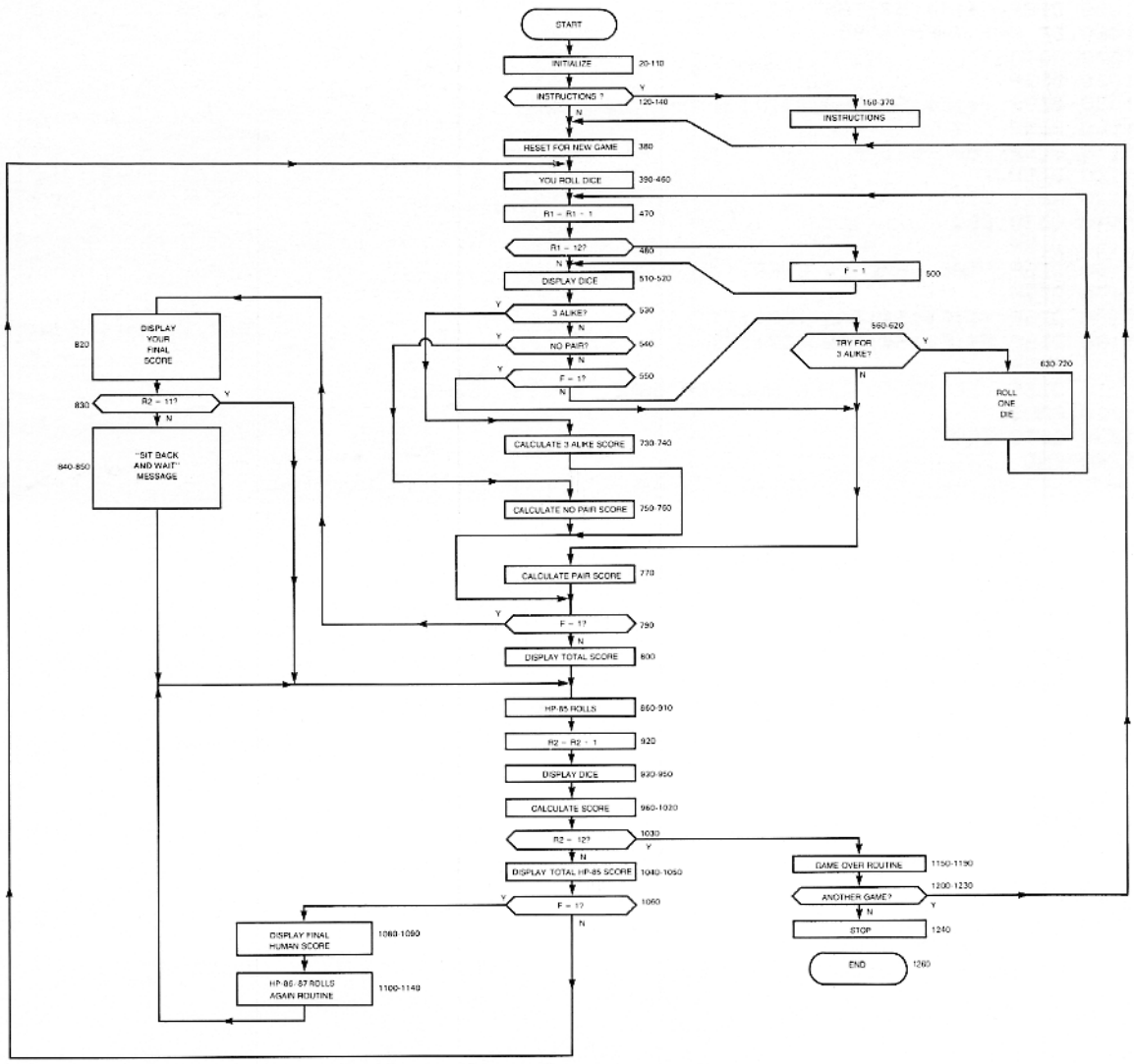
Flowchart for "THROW"

Reference: Page 19-16

R1 = Counter for
human's games

R2 = Counter for
HP-86/87's games

F = 1: Means
human has
rolled 12 times



Answer to ON...GOTO Question

Reference: Page 20-3

No, the INT may not be dropped from INT(5*RND+1) when the value of this expression is used in the computed GOTO in "Congratulations." The fourth time the modified "Congratulations" was run below, 5.82621700333 rounded to 6. Since there are only five line numbers in the ON...GOTO list, an error resulted. Remember, INT means the largest integer *less than* or equal to the number. When ON...GOTO rounds, the number is replaced by the *nearest* integer, which can be *greater than* as well as less than the number.

