# The human computer
# Debriefing: Register

## How realistic was this?

## Not so bad.

One important concept that was illustrated here is that numbers, letters or instructions are stored exactly in the same way: binary numbers. So the register has no way to know which is which: it is the responsibility of the programmer to write the instructions so that they access what they should (e.g. a JMP should jump to the location of an executable instruction).

Another important aspect is that, if the register is unreliable, e.g. writes a wrong value when instructed to do something which was yet correct, the whole function of the computer can be hampered. More importantly, the error is nearly impossible to detect and recover from, since nothing resembles a binary value more than another binary value. If the JMP now links to a wrong cell, so be it.

Finally, important to remember a register is not part of the computer's main memory. This means it is tightly integrated in the CPU (see below) and ensures very fast access from the control unit. For more details on main memory, look at the debrief page for that part of the computer.

## Of course, in reality, registers are more complex.

In the program role-played here, the register was only used to hold the truth values used to determine whether branching should be done or not. This type of register is called a "conditional register".

The program also used another type of register, the program counter which holds the address of the next instruction to be executed. This register is called a "control register". There are many other types of registers, depending on the type of processor. These can be classified according to their content or instructions that operate on them, for instance data registers (used to hold numeric values) and address registers (used to hold addresses and are used by instructions that indirectly access memory).

In normal computers, registers are used for many other things. Most, but not all, modern computer architectures operate on the principle of moving data from main memory into registers, operating on them, then moving the result back into main memory—a so-called load-store architecture. A common property of computer programs is locality of reference: the same values are often accessed repeatedly; and holding these frequently used values in registers improves program execution performance.

## Remember:

CPU (central processing unit) = control unity + arithmetic and logic unit (ALU) + registers + basic input and output devices

Most basic computer= CPU + memory

A CPU is usually constructed on a single integrated circuit called a micro processor (e.g. Intel Pentium III). CPU and memory are usually incorporated with other devices on an electronic circuit called the motherboard.

For more information, you can have a look at the role-sheets and debriefing information of the parts played by other people in the class.