Controlling Computers

Your computer 'hardware' won't run without the aid of appropriate 'software'. We explain this crucial term — and how to assess the software you find in the shops

Software is the invisible half of a computer system. Without software, the computer is no more than an inert mass of electronic machinery. Without software, the computer can do literally nothing.

Peek inside the silicon chips of a computer and you will find they consist of thousands, perhaps millions, of microscopic electronic switches. Just as a light switch cannot turn a light on or off by itself, the switches in a computer need to be made to turn on or off. They don't all turn on or off together, however. Each individual switch needs to be specifically turned on (or off) and in exactly the right sequence in relation to all the other thousands of switches. Software is how that is done.

Software is the name given to the instructions which make the computer work. These instructions are in the form of numbers which, when presented to the CPU (the heart of the computer; see page 4), set and reset the internal switches to cause specific things to happen. These numbers are only 'understood' by the computer when they are in so-called binary form (converted into ones and zeros as explained on page 28).

These ones and zeros which the computer understands (in the sense that they make it perform specified tasks) are the end product of a long chain of events that started as ideas in the mind of the program writer. A computer program ('program' is the word for any particular single piece of software) can exist in many different forms. The only definite thing we can say about any program is that it must end up in the form the computer understands. Let's take a specific example. Suppose a traffic engineer wants to control a set of traffic lights using a computer. To do this the controlling computer will need a program to make it instigate the correct sequence of events (it's no good having all the lights on green at the same time!). But before this software can be written, the engineer has to think carefully about what exactly it is he wants the computer to do. Usually, these

ROM

ROM (Read Only Memory) is one of the main kinds of computer memory devices. A product of the silicon chip revolution, it allows computer programs to be stored permanently. Most home computers are supplied with a ROM chip containing the BASIC programming language. Other ROMs can be bought for some computers to upgrade their performance by adding another language. Word processor ROMs, which turn the computer into an 'intelligent typewriter', are also available

CASSETTE

Software is often supplied on cassette tape identical to that used in recording sound. Games programs usually come in this form. A program is fed from the tape into the computer by connecting the machine to an ordinary cassette recorder and 'playing' the program cassette. The tape is stopped when the program has been loaded and the computer normally does not need to 'look' at it again

FLOPPY DISK

Software (programs) can be stored by recording it on a disk of magnetic film. The recording is made in 'tracks' on the surface, like the bands on an ordinary LP, by a magnetic 'read/write' head, which also 'reads' (plays back) the program when required. Disks offer a large capacity and a high speed of operation, which have to be paid for: they need sophisticated 'disk drives' (see page 8), which are expensive

CARTRIDGE

A cartridge is essentially a ROM packaged in a convenient housing. Some home computers have readily accessible sockets into which these cartridges can be plugged. The software that comes on cartridges tends to be either a programming language (such as BASIC) or sophisticated arcade-style games

To make your computer work, it needs to be 'fed' with software (a set of electronic instructions). The devices pictured here are the 'media' on which those instructions can be stored. They represent the four commonest ways in which software is supplied. Each has its own special advantages. Software is tailored for each make of computer — a program written for one make will not necessarily work on another