Note: ARG means "argument".

```
10 ! CONGRATULATIONS
20 R=0
30 N=5*RND +1
40 DISP N
50 ON N GOTO 60,80,100,120,140
60 DISP "YOU DO GOOD WORK."
70 GOTO 150
80 DISP "CORRECT"
90 GOTO 150
100 DISP "YOU'RE RIGHT."
110 GOTO 150
120 DISP "EXCELLENT"
130 GOTO 150
140 DISP "WELL DONE"
150 R=R+1
160 END
```

```
 1.3966332085
YOU DO GOOD WORK.
 3.0737107943
YOU'RE RIGHT.
 2.22298261741
CORRECT.
 5.82621700333
Error 11 on line 50 : ARG OUT OF RANGE
```

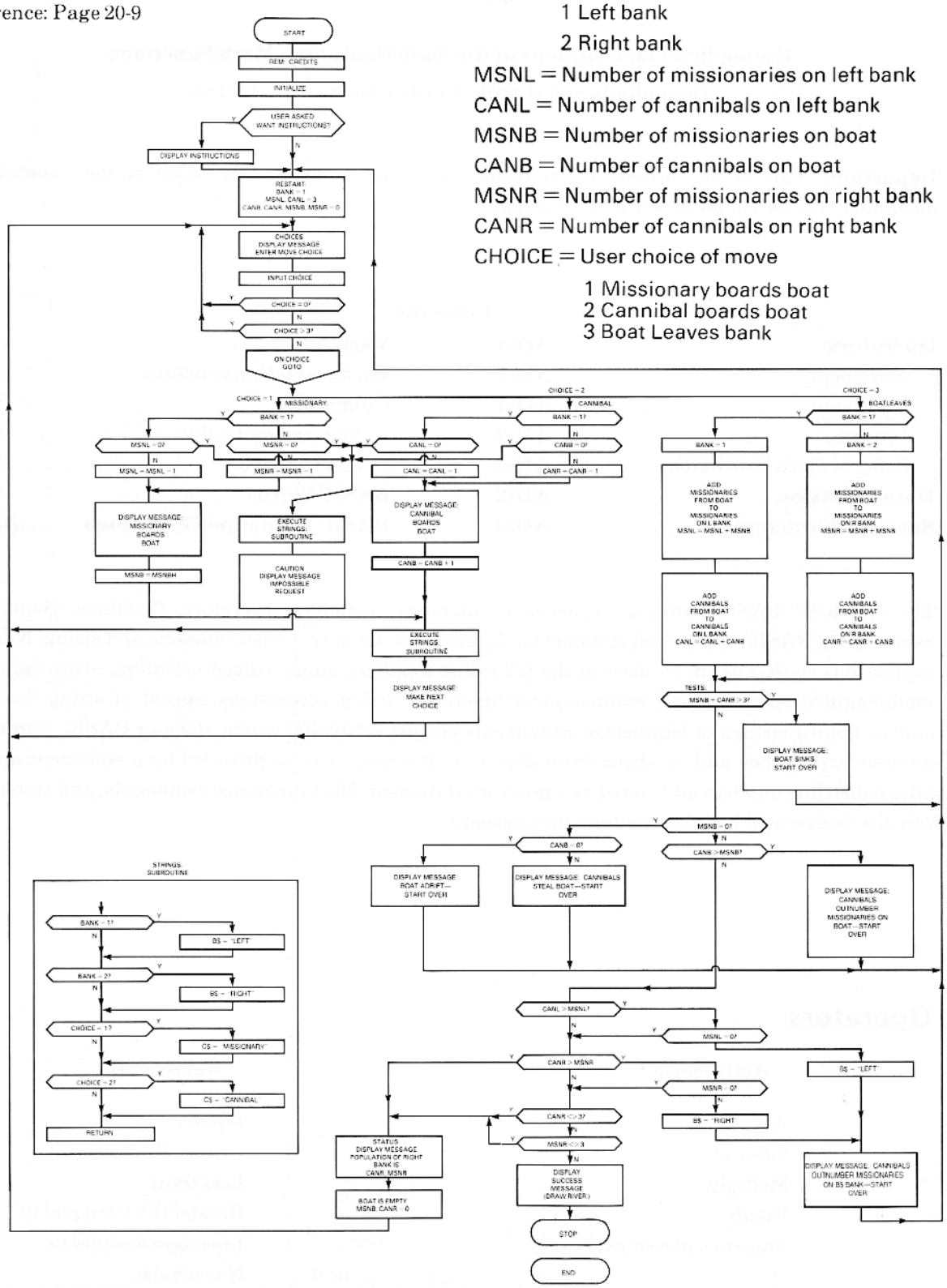
Listing for "MATH," or "Arithmetic Quiz 1," the first of the six stages of "MATH"
Reference: Page 20-8

```
10 ! ARITHMETIC QUIZ 1 - "MATH"
20 CRT IS 1,80
30 NORMAL
40 CLEAR
50 RANDOMIZE
60 DISP
70 DISP "TEST YOUR POWERS OF ADDITION!"
80 DISP
90 DISP "THE LARGEST AND SMALLEST NUMBERS WHICH WILL BE USED ARE:"
100 DISP TAB (3);"LARGEST:";TAB (13);"99";TAB (26);"SMALLEST:";TAB (36);"2"
110 DISP
120 DISP "TYPE YOUR ANSWER, THEN PRESS (END LINE)."
130 DISP "YOUR SCORE WILL BE SHOWN AFTER EVERY 10 ANSWERS."
140 DISP
150 RIGHT,TOTAL=0
160 A=INT (98*RND +2)
170 B=INT (98*RND +2)
180 DISP "WHAT IS";A;"PLUS";B;
190 INPUT NUMBER
200 DISP
210 IF NUMBER=A+B THEN 240
220 DISP A;"+";B;"=";A+B
230 GOTO 260
240 DISP "CORRECT!"
250 RIGHT=RIGHT+1
260 TOTAL=TOTAL+1
270 DISP
280 IF TOTAL=10 THEN 300
290 GOTO 160
300 DISP "YOUR SCORE FOR THE LAST 10 QUESTIONS IS";RIGHT*100/TOTAL;"%"
310 GOTO 140
320 END
```

