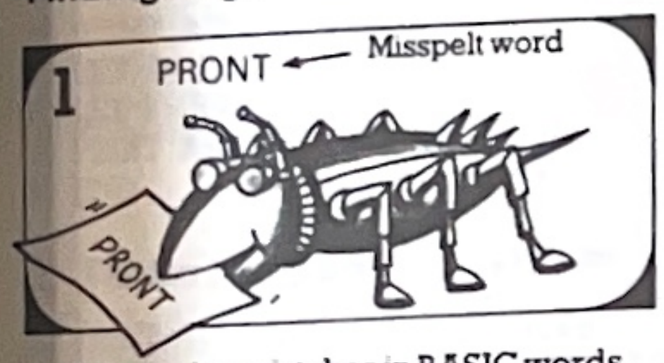
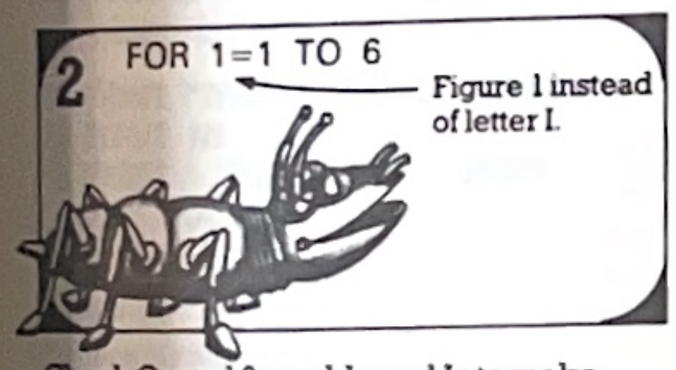
Programming tips

On these two pages there are some tips to help you write your own programs, and a list of the most common bugs you might get, and what causes them. The most likely bugs are listed first, so if you have a program which will not work, check through this list until you find the reason.

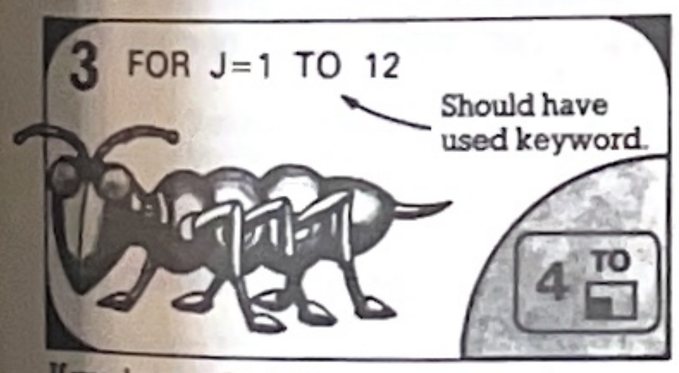
Finding bugs



Look for typing mistakes in BASIC words. If you misspell one of these words the computer will not recognize it.



Check Os and 0s and 1s and Is to make sure you have typed the right ones in the right places.



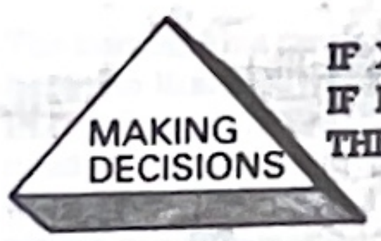
If you have a Sinclair computer, make sure you have not typed a word in letter by letter instead of pressing the key for that word.

Writing programs

When you are writing programs it helps to remember that the computer can carry out three main activities: simple instructions, repeating things and making decisions. These are the building blocks of all programs.


```
LET A=3
LET N=N+1
PRINT A/T
PLOT (X,Y)
```

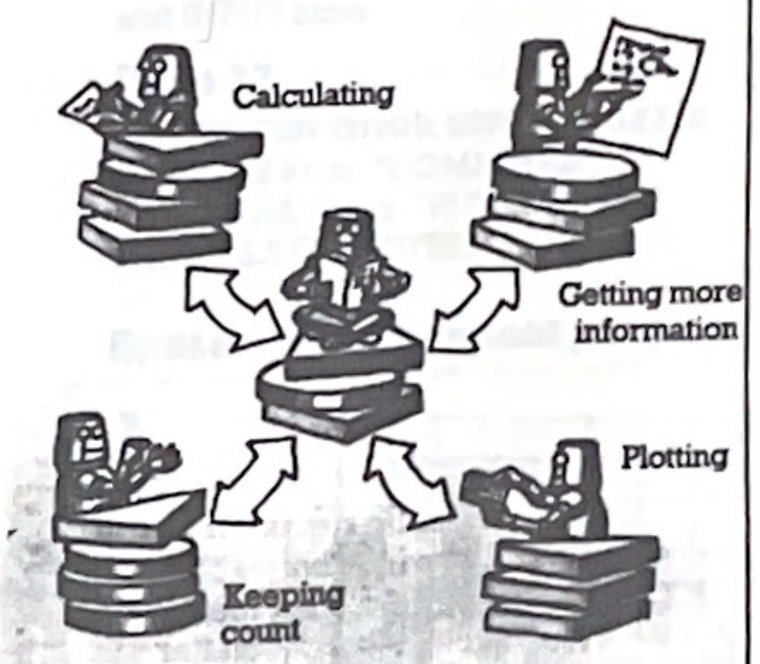

```
FOR J=1 TO 6

20 LET A=1
30 IFA< 10 THEN
   GOTO 100
```



