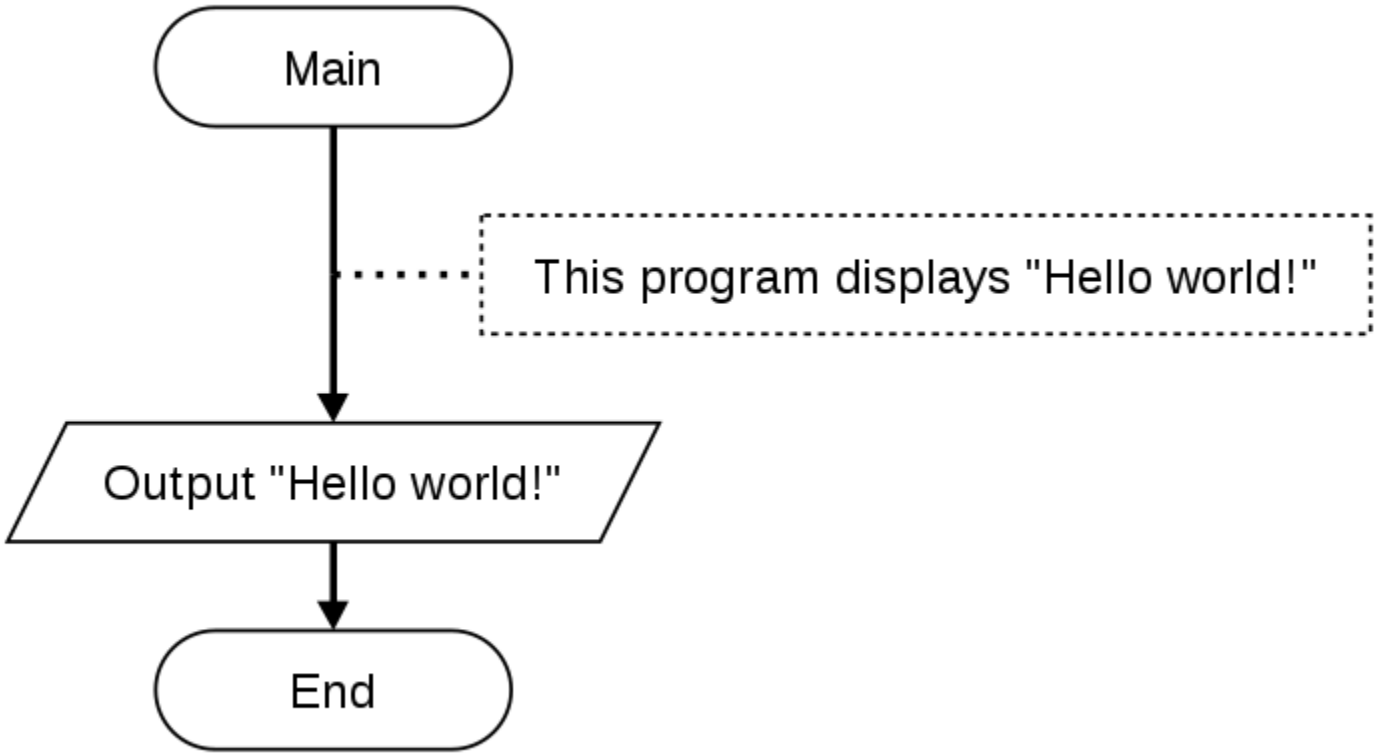
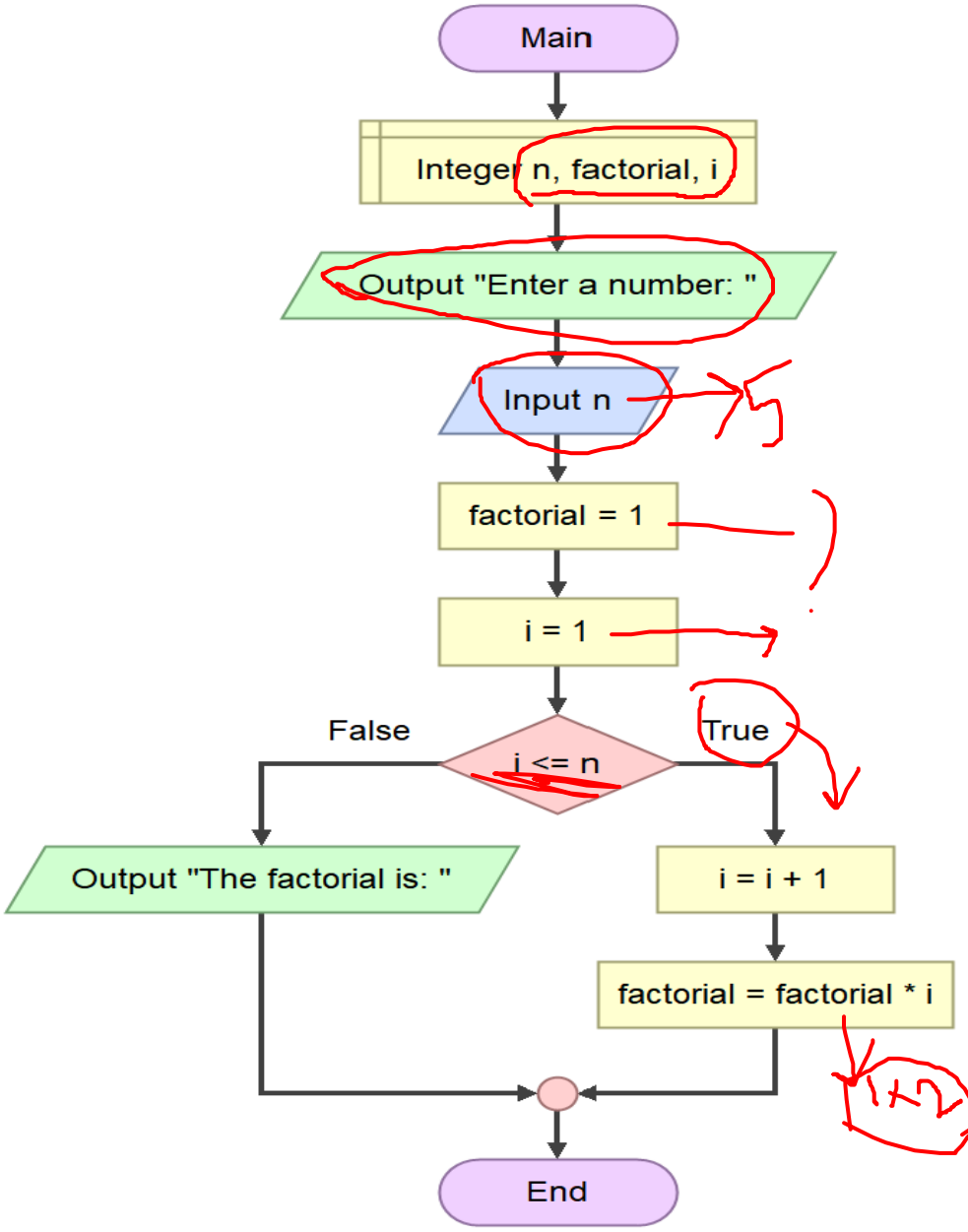
C++ L01