Flowchart for "CAMIS"

Reference: Page 20-9

BANK = Flag: At which bank is boat moored
1 Left bank
2 Right bank
MSNL = Number of missionaries on left bank
CANL = Number of cannibals on left bank
MSNB = Number of missionaries on boat
CANB = Number of cannibals on boat
MSNR = Number of missionaries on right bank
CANR = Number of cannibals on right bank
CHOICE = User choice of move
1 Missionary boards boat
2 Cannibal boards boat
3 Boat Leaves bank



Abridged Dictionary of HP-86/87 BASIC Language

Giving Brief Descriptions of the Math Operators, Math Functions, Commands and BASIC Words Used in This Course

Important: This is an incomplete dictionary. Your *HP-86/87 Pocket Guide* is the "unabridged dictionary" for the HP-86/87's BASIC language.

Contents

The HP-86/87 BASIC language consists of numbers, variables, operators, functions, commands, expressions, BASIC words, and statements. Expressions are of two types, numeric and string. Numeric expressions consist of one or more of the following: numbers, single-valued variables, array variables, mathematical operators and mathematical functions. String expressions consist of string variables and/or quoted strings of characters. Statements consist of BASIC words alone or BASIC words plus numeric expressions and/or string expressions. A statement may be preceded by a statement number (also called line number) and stored as a program statement. Most functions, commands, and statements can also be executed individually from the keyboard.

Operators

Arithmetic

+	Add
-	Subtract
*	Multiply
/	Divide
^	Raise to a power (exponent)

Relational

=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<> or #	Not equal to

Logical

AND
OR

TRUTH TABLE

If A is	and if B is	then (A AND B) is	and (A OR B) is
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Order of Math Calculation

( ) Perform First
Functions
^
*, /
+, − Perform Last

Expressions are evaluated from left to right for operators at same level. Operations within parentheses are performed first. Nested parentheses are evaluated inward out.