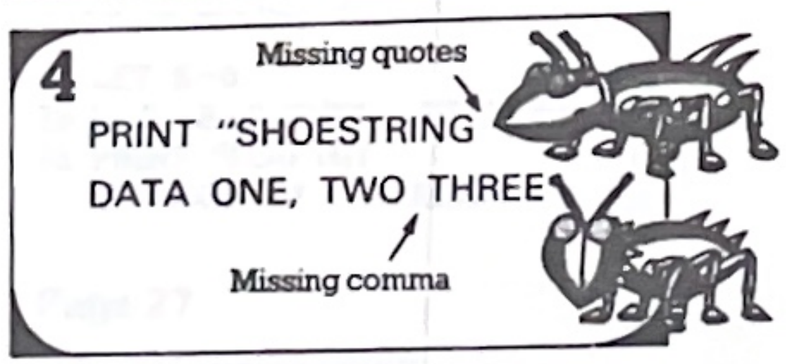

```
IF X=Y THEN STOP
IF K$="HELLO"
THEN PRINT A
```

This book has covered all the main instructions you need in BASIC to tell the computer to carry out these activities. When you are writing a program, work out what the computer needs to do at each stage, then decide which instructions you want to use.

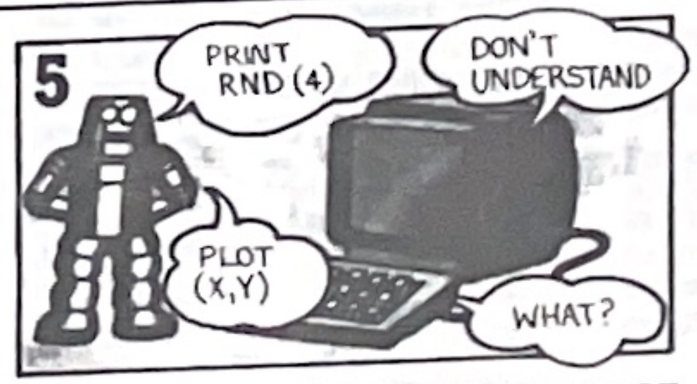
There are usually several different ways to write a program and some of them may be neater and shorter than others. When you are writing a long program it is a good idea to divide it up into lots of sections with subroutines to carry out each activity. The central core of the program may be a simple set of instructions, decisions and repeats which controls when and how often the computer carries out the subroutines.



Breaking up programs into sections like this makes it much easier to find any mistakes. Each section can usually be tested by itself without running through all the program. Remember to label each section with a REM line so you know what it is for.



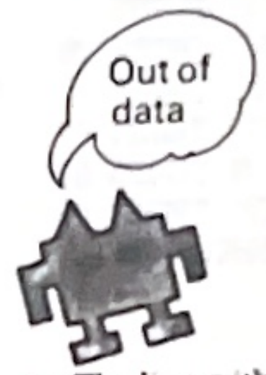
Make sure you have not left out any quotation marks, or the commas between data items. Check complicated lines which have lots of symbols especially carefully.



Make sure you use the correct RND, PLOT and CLS commands for the computer. Check, too, that you have given the computer a general graphics line if it needs one.

Error messages

All computers print out some sort of message when there is a bug in the program and the messages are explained in the computer's manual. Here are some of the most common messages you may get.



◀ This means there are not enough data items for the computer to read in the DATA lines. It may be because you have missed out a comma between two items, so the computer has read them as one.

▶ The line with the number given in a GOTO or GOSUB statement does not exist. You may have accidentally erased the line by typing in another line with the same number, or you may have just mistyped the number.

◀ You may get this report on a BBC or Sinclair computer. It usually means you have not set up a variable with a line such as LET C=0 or LET C="" before using it.

▶ This means the NEXT line of a loop is missing. It may be because you typed the wrong variable name, or even put a 1 instead of an I so the computer did not recognize it.

Last word

Some bugs are very hard to find, but if the computer will not run the program there must be a bug in it somewhere. If you really cannot find the bug, try typing in suspect or complicated lines again, you might get them right the second time without even noticing what the bug was.