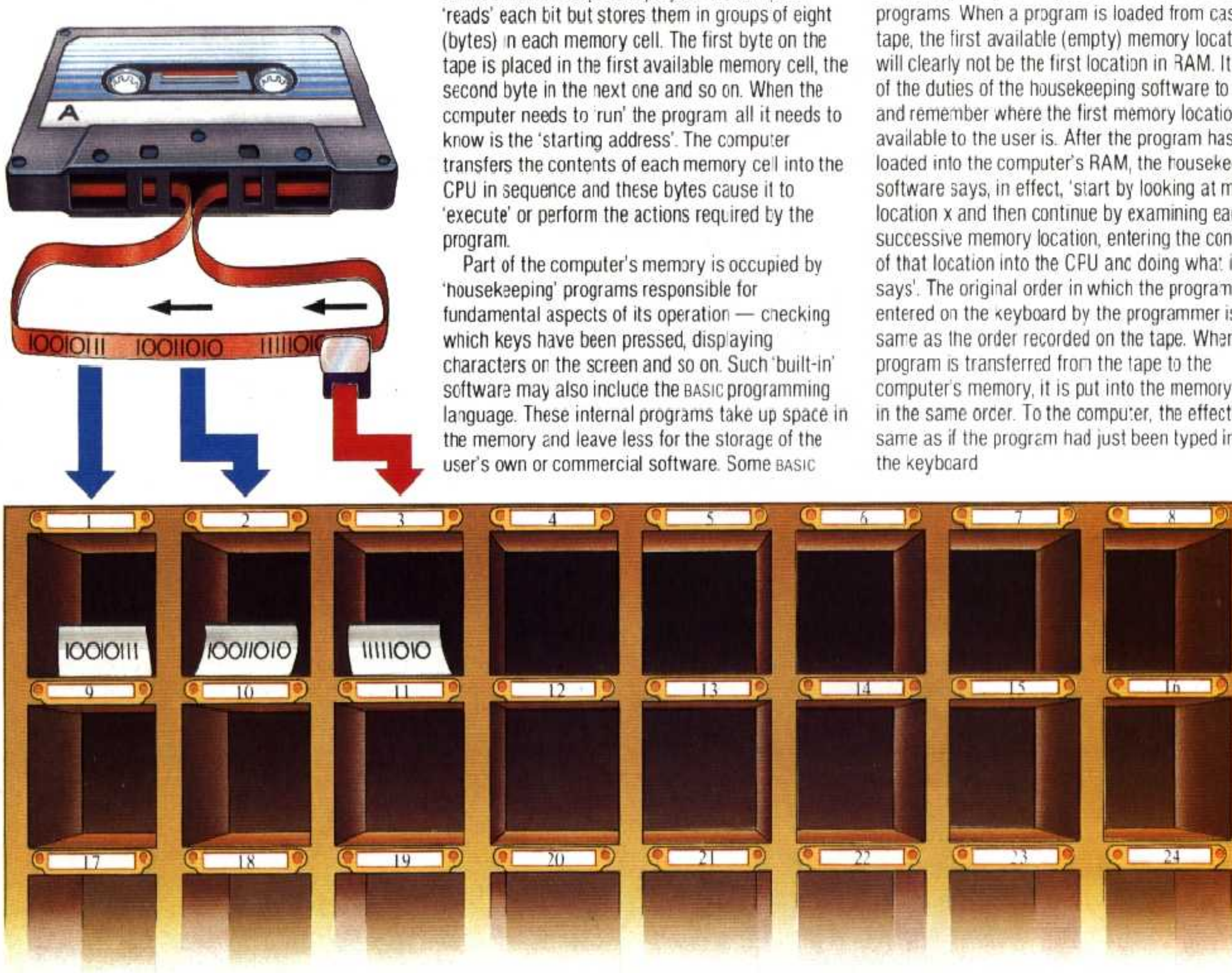
Memory



TONY LODGE

Programs recorded on tape are stored 'sequentially', with every bit from each byte recorded one after the other. When the tape is replayed, the computer 'reads' each bit but stores them in groups of eight (bytes) in each memory cell. The first byte on the tape is placed in the first available memory cell, the second byte in the next one and so on. When the computer needs to 'run' the program all it needs to know is the 'starting address'. The computer transfers the contents of each memory cell into the CPU in sequence and these bytes cause it to 'execute' or perform the actions required by the program.

Part of the computer's memory is occupied by 'housekeeping' programs responsible for fundamental aspects of its operation — checking which keys have been pressed, displaying characters on the screen and so on. Such 'built-in' software may also include the BASIC programming language. These internal programs take up space in the memory and leave less for the storage of the user's own or commercial software. Some BASIC versions, for example, are stored in 16 Kbytes of memory. If the computer is supplied with 64 Kbytes of memory, only 48 Kbytes will be left for other programs. When a program is loaded from cassette tape, the first available (empty) memory location will clearly not be the first location in RAM. It is one of the duties of the housekeeping software to know and remember where the first memory location available to the user is. After the program has been loaded into the computer's RAM, the housekeeping software says, in effect, 'start by looking at memory location x and then continue by examining each successive memory location, entering the contents of that location into the CPU and doing what it says'. The original order in which the program was entered on the keyboard by the programmer is the same as the order recorded on the tape. When the program is transferred from the tape to the computer's memory, it is put into the memory cells in the same order. To the computer, the effect is the same as if the program had just been typed in on the keyboard.

Computers have no intelligence, and cannot organise their memories for themselves. The only reason a computer can store anything is because someone has put it into the right pigeonhole in the right order, and at the right time. How does that happen in a typical home computer system?

When you turn your home computer on, a message usually appears on the screen to tell you it is working. In most cases, it also informs you that you can start writing a program. This message, and the facilities that let you start programming, are stored in part of the computer's internal memory; they need to be stored in the long-term memory (usually in a Read Only Memory or ROM chip — see page 9).

This portion of the computer's memory contains programs that check if keys have been pressed, 'print' letters on the screen, and perform other essential 'housekeeping' jobs. It also contains a special program that translates commands usually written in BASIC into the much simpler binary language of ones and zeros understood by the computer.

When the home computer is switched on, the message on the screen often says 'x bytes free', where 'x' is something like 15,797 or another such strange number. What this tells you is the number of pigeonholes in the computer's memory that are free for you to use. Hitting keys on the keyboard starts to fill these free pigeonholes up — and here we come to the other important thing about computer memory, the *order* in which information is stored.

Pressing a key on the keyboard sends one byte (representing the letter pressed — see page 3) to the memory for storage. Hitting the 'k' key, for example, puts the letter 'k' into a pigeonhole in memory in binary form.

But which pigeonhole does that 'k' go into? It goes into the first free slot in the computer's short-term memory. If you think of a block of empty pigeonholes hung on a wall, the 'k' would go in one in the top left-hand corner.

Hit another key, say 'e', and the appropriate pattern of bits goes into the second empty hole, to the right of the 'k'. Hit a third, a 'y', and it goes into the third hole next to the 'e'. Looking at our block of pigeonholes, the codes for the word 'key' appear on the top row.

The computer has an internal 'counter' to assess which pigeonhole it has reached; it knows where to start because the built-in 'housekeeping' program tells it where the free area of memory starts. As each letter is stored, the counter is increased by one to nominate the next pigeonhole for the next letter typed.