Data Precision

Precision	Accuracy	Range	Max array size with standard memory and negligible program
REAL*	12 Digits	± 9.99999999999E ± 499	3500 numbers
INTEGER	Integer	− 99999 through 99999	9350 numbers

* Unless otherwise instructed, the HP-86/87 always handles data with REAL precision.

Special Characters

`!` Remark follows.

```
110 DISP C ! DISPLAY COST
```

`?` `INPUT` prompt. Entry of input item or items expected.

`" "` String boundary markers (string delimiters). Define beginning and end of quoted (literal) text.

```
120 PRINT "THROW";D;"DICE "
```

Variables

Simple numeric variables: `A1, B, Value, N2UV9, SCORE`

The name consists of up to 31 letters, digits, and underscores in combination. `REAL` precision (12 digit accuracy) is assumed unless an `INTEGER` type is declared.

One dimensional arrays

Arrays are lists of values. The name consists of up to 31 letters, digits, and underscores in combination. The name of an array may be the same as a simple variable used elsewhere in the program.

Subscripted variables: `A(4), ARRAY(8), N7V9(24)`

Each number in a one dimensional array is represented by a subscripted variable, whose name consists of the array name followed by a numerical subscript in parentheses.

The subscript gives the array position or element where the number is located. At power on, `OPTION BASE 0` is in force, which means the subscript for the first element is zero. To make the subscript for the first element one, an `OPTION BASE 1` statement must be executed before any statement involving an array name or subscripted variable.

The dimension of a one dimensional array is 11 for `OPTION BASE 0` or 10 for `OPTION BASE 1` unless it is dimensioned otherwise in a `DIM` statement. For example, `20 DIM A(100),C3(5)`.

String variables: `F$, Word$, M5St$`

The name consists of up to 31 letters, digits, and underscores in combination, followed by a dollar sign.

The dimension of a string variable is 18 characters unless it is dimensioned otherwise in a `DIM` statement using square brackets. For example: `20 DIM F3$[30],H$[10]`

Strings and string variables can be compared in `IF...THEN`, `IF...THEN...ELSE`, `IF...AND...THEN`, `IF...OR...THEN`, `IF...AND...THEN...ELSE`, and `IF...OR...THEN...ELSE` statements.

Terms and Conventions

These terms and conventions are used in the following list of BASIC words and commands.

DOT MATRIX	All items in dot matrix denote system commands or BASIC words which must appear exactly as shown.
[]	All items enclosed within square brackets are optional unless the brackets are in dot matrix.
statement or *line number*	An integer from 1 through 99999.
numeric expression	A logical combination of variables, constants, operators, and functions grouped within parentheses as necessary.
file name or *program name*	A program or file name can be a quoted string composed of at most 10 characters, including spaces. Any space, letter, number, or symbol may be used within the quotation marks except quotes or the null string. File names longer than 10 characters are truncated.

Commands

Non-Programmable

AUTO [*beginning statement number* [, *increment value*]]

Enables program statements to be numbered automatically by the system. If no parameters are specified, numbering begins with 10 and is incremented by 10. If one parameter is specified, numbering begins with that statment and is incremented by 10. AUTO numbering can be halted by executing NORMAL or any executable statement or command without a statement number.

```
AUTO
AUTO 100
AUTO 50,25
```

CONT [*statement number*]

Continues execution of program at specified statement, or where it was halted by PAUSE, without altering program conditions.

DELETE *first statement number* [, *last statement number*]

Deletes individual statement or inclusive range of statements.

```
DELETE 20
DELETE 110,150
```

INIT

Reserves system memory for all program numeric variables. If CONT is executed following INIT, program will start at the first statement.

