Looping The Loop

Breaking out of loops, going round them a required number of times, and line numbering are covered in the second part of our programming course

We ended the last part of the 'Basic Programming' course with the program listed below. It worked fine but because of the GOTO in line 70 the program kept looping back to the beginning and never stopped. The only way to get out of the loop was to use the BREAK key or the RESET key.

Now we are going to look at one of the ways we can get out of a loop like this by incorporating a test in the program. The usual way it's done is to test for a number we would never actually want to use in the program. The program allowed us to type in a number that the computer then printed on the screen with a 1 added to it. We might decide that we would never want to enter a number bigger than 999. In that case we could test to see if the number that has been input is greater than 999. Type in the program and then add:

```
35 IF A > 999 THEN GOTO 80<CR>
```

Now run the program again and it will function as before — unless, that is, you enter a number greater than 999. Try typing 1000<CR> and see what happens.

Why did the program stop this time? The IF in line 35 is what made it happen. When BASIC finds an IF statement it knows that a logical test is coming. The > sign means 'greater than'. Line 35 therefore means IF (variable) A (is greater than) 999 THEN GOTO (line) 80. If you just typed in 1000, the value of A becomes 1000 which is greater than 999 so the program THEN GOes TO line 80 which is the end of the program. If A is not greater than 999, the THEN part of the line is ignored and the program continues to the next line.

When running this program, then, you can input numbers as often as you like, just as long as they are not greater than 999. As soon as a number bigger than 999 is input, the IF-THEN statement detects the fact and terminates the program by GOing TO the END. When a BASIC program has reached the end or been terminated, you will be given a 'ready' prompt on the screen. Depending on your computer, this prompt may take several forms. On the BBC Microcomputer the ready prompt is a sign like this: >. On the Dragon it's OK. On the Sord it's READY. Whatever form it takes, the ready prompt is BASIC's way of telling you that no program is running and that it is awaiting further orders.

There is a lot of variation in the way different versions of BASIC use THEN. Details are given in the 'Basic Flavours' box on page 39.

Other comparisons used in BASIC are < (less than), = (equals), >= (greater than or equal to), <= (less than or equal to) and <> (not equal to). We'll see these comparisons used often as the course progresses.

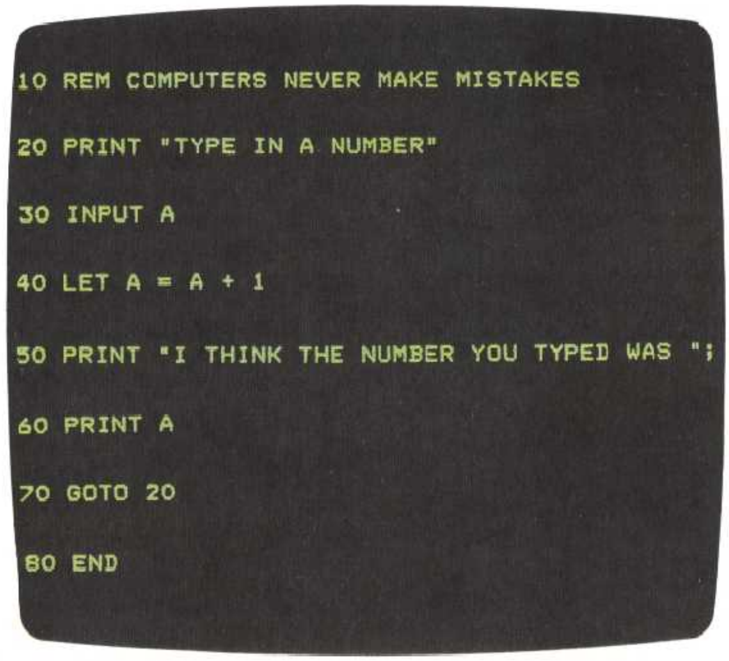
Before continuing any further, it's worth trying out a few exercises to get the feel of using these comparisons.

Exercises

- Change one of the lines so that the program will be aborted if A = 1000.
- Change one of the lines so that the program will be aborted if the number input is less than zero.
- Change the GOTO line so that it makes the program loop back to the beginning if A is equal to or less than 500. Hint: you will not need a separate IF-THEN line and a GOTO line.

Discovering FOR-NEXT

When writing programs there will be many occasions when you would like some items in the program to be repeated a precise number of times. The GOTO in line 70 enabled the program to loop as many times as we wanted. We later added an IF-THEN statement in line 35 which enabled us to escape by entering an 'out of range' number.



```
10 REM COMPUTERS NEVER MAKE MISTAKES

20 PRINT "TYPE IN A NUMBER"

30 INPUT A

40 LET A = A + 1

50 PRINT "I THINK THE NUMBER YOU TYPED WAS ";

60 PRINT A

70 GOTO 20

80 END
```