REM stands for REMark. Anything appearing on the same line after REM is ignored by the computer. Remarks are a handy way of reminding yourself what the computer is doing. This particular REM is just a title — it does not tell us what the program is doing. We'll see how helpful properly written REMs are later in the course. Now let's look at:

```
20 PRINT "TYPE IN A NUMBER"
```

When BASIC gets to the word PRINT, the part that follows it is 'printed' on the computer screen. Notice that the sentence is enclosed in double quote marks. One of BASIC's rules is that the characters (letters) appearing inside double quote marks after a PRINT statement will appear on the screen exactly as they were typed in. We'll see another way of using PRINT in line 60. Next comes:

```
30 INPUT A
```

We'll skip this line for now and come back to it after looking at line 40.

```
40 LET A = A + 1
```

The letter A is used here as a variable. A variable is like a labelled box that can contain either a number or some characters. Instead of having to remember what's in the box, all we have to know is what the box is called in order to reference it. It's like saying "Pass me the box labelled B" instead of "Pass me the box containing the 15mm cheese-head screws".

In this line we have a 'box' called A. This box is called a variable, because the value of what we put in it can vary. We can assign virtually any value to a variable. A value was assigned to variable A in line 30, so let's see how it was done:

```
30 INPUT A
```

Using the word INPUT is one of the ways in BASIC of assigning (giving) a specific value to a variable. When the BASIC program gets to a line starting with INPUT it waits for something to be typed in from the keyboard. INPUT A lets the computer know that we have a variable called A and that whatever is typed in at the keyboard will be assigned to that variable. Typing 7<CR> at this point puts 7 in box A, or to use computer jargon, assigns the value 7 to variable A. Now that we know what a variable is, and one of the ways of assigning a value to it, let's look at line 40 again.

```
40 LET A = A + 1
```

The name of the variable to which a value is assigned always appears on the left of the equals sign. Here we are giving a new value to A. The statement means 'LET the new value of A equal the old value plus 1.' The old value of A was 7. We have now made it 7 + 1, so the new value is 8.

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

This is our print statement again. It 'prints' the character string (that is, the words or numbers you have typed) between the double quote marks. Notice the semi-colon at the end of the line. It helps to specify the positions at which things are printed on the screen. Later in the course we'll return to how the semi-colon is used in more detail. Now let's look at:

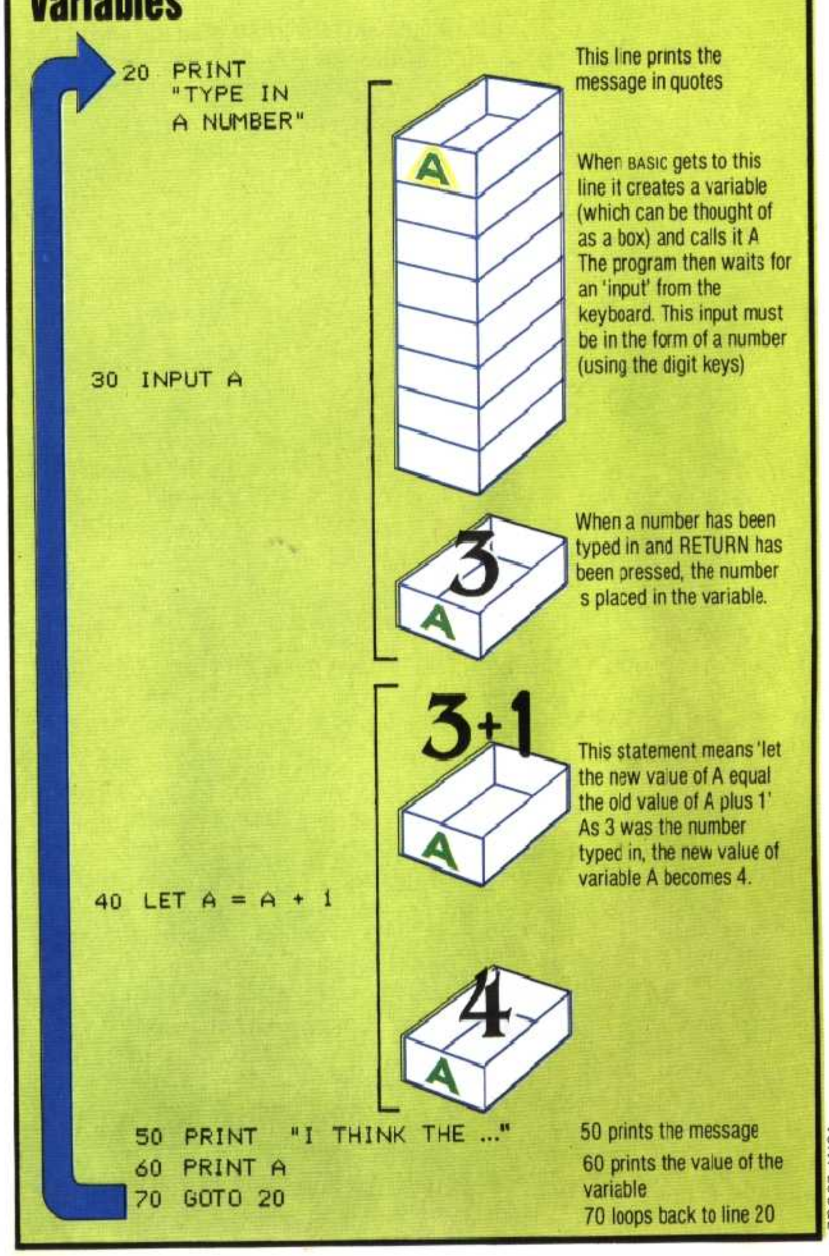
```
60 PRINT A
```

Here's another PRINT statement, but this time there are no quote marks around the A. We already know that the program will not print an actual A on the screen because we have seen that quote marks are needed to do that. Without the quotes, BASIC looks for a variable with the same label as the character after PRINT. If it finds one, it prints the value of the variable. (If it doesn't find one, it gives an error message!) This program already has a variable called A and so BASIC prints its value — what is it?

If you thought the answer was 7, remember that BASIC works through programs line by line,

The box below shows how 'variables' are used in BASIC. It also illustrates how the GOTO statement (see next page) is used to form a loop

Variables



20 PRINT "TYPE IN A NUMBER"
This line prints the message in quotes
A
When BASIC gets to this line it creates a variable (which can be thought of as a box) and calls it A The program then waits for an 'input' from the keyboard. This input must be in the form of a number (using the digit keys)
30 INPUT A
3
A
When a number has been typed in and RETURN has been pressed, the number s placed in the variable.
3+1
A
This statement means 'let the new value of A equal the old value of A plus 1' As 3 was the number typed in, the new value of variable A becomes 4.
40 LET A = A + 1
4
A
50 PRINT "I THINK THE ..."
60 PRINT A
70 GOTO 20
50 prints the message
60 prints the value of the variable
70 loops back to line 20

TONY LODGE