LOAD "*program name*"

Loads from disc into memory a copy of the specified program. The program must have been previously stored with a STORE command. Computer memory is scratched before the program is loaded. The volume label, preceded by a period, may be included.

```
LOAD "CAMIS.BASIC"
```

REN[*first statement number*[, *increment value*]]

When parameters are not specified, renumbers all statements in the current program beginning with 10, in intervals of 10. The beginning line number and interval may be specified. If renumbering will cause statement numbers to be larger than 99999, REN automatically renumbers the entire program beginning with 1, and incrementing by 1.

```
REN
REN 500
REN 2010,2
```

RUN[*statement number*]

Starts execution of current program at first statement, or at specified statement.

```
RUN
RUN 100
```

SCRATCH

Deletes current program and all variables from computer memory.

STORE "*program name*"

Stores current program in memory into a program file on a disc. The volume label, preceded by a period, may be included.

```
STORE "SAVING"
```

UNSECURE "*file name*","*security code*", *security type*

Removes security from a file secured with the SECURE statement. The file name, security code and security type must match those specified in the SECURE command.

Programmable

The following system commands may also be used in statements as BASIC words.

CAT

Produces a catalog or list of information about user disc files.

INITIALIZE "*new volume label*","*msus*"

Prepares a disc for storage of programs or data files. Destroys all data on a previously recorded disc.

LIST [*beginning statement number* [, *ending statement number*]]

Lists on the current display all or part of the current program from lowest numbered statement to highest numbered statement. If no statement number is specified, listing begins with the lowest or the next statement number and continues for one screen. If one statement number is specified, listing begins with that statement and continues for one screen. If two statement numbers are specified, that program segment is listed. Following the listing of the final program statement, the remaining number of memory locations or bytes is displayed.

```
LIST
LIST 200
LIST 1050, 1200
```

MASS STORAGE IS [" , *volume label* " / " : *msus* "]

Sets the default mass storage (disc) device. Once set, the system automatically uses that device when a volume label or msus is not specified.

PLIST [*beginning statement number*[, *ending statement number*]]

Same as LIST except that the listing appears on the current printer and all the program or program segment will be listed at once, unless stopped by any key.

PRINT ALL

Produces a printed copy of all user activity. Any information appearing on the display is printed on the printer.

SECURE "*file name*" , "*security code*" , *security type*

Secures a program file against being listed, edited, stored, rewritten, or prevents the name of a program data file from being listed in the directory. The file name must exist in the directory before it can be secured. The security code consists of any two characters designated by the user. See the HP-86/87 operating manual for security type usage.

VOLUME [" : *msus* " or " , *old volume label* "] IS "*new volume label*"

Assigns a volume label to a disc or changes the volume label of a disc.

BASIC Words

Most BASIC words can also be executed as commands without statement numbers. The exceptions are DATA, DIM, FOR–NEXT, GOSUB–RETURN, GOTO, INPUT, ON ... GOTO, and READ.

BEEP

Produces a sound. The frequency is approximately 1200 Hz, and duration is 100 milliseconds.

```
140 BEEP
```

CLEAR

Clears the alphanumeric screen. It does not clear graphics screen.

```
190 CLEAR
```

CRT IS *output code number* [, *number of characters per line*]

Specifies the device that will display normal system messages and the items in a DISP statement display list. When the HP-86/87 is switched on the display device is the CRT screen.

Code Number	Device
1	CRT (display)
*701	Printer

```
220 CRT IS 701,80
```

DATA *data list*

Provides numeric constants and quoted or unquoted strings of characters from which the READ statement can obtain values for numeric and string variables. A comma within a quoted string is part of the data item. A comma outside of a quoted string separates two data items. Data is read from left to right. DATA statements may be located anywhere in a program and are considered a continuous list. DATA cannot be executed as a command from the keyboard.

```
150 DATA 3,76.5,"HENRY","1,525"
```

DEG

Sets degrees mode for results and arguments of trigonometric functions. There are 360 degrees in a circle. The HP-86/87 is set to radians mode when switched on.

```
25 DEG
```

DIM *dimension list*

Declares the maximum physical size for one dimensional REAL precision arrays and the maximum length for strings. Dimension list items are separated by commas. A string variable name must be followed by the length in brackets; an array name must be followed by the maximum subscript value in parentheses.

```
340 DIM L(2),N$[90],D0(5),C3$[6]
```

* Changes as printer address and select code change.