WRITING OUR FIRST PROGRAM!

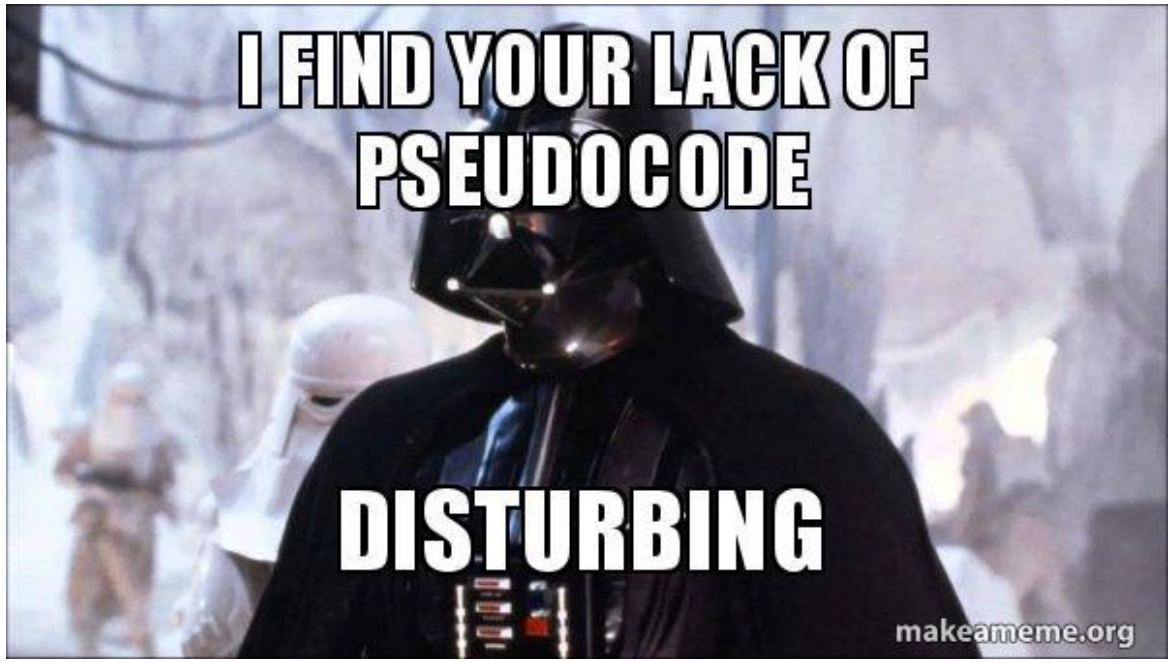
PSEUDOCODES

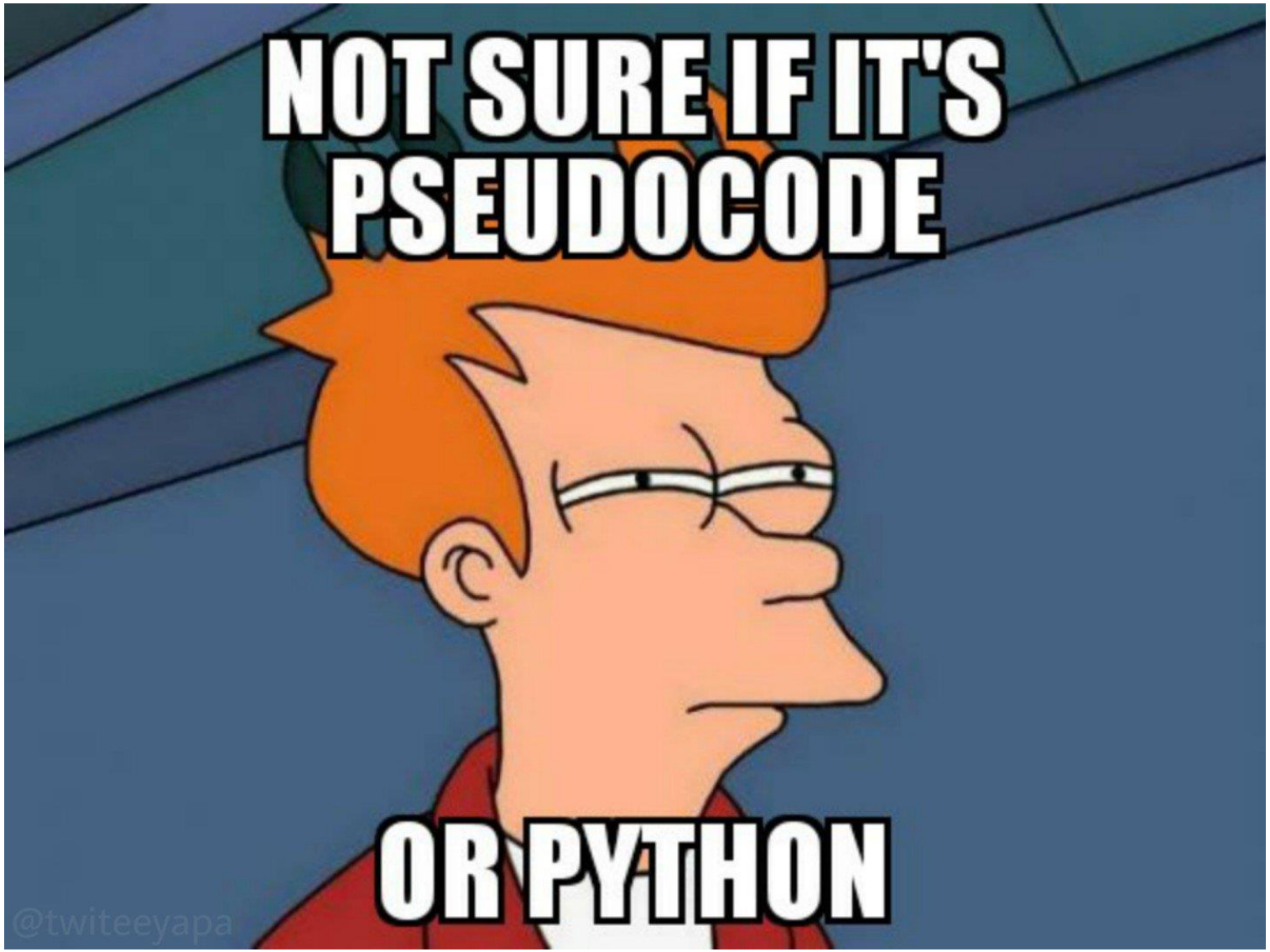



Main
Integer n, factorial, i
Output "Enter a number: "
Input n
factorial = 1
i = 1
False
i <= n
True
Output "The factorial is: "
i = i + 1
factorial = factorial * i
End

