CHAIN

This is the name for another form of data structure, similar to 'tree', 'stack' and 'list'. In a *chain*, each item of data contains a pointer to the location of the next item. This is particularly useful for storing information on disk, where for reasons of efficiency it is often necessary to spread a single file across several areas on the disk surface.

As the name implies, however, it needs only one of the links in the chain to be broken — perhaps by a tiny flaw in the magnetic recording surface — for the whole of the file to be lost, so additional means of access are usually built in for safety.

Many computers can also 'chain' programs. In this case the computer loads one program, runs it, then automatically loads another program and repeats the process. A number of short programs can thus be 'chained' to give the effect of a single long program. A common example of this is in games programs, which often load the playing instructions first, and, once they have been read, overwrite them with the main program. In this way the whole of memory becomes available to the main program and its variables, and no space need be wasted on storing instructions.

A *daisy-chain* is somewhat different in that it describes a hardware configuration — usually the way that peripherals are connected to the central computer. One such configuration has each peripheral plugging in to its own dedicated port, another uses a common data bus to link all devices together (as in a local area network). In a daisy-chain, the first peripheral plugs into the computer's peripheral port. This peripheral features its own port, into which a second add-on device is plugged, and so on. The end result has the appearance of a daisy-chain, with the flowers representing the devices, and the stalks the links between them.

CHANNEL

A *channel* is a route through which data can flow, though in microcomputer terms it refers not to the bus or interface that transmits the data, but the software that controls it. With most operating systems, before data can be sent to a device (the screen, disks or printer, for example) a channel must first be opened, which amongst other things will usually reserve an area of RAM to act as a buffer.

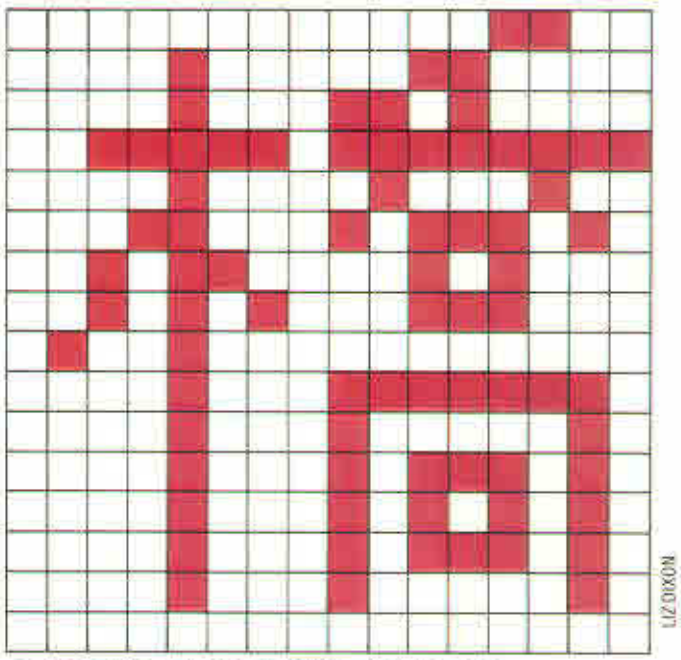
Thereafter, all data for that device will be sent to the channel, which is identified by a number or a name, and the operating system will then automatically route it to the hardware device, without further intervention by the program. On some older computers all this had to be set up by the programmer, whereas on many home computers the whole process is now taken care of by the operating system.

One of the advantages of using channels is that the programmer is not dealing directly with the hardware devices. If a program has been written to address a printer through, say, channel number five, and the programmer wishes to convert it to run on a plotter, he need only change one instruction at the start of the program to specify that channel five now refers to the plotter, instead of having to change all the PRINT and WRITE statements individually.

CHARACTER GENERATOR

The device that specifies the design of characters on the screen is known as the *character generator*. This is a section of the memory containing patterns of ones and zeros that specify the arrangement of dots on the screen.

These patterns are generally stored in ROM, although most home computers will allow the user to specify an area of RAM to replace the standard character generator. This enables you to alter the designs of any alphanumeric or graphic characters, and create your own customised character set.



User-Defined Character
Character generators build text and other characters from a number of dots. Some character generators allow users to define their own characters, such as the Japanese symbol for 'bridge' shown here. However, few micros possess the necessary 16 by 16 grid of dots for such a complicated symbol

CHECK DIGIT/CHECK BIT

A specific number included in data to help spot errors in its recording or transmission is known as a *check digit* or *check bit*. Check digits are most commonly used when transmitting information over a long-distance line, or recording it onto a magnetic surface — both operations that are prone to errors.

The check digit is simply an additional piece of information that is a mathematical product of the actual data being sent, and is transmitted at the end of that data. When the data is received, or read back off the disk, the mathematical function is again performed, and if the new result does not agree with the check digit, an error has occurred.

The mathematical function involved may be quite complex, or it may be as simple as adding together all the bytes in the data block, dividing the result by 256 and then using the remainder as the check digit — this is usually known as a *checksum*.

Check digits are not the exclusive preserve of computers — they can be found in some credit card and cash retrieval systems, as well as International Standard Book Numbers found on book covers.