DISP [*display list*]

Causes items in the display list to be displayed on the CRT (or CRT IS device). The display list may contain variables, arrays, expressions, or TAB. Items in the display list must be separated by commas or semicolons. A comma advances an item to the next display zone, a semicolon suppresses the advance to give close spacing. DISP without a display list "displays" a blank line.

```
120 DISP
680 DISP "A =",A,"B =",B
955 DISP "*** WAIT A MINUTE ***"
370 DISP SIN(X)^3+5;G$
```

END

Terminates program execution.

```
820 END
```

FOR *loop counter* = *initial value* TO *final value* [STEP *increment value*]

The FOR statement is used with NEXT and defines the number of times a FOR—NEXT loop is to be executed. The loop counter must be a simple variable. Starting with a specified initial value, the loop counter moves to the final value by the optionally specified step. If not stated, the step is 1.

```
380 FOR A=1 TO 5
390 FOR B=-1 TO -3 STEP - .3*A
400 PRINT B
410 NEXT B
420 NEXT A
```

GOSUB *statement number*
statement label

Transfers program control to the subroutine beginning at the specified statement number line or statement label. The RETURN statement at the end of the subroutine returns control to the statement following GOSUB.

```
125 GOSUB 760
```

GOTO *statement number*
statement label

Transfers program control to the specified statement or statement label.

```
110 GOTO 310
```

GRAD

Sets grads mode for results and arguments of trigonometric functions. There are 400 grads in a circle. The HP-86/87 is in radians mode when switched on.

```
30 GRAD
```

IF *numeric expression* THEN *statement number / statement label* [ELSE *statement number / statement label*]

Provides conditional branching. If the numeric expresison is evaluated as true, execution is transferred to the specified line. If false, execution continues with the next sequential statement.

```
450 IF X=3 THEN 950
```

Other forms of the IF ... THEN statement include IF ... AND ... THEN, IF ... OR ... THEN, IF ... THEN ... ELSE, IF ... AND ... THEN ... ELSE, and IF ... OR ... THEN ... ELSE.

INPUT *variable name list*

Allows variable assignments to be entered from the keyboard during program execution. INPUT can only be executed in a running program. Separate variable names in list with commas. Any numeric expression is valid for numeric input; any string expression, quoted or unquoted, is valid for string input as long as an unquoted string contains no commas. Null is allowed with string input but not with numeric input. Separate input values with commas.

```
75 INPUT A,B,G$,H(35)
```

INTEGER *dimension list*

Declares the maximum physical size for one dimensional INTEGER precision arrays. Also declares simple variables as having INTEGER precision. Dimension list items are separated by commas. An array name must be followed by the maximum subscript value in parentheses.

```
20 INTEGER A(2),B3
30 INTEGER C,D,E,F(2)
```

[LET] *numeric variable list* = *numeric expression*
[LET] *string variable list* = *string expression*

Assigns a value to a variable or variables. The word LET is optional. Variables in the list are separated by commas.

```
750 X=Y^2
580 LET A,B,C,D,E=100
285 X$="PRICES"
```

NEXT *loop counter*

Used with the FOR statement, defines the last statement of a FOR—NEXT loop and causes the loop counter to be incremented and tested.

```
380 NEXT L
```

NORMAL

Cancels all tracing operations, returns to normal print mode from PRINT ALL mode, and stops AUTO statement numbering.

```
160 NORMAL
```

ON *numeric expression* GOTO *statement number list*
statement label list

Transfers program control to one of one or more statements in the current program based on the nearest integer value of the numeric expression. A value of 1 corresponds to the first statement number in the list, 2 to the second, etc. Statement numbers in the list are separated by commas

```
670 ON A4+3 GOTO 100,200,300,400
```

```
OPTION BASE 1
OPTION BASE 0
```

OPTION BASE 1 specifies that the subscript of the first element of all arrays is one rather than the normal zero. Although OPTION BASE 0 is always in force unless OPTION BASE 1 is executed, an OPTION BASE 0 statement may be used for documentation purposes. The OPTION BASE 0 or OPTION BASE 1 statement must be executed before any statement using an array variable or subscripted variable.

```
10 OPTION BASE 1
```

PAGESIZE *number of lines*

Sets the display screen size at 16 or 24 lines. Default is 16 lines.