PSEUDOCODE FOR A FACTORIAL

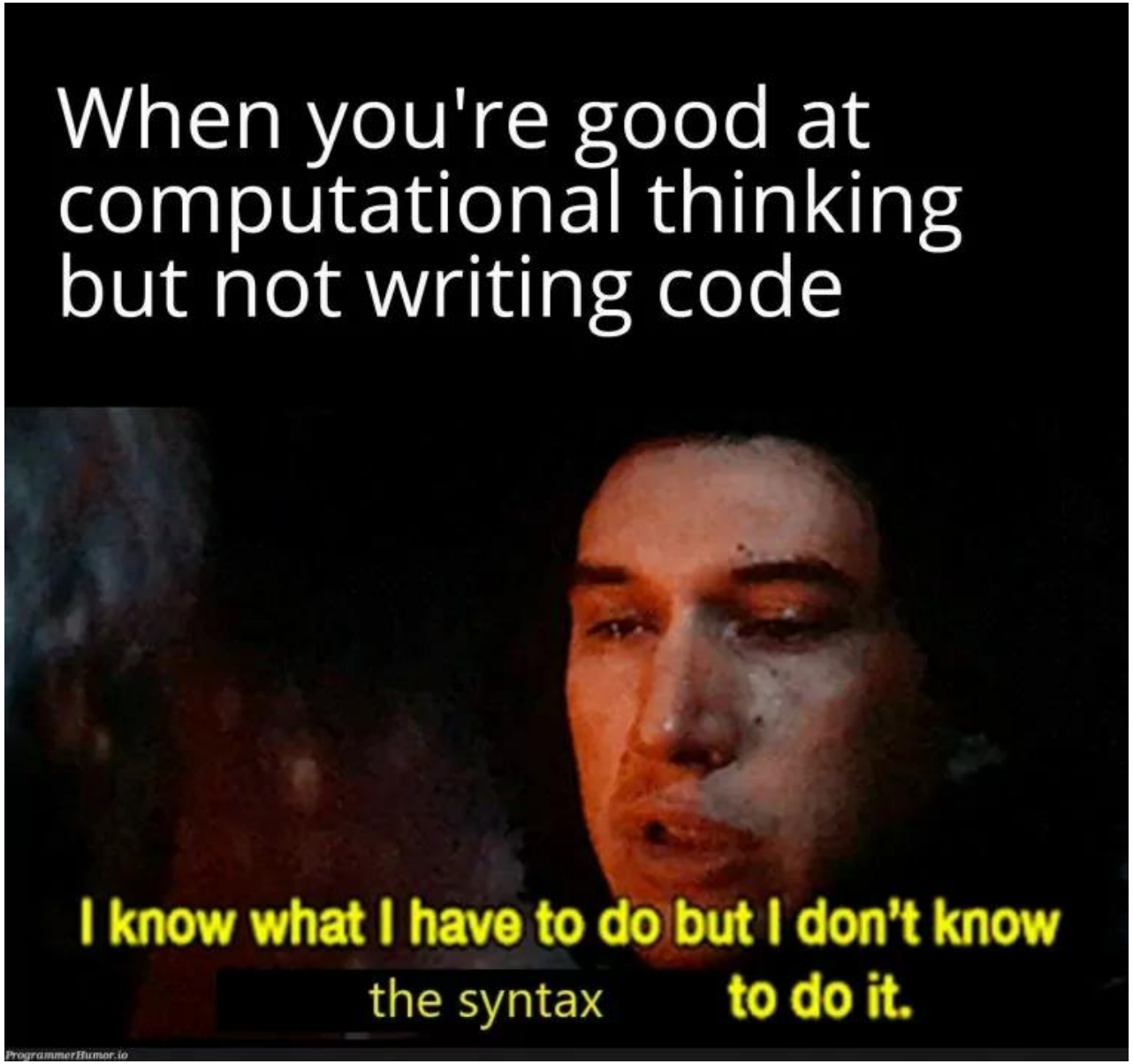
```
IF spaceship sprite touches asteroid sprite THEN
    show explosion sprite
    play explosion sound
    subtract a life
END IF


IF lives = 0 THEN
    stop game
    show game over screen
ELSE
    restart game
ENDIF
```



