

```

5 LET R = 0
10 PRINT "HIT THE SPACE-BAR"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = " " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 PRINT R

```

In this program, R can never be less than 1 or greater than 9. By the time the space bar is pressed (and recognised by the INKEYS function in line 50), R will have a value somewhere between 1 and 9 inclusive.

This program was tested 1,000 times and found an average value for R of 5.014. Since a perfect average would be 5 and the error is only 0.28% high, this suggests that the program does indeed generate a random number very close to the theoretical average. The point is, of course, that even when a program appears reasonable on paper, there may be unforeseen flaws in it. Actual testing is well worth while.

Some readers will have noticed that these random number programs could be shortened by using various GOTO statements in place of the FOR . . . NEXT loop. Our reason for avoiding GOTO statements will become clearer in future parts of the Basic Programming course.

Basic Flavours

RANDOMIZE

On the BBC Micro and the Oric-1, delete line 20, and replace line 320 by:

```
320 LET D = INT(10 * RND(1))
```

On the Dragon-32, delete line 20, and replace line 320 by:

```
320 LET D = RND(6)
```

and delete lines 330 and 340.

On the Lynx, replace line 20 by:

```
20 RANDOM
```

On the Vic-20 and the Commodore 64, replace line 20 by:

```
20 LET X = RND(-1)
```

and replace line 320 by:

```
320 LET D = INT(10 * RND(1))
```

INKEYS

On the Oric-1 and the Lynx, replace INKEYS by KEYS

On the Vic-20 and the Commodore 64, replace line 40 on page 173 by:

```
40 GET AS
```

and replace line 50 on page 175 by:

```
50 GET AS: IF AS = " " THEN GOTO 80
```

On the ZX81, modify lines 10 and 50 in the programs on pages 173 and 175 by replacing the space (" ") with "X"

On the BBC Micro, replace INKEYS by INKEY\$(10). The number in brackets is the time in hundredths of a second during which the system will wait for a keypress

Exercises

■ **RND Function** Modify the last program in the text to give a random number in the range 1 to 6 (inclusive).

■ **Loop And Average** Add lines to the last program in the text to make it repeat 100 times and produce an average of the 100 results.

■ **Replace With Subroutine** Replace lines 50 and 130 in the main program (the random dice throw subroutine) with a GOSUB calling your 'random number generator' in the first exercise.

■ **INKEY\$** Using the INKEYS function, how would you write a program to read any key typed at the keyboard and print: THE KEY YOU HIT WAS: * as a result (* represents the key you pressed).

■ **Timing Loop** Write a timing loop (a 'counting' loop) and use the INKEY\$ function to find how big the value of a variable becomes after 10 seconds (you'll need to use a watch). Write the program so that the final printout reads: THE VALUE OF R AFTER 10 SECONDS IS: * (* represents the value of R).

■ **IF-THEN Tests** Write a simple game program in which the computer generates a random number between 1 and 100 (inclusive) and the player has to guess what the number is. The player has five tries. Each time, the program responds with the messages YOUR GUESS IS TOO LARGE, YOUR GUESS IS TOO SMALL, or YOU ARE RIGHT, CONGRATULATIONS, or NO MORE GOES. YOU LOSE!

Answers in the next issue.

Answers To 'Exercises' On pages 148-149

Loops 1

THE VALUE OF A IS 450

Loops 2

START

STOP

Loops 3

THE VALUE OF A IS NOW 160

Loops 4

I LIKE BASIC

I LIKE BASIC

I LIKE BASIC

:

:

Until you RESET or BREAK the program

Loops 5

I'M FEELING LOOPY

(15 times)

Read-Data 1

WE'RE TESTING THE READ STATEMENT

170

Read-Data 2

X=1

X=2

:

:

X=23