Pointers in C / C++

# Memory

**Volatile memory** (RAM) is a data structure comparable to an array of bytes. To identify an element in an array, we use its index. In the case of RAM, we use **addresses** instead of indices, but the principle is the same. Each byte of RAM has a unique address, just as each element of an array has a unique index. An address is stored as 4 bytes.

A **memory block** is a grouping of consecutive bytes of memory. The **address** of a memory block is the address of the first byte of the block. To identify a memory block, we must know where it begins (and thus what its address is), and what its size is.

# The concept of pointers

To manipulate the addresses of memory blocks, we use **pointers**. The value of a pointer is the address of the block, and the type of the pointer determines the size of the block.

To declare a pointer named **ptr** that points to a memory block of type **<type>**, we write **<type> \* ptr;**

# Accessing a pointed-to block: the star operator (\*)

To access the memory block pointed to by a pointer, we use the **\*** operator (known as the **dereference** operator). Thus, if **<expr>** is an expression of type **<type> \***, then **\*<expr>** represents the value at the memory address that is the value of **<expr>,** and its size is that of **<type>**.

# Address of a memory block: the ampersand operator (&)

To get the address of a memory block, use the **&** operator (known as the **address-of** operator). Thus, if **<expr>** designates a memory block of type **<type>**, **&<expr>** returns the address of the block, and is of type **<type> \***.