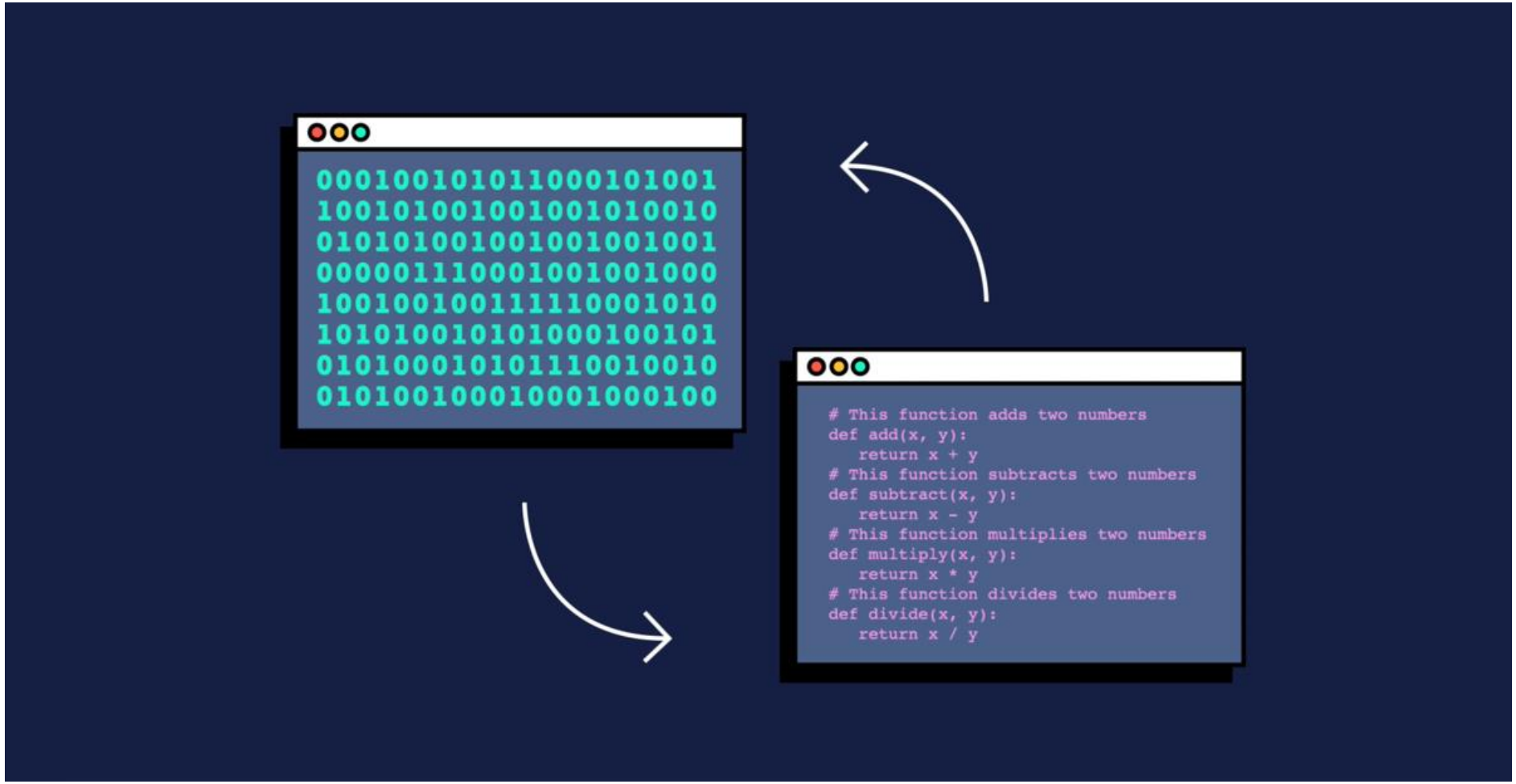

NOT SURE IF IT'S
PSEUDOCODE
OR PYTHON
@twiteeyapa

HOMEWORK

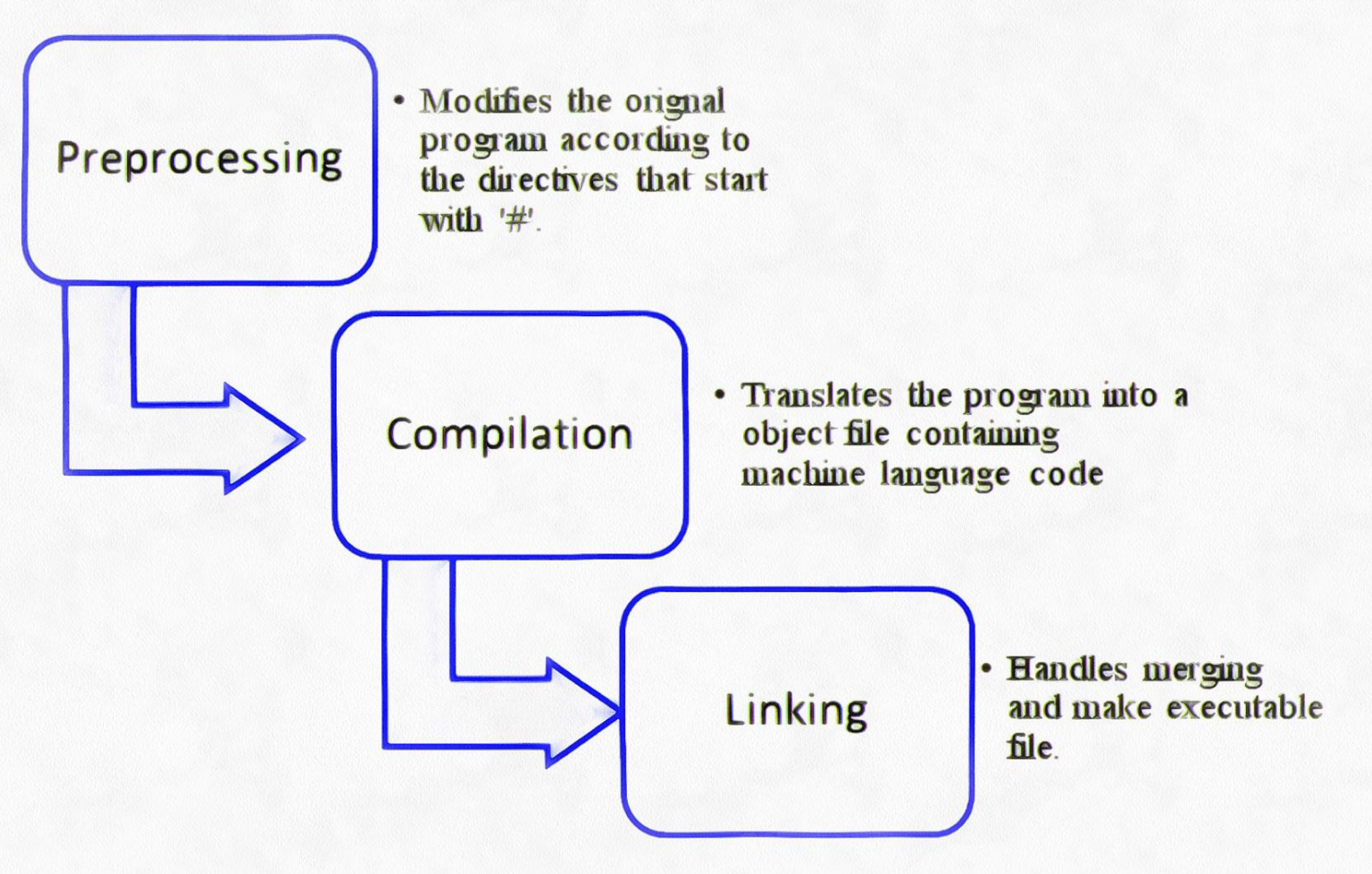
1. Write a pseudocode to print numbers from 1 to 10.
2. Write pseudocode to print if the integer which the user input is prime or not prime.
3. Write a pseudocode to print sum of n(integer which the user input) numbers starting from 1 to n.


When you're good at computational thinking but not writing code
I know what I have to do but I don't know the syntax to do it.
ProgrammerHumor.io

WHAT IS PROGRAMMING LANGUAGE?



COMPILATION PROCESS IN C++



WRITE YOUR OWN FIRST PROGRAM!!

Slido - https://app.sli.do/event/ibNyCP623eGzDRz1G4fu2V

WHAT DOES “ USING NAMESPACE STD; ” MEAN?

In C++ programming, the `using namespace std;` statement is used to indicate that you want to use the standard namespace in your program.