PAUSE

Halts execution of a program. Execution may be resumed at the next statement with a CONT command.

```
290 PAUSE
```

PRINT [*print list*]

Causes items in the print list to be printed on the system printer (or current PRINTER IS device). The print list may contain variables, arrays, expressions, or TAB. Items in the print list must be separated by commas or semicolons. A comma advances an item to the beginning of the next print zone. A semicolon supresses the advance to give close spacing. PRINT without a print list "prints" a blank line.

```
760 PRINT
220 PRINT A;TAB(10);B;TAB(24);C
400 PRINT "KEY IN A NUMBER"
585 PRINT A,B,C,D
```

PRINTER IS *output code number* [*number of characters per line*]

Specifies the device that will print the items in a PRINT statement print list. When the HP-86/87 is turned on, the print device is the CRT.

Code Number	Device
1	CRT (display)
701	Printer

```
230 PRINTER IS 701,80
```

RAD

Sets radians mode for all results and arguments of trigonometric functions. There are 2π radians in a circle. The HP-86/87 is always in radians mode unless DEG or GRAD is executed.

```
35 RAD
```

RANDOMIZE [*numeric expression*]

Regenerates the random number seed. With no numeric expression, the seed is generated using the system timer. Specifying a numeric expression enables you to repeat the pseudo-random number sequence. A numeric expression of zero generates a constant series of zeros.

```
40 RANDOMIZE
25 RANDOMIZE PI/10
```

READ *variable name list*

Reads one or more numeric or string items from a DATA statement or statements and assigns them to the specified variables. Variable names in the list are separated by commas.

```
300 READ A,B,C$,A(4)
900 READ V7$
```

REM [*any combination of characters*]
! [*any combination or characters*]
any statement ! [*any combination of characters*]

Places remarks in a program listing to provide documentation to make the listing easier to follow. Either REM or ! may be used immediately following the line number. Only the exclamation mark may follow a statement.

```
45 REM INITIALIZATION
190 A=17 ! AREA OF FIELD #1
```

RESTORE [*statement number*]

Resets the data pointer to the beginning of the specified DATA statement, or to the lowest numbered DATA statement in the current program if no statement is specified, so that values can be reused.

```
250 RESTORE
637 RESTORE 420
```

RETURN

Returns control from a GOSUB branch to the statement immediately following the GOSUB statement that referenced it. RETURN is the last line of a subroutine.

```
300 RETURN
```

STOP

Terminates program execution.

```
70 STOP
```

TRACE

Used to follow the order of statement execution in all or part of a program. Any branching causes a trace output to be printed, showing from which line the branching came and to which line it was going. Cancelled with NORMAL.

```
980 TRACE
```

TRACE ALL

Used often with the (STEP) key to trace all statements and variable assignments in a running program. The TRACE ALL output indicates the previously executed statement, current statement number, and any variable changes. TRACE ALL combines the actions of TRACE and TRACE VAR. Cancelled with NORMAL.

```
990 TRACE ALL
```

TRACE VAR *variable list*

Monitors and reports value changes of listed variables in a running program. The trace output prints the statement number in which the assignment occurred, plus the name and new value of simple variables, the name, element number and new value of all array elements, and the name only of string variables. Array variable names in the list must be followed by an empty pair of parentheses. All variable names in the list are separated by commas. Cancelled with NORMAL.

```
100 TRACE VAR A$,D()
 90 TRACE VAR S,Y,H,J$,K
220 TRACE VAR C(),Z(),W5
```

WAIT *number of milliseconds*

Delays program execution approximately the number of milliseconds specified before it continues. The range of the wait is about 0 to 27 minutes.

```
710 WAIT 250
630 WAIT X
```

BASIC Predefined Functions

ABS(X)	Absolute value of X.
INT(X)	Largest integer $<=$ X.
LGT(X)	Log to the base 10 of X, X $>$ 0.
PI	3.14159265359
RND	Next number, X, in a sequence of pseudo-random numbers, 0 $<=$ X $<$ 1.
SIN(X)	Sine of X.
SQR(X)	Positive square root of X.
TAB(X)	Skips to the specified column. The HP-86/87 has 80 columns or character positions, numbered 1 through 80.

Personal Computer Division
1010 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

Part Number
82832-90002

Printed in U.S.A. 5/83