



been generated is passed back.

Data moves around programs inside variables and the freedom of movement of a variable is called its 'scope'. Many programming languages can restrict the scope of a variable to particular subroutines. In PASCAL, the variables used in a particular subroutine (procedure) must be 'declared' for that procedure. Variables declared for the main program are *global* and may be used anywhere in the program (including within any of its modules). Variables declared within a particular procedure, however, are *local* to that procedure and can only be used there.

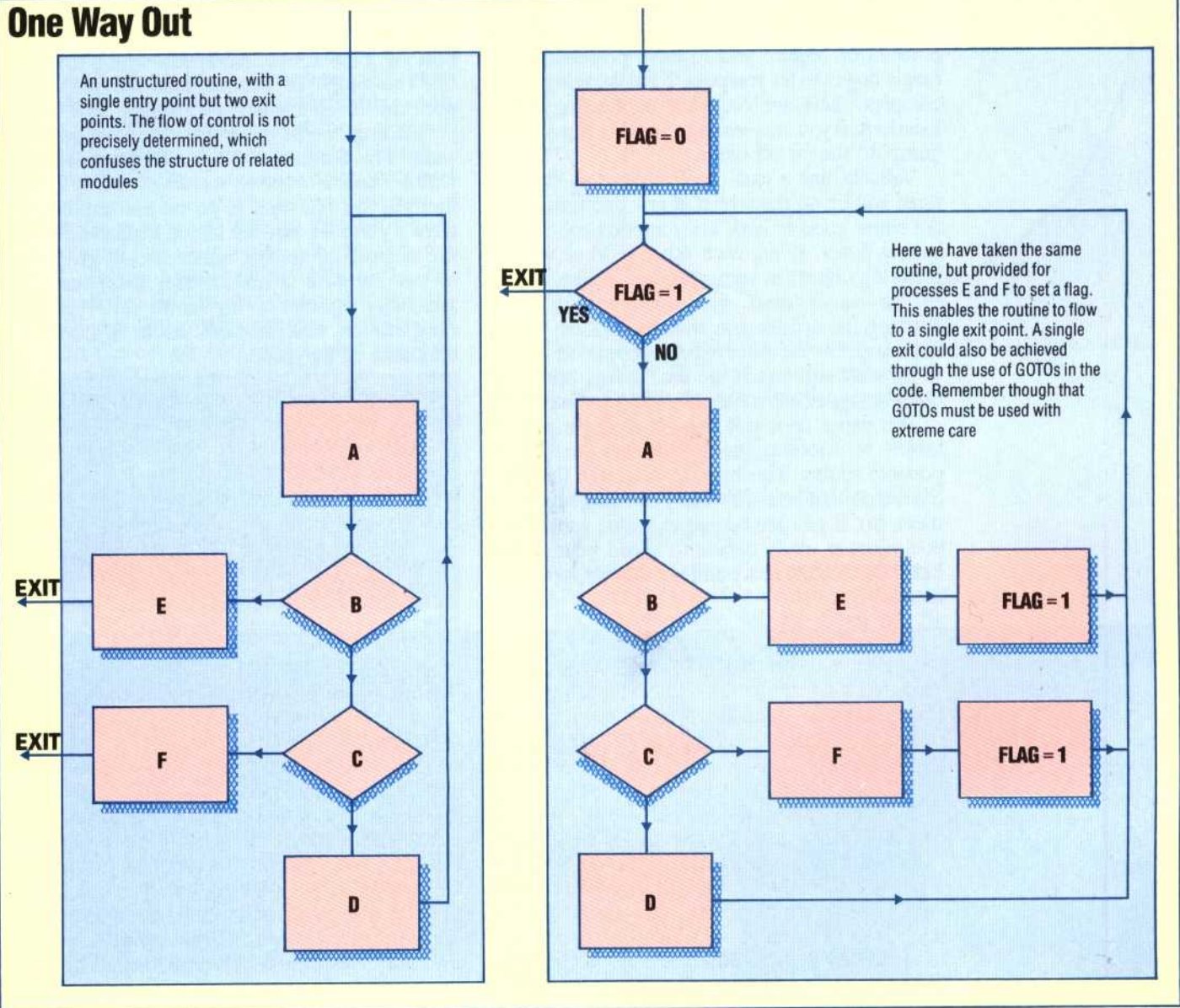
Local variables can have the same names as global ones and using one does not affect the value of the other. Using a language that supports local variables allows us to write subroutines without having to worry about how the variables used in the routine might affect variables in other routines. Unfortunately, very few versions of the BASIC language support local variables, which means that if we wish to write independent

subroutines we must somehow simulate the effect of having local variables.

The simplest way to do this is to adopt naming conventions that distinguish variables that do different jobs. Some conventions already exist and are widely used by programmers. Using I, J and K as loop counters and index values is very common, a practice that has been adopted from mathematics.

Having described a program with a flowchart, it is a simple matter to number the subroutines involved, or to give them some other kind of code. Any global variables that need to be made local to a particular subroutine can then be suffixed by this code to make them unique. Thus, routine number 5 may use the local variables SUM5 and TOTAL5 to distinguish them from SUM12 and TOTAL12 in routine number 12. Be careful though that the BASIC you are using doesn't look only at the first two characters! Variables that are used to pass values between subroutines and those used only in the main program need not be coded.

One Way Out



NODIX271