The C++ standard library provides a collection of functions and data types that are commonly used in C++ programming. These functions and data types are defined in the `std` namespace to avoid naming conflicts with other libraries and user-defined types and functions.

When you write `using namespace std;`, you are telling the compiler that you want to use the names of the functions and data types in the `std` namespace without having to prefix them with `std::`. For example, instead of writing `std::cout << "Hello, world!" << std::endl;`, you can write `cout << "Hello, world!" << endl;` after including the necessary headers.

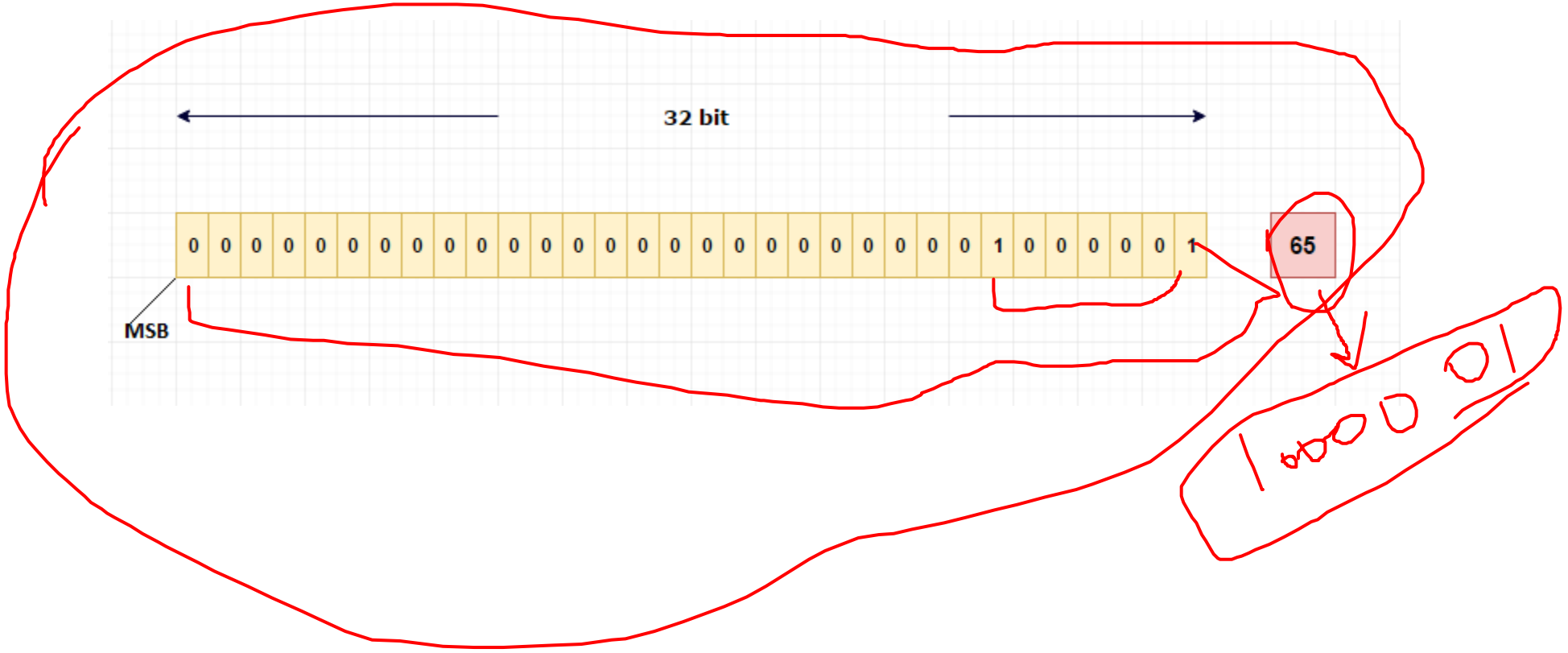
It's important to note that using `using namespace std;` can potentially introduce naming conflicts if there are functions or data types in other namespaces with the same name as those in the `std` namespace. To avoid this, some programmers prefer to explicitly prefix the functions and data types with `std::` to make it clear which namespace they are coming from.

