



C

COMPLEMENT

The *complement* of a single-digit number is the value that, when added to the number, results in the base value of the number system. In decimal (base 10), the complement of 3 is 7. This enables subtraction to be performed by using the addition function and a couple of other very simple operators. Suppose in decimal we wish to subtract three from eight ($8-3=?$). An alternative way of writing the expression is:

$$(8 + (10 - 3)) - 10 = ?$$

This may seem more complicated, but it isn't. Seven ($10 - 3$) is the complement of three, and to subtract 10 from the final result, we need only strike off the leftmost digit.

To complement in binary, the obvious approach is to swap all ones and zeros, so that 1011 becomes 0100. This is called *one's complement*. More commonly used, however, is *two's complement*, which consists of a one's complement with one added to the result, so that 1011 becomes 0101. Let's try our initial subtraction problem in binary:

Three is 0011
Two's complement is thus 1101
Eight is 1000
Adding these numbers gives 10101
Deleting the leftmost digit gives five (0101).

CONCATENATE

The dictionary definition of the verb *concatenate* is 'to join or link together, especially in the form of a chain'. In programming terms it means the same, and is usually applied either to alphanumeric strings in RAM or to files on a disk. The assignment statement $AS = BS + CS$ is one form of concatenation. To concatenate two or more files on a disk means to link them into one larger file, with a single new filename.

CONCURRENCY

A *concurrent* operating system on a microcomputer is one that allows several applications or utility programs to be run simultaneously on the same processor. In practice, this means switching rapidly and automatically between tasks, which is why the technique is also

known as *multi-tasking* or *time slicing*. Though multi-tasking has been common on mainframe computers for many years, full scale concurrency has yet to be widely implemented on microcomputers — which, until recently, have featured relatively slow processors. Concurrent CP/M from Digital Research will run up to four CP/M-86 programs simultaneously.

Since there is only one keyboard, screen and operator in such a system, it follows that three of the tasks must be autonomous — they have no need for operator intervention. By pressing a couple of keys, the operator can select which of the four tasks is shown on the screen and can start or finish any of them. The most sophisticated feature of Concurrent CP/M is known as *piping* — automatically directing the output of one task to the input of another. An application of this might consist of using a database to select names and addresses from a file, feeding these to a word processing program and using a third program to print the reports in the background.

CONSTANT

Unlike a variable, which is a character or string used to label an item stored in the computer that may change its value during a program run, a *constant* is any item (string or numeric) that remains unchanged. In the program statement:

$$10 A = B * 6.5731$$

A and B are variables, while 6.5731 is a constant. This should be familiar ground to anyone who has mastered BASIC, but many users are not aware of the implications of using constants or variables.

Constants should be used sparingly because they have several drawbacks, and should be replaced by variables wherever possible. For a start, they slow down calculations. This is because most computers perform arithmetic in floating point binary format, not in binary coded decimal (see page 168). Each time BASIC encounters a line such as the example given, it must convert the constants into floating point format, whereas the variables are already stored in this format.

Another drawback is that constants waste memory if the same value is used several times in a program. It is better to assign the constant to a single-letter variable at the start and use that instead. Programs using variables are easier to edit, as all constant values may be found in a block of assignment statements at the start of the program, making changes much easier.

CONTENTS ADDRESSABLE

This is a method of addressing a memory location by referring to the data stored there rather than by specifying the address itself. This is particularly useful in database applications, providing a search mechanism for a chosen data item. For example, a file of names, addresses and telephone numbers could be searched for occurrences of dialling code '01'; the result will be a list of names and addresses of people living in London.

Multi-Tasking

Concurrent CP/M is available on many business micros, including the IBM PC. Fast disk drives, a 16-bit processor and large memory are essential when four separate tasks are concurrent

