Goal

In this project, you will employ bitwise operations and bitmasks for manipulating bits of an integer. The goal is to understand their significance in saving memory costs. This will be demonstrated and executed through the task of representing weapon inventory in a video game.

**Checkpoint A**

**In this phase of the project, you will set up a weapon inventory system for an assassin in a video game. You will define certain functions to check and display which weapons the assassin possesses.**

Your assassin has the choice of these weapons: *None*, *Melee*, *Fire*, *Ice,* and *Poison*. The assassin may not have any weapons or have multiple weapons at the same time.

* Write an **enum**erated data type called **Weapon** that stores each of the 5 weapon types as a **bitmask**, giving them values of None = 0, Melee = 1, Fire = 2, Ice = 4, and Poison = 8.

Your program will store weapon inventory (that the assassin possesses at any given time) in an integer of type **uint8\_t** (why is 8-bit integer sufficient?). Let's call a variable (declared to be of type **uint8\_t**) as *myweapon*.

* Based on the value of your bitmasks in enum Weapon, declare an 8-bit constant integer *maxWeapons* that stores the maximum value that *myweapon*can get (i.e. when all the weapons are possessed)*. Hint: All the weapons are possessed when the bit corresponding to each weapon type is set. (Is that AND of the bitmasks or OR of the bitmasks?)*

*Functions*

* Write a boolean function *checkWeapon* that checks if a specific weapon exists in the inventory or not. It has the following definition:

bool checkWeapon(uint8\_t currentWeapon, **Weapon** weapon) //checks if weapon is part of currentWeapon or not

where *currentWeapon* is the variable corresponding to the current weapon inventory and *weapon* (of type Weapon) represents the specific weapon type (example: **Weapon**::Melee). It returns *True* if the *weapon* is present, otherwise *False*. For example:

checkWeapon(0b0001010, **Weapon::Melee**) returns False  
checkWeapon(0b0001010, **Weapon::Fire**) returns True  
checkWeapon(0b0001010, **Weapon::None**) returns False

* Write a function *displayWeapon* that prints out all the weapons in the current inventory (including no weapon if that is the case). It must call *checkWeapon* function for each of the **Weapon** types. It has the following definition:

void displayWeapon(uint8\_t currentWeapon)

For example:

displayWeapon(0b0001010)

prints

Your Weapons: Fire Poison

**Main function**.

* In the main function, randomly initialize value of *myweapon* to lie between 0 and *maxWeapons*and print it(see sample output below).
* Print out the binary representation of *myweapon. Y*ou may use the standard library *std::bitset* for this purpose.
* Call the *displayWeapon* function to print out the weapons currently in the inventory, as represented by the integer *myweapon.***Your output must match the sample output exactly.**

**Sample Output for CheckPoint A (note: value for***myweapon***is generated randomly)**

**Starting with myweapon = 0  
Binary representation: 00000000**  
**Your Weapons: None**

**Starting with myweapon = 5  
Binary representation: 00000101  
Your Weapons: Melee Ice**

**Starting with myweapon = 8  
Binary representation: 00001000**  
**Your Weapons: Poison**

**Starting with myweapon = 11  
Binary representation: 00001011**  
**Your Weapons: Melee Fire Poison**

**Starting with myweapon = 15  
Binary representation: 00001111**  
**Your Weapons: Melee Fire Ice Poison**