DATATYPES & VARIABLES

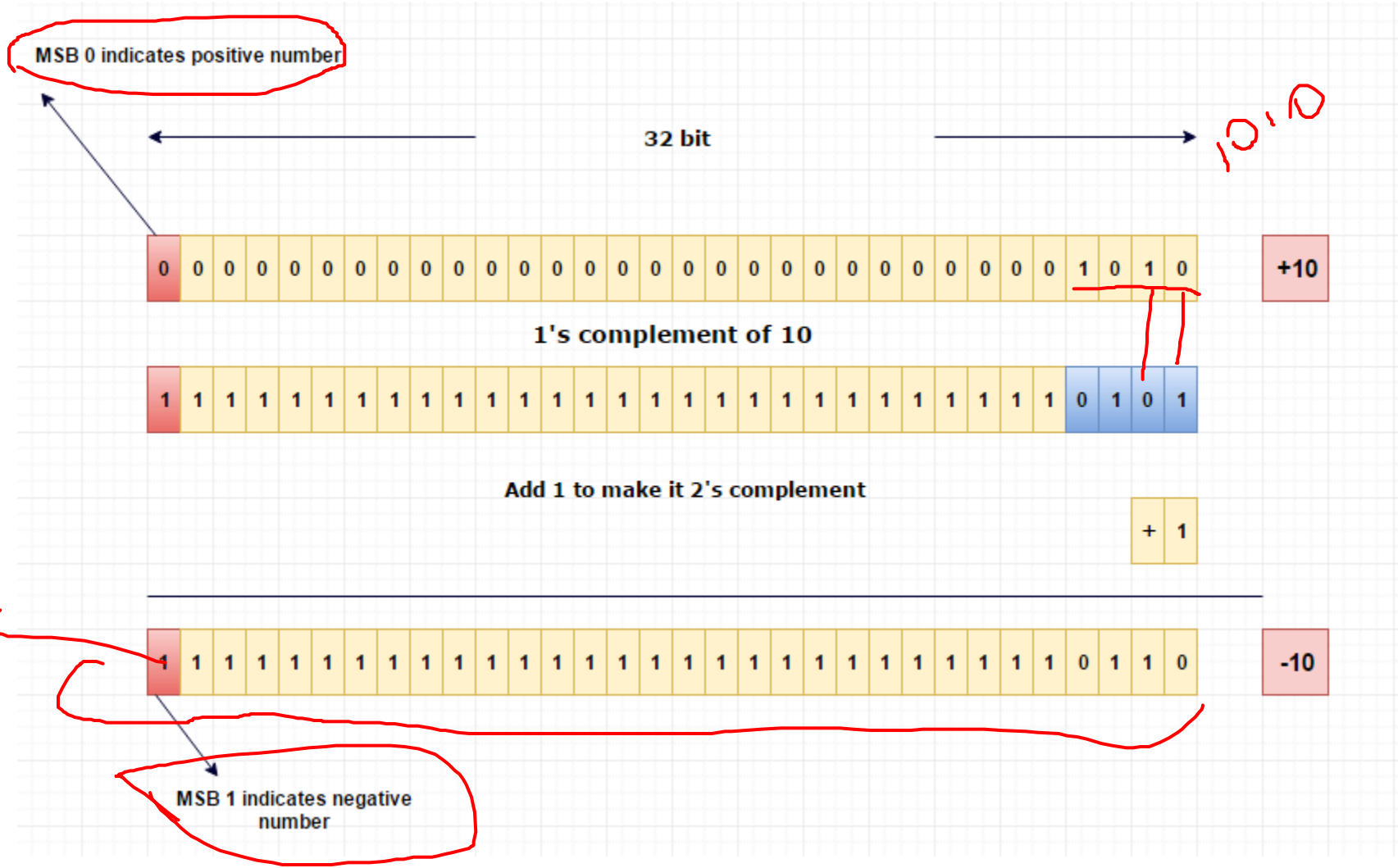
A data type is a classification of data that defines the type of value a variable can hold, the operations that can be performed on it, and the memory space required to store it.

A variable is a named storage location in a computer's memory that is used to store a value that can be accessed and modified by a program. Variables can hold different types of data, such as integers, floating-point numbers, characters, strings, and Boolean values.

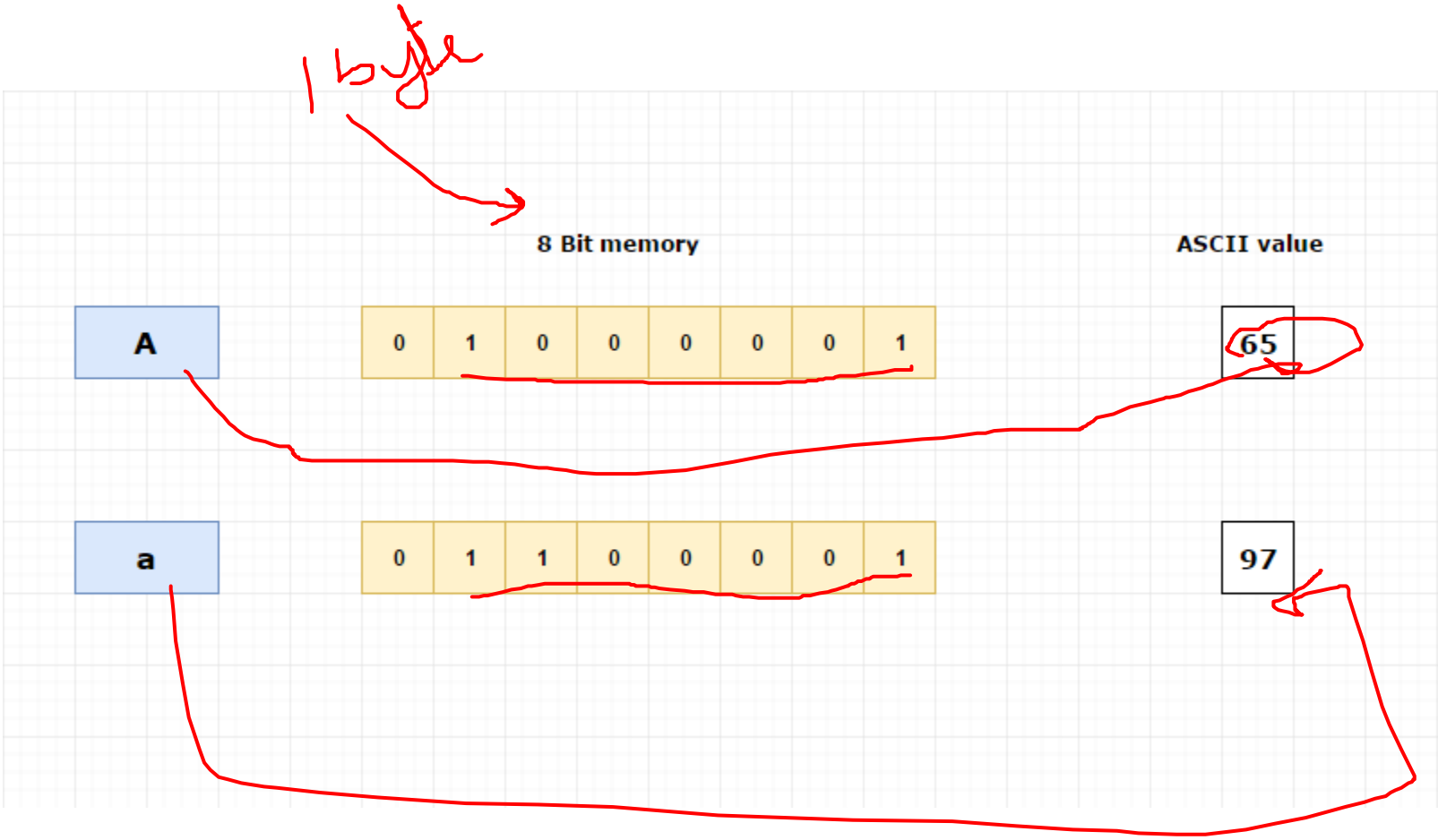
HOW POSITIVE INTEGERS ARE STORED IN MEMORY?



HOW NEGATIVE INTEGERS ARE STORED IN MEMORY?



HOW CHARACTERS ARE STORED IN A MEMORY?



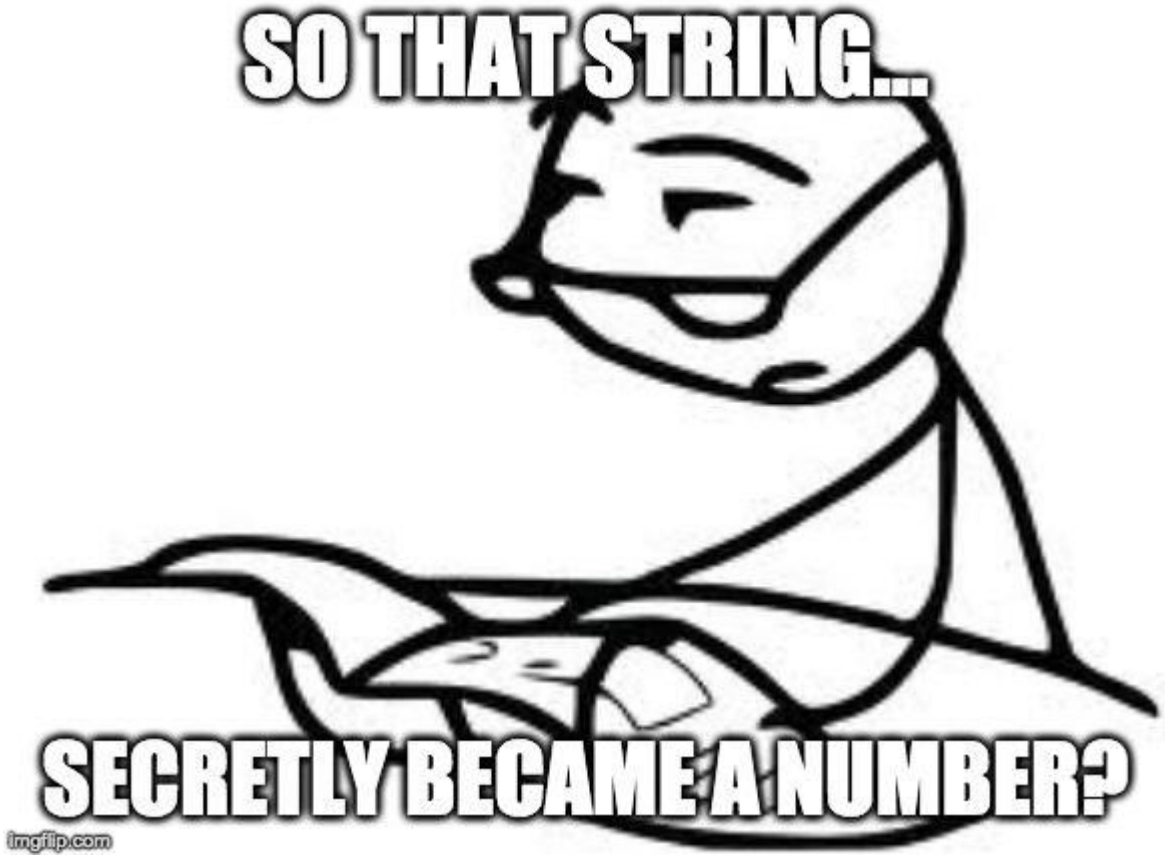
TYPECASTING

Typecasting, also known as type conversion, is the process of converting a value from one data type to another in C++. Typecasting can be either implicit or explicit.

Implicit typecasting occurs automatically when the compiler converts a value from one data type to another in certain situations. For example, if you assign an integer value to a floating-point variable, the compiler will automatically convert the integer to a floating-point value.

Explicit typecasting, on the other hand, requires the programmer to explicitly specify the conversion using a cast operator.

Example of Implicit Type Casting:-



Example of Explicit Type Casting:-