Workshop CS

Requirements:

To compile and run C#:

- On linux:

```
sudo dnf/apt install mono-update
mcs -out:a.exe test.cs
mono a.exe
```

- On Windows:
  - https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-console?view=vs-2022
  - Stop at step 4 and select .NET 6.0

Exercise n°0: Hello World!

What would be learning a new language without starting with a 'Hello World!' ?

```
using System;

class Program
{
    static void Main()
    {
        string a = "Hello World!"
        // Print a to the console
    }
}
```

Exercise n°1: Simple Concatenate

Create a simple program to :

- Takes two string as a **user input**.
- Prints the **length** of the **two concatenated strings**.

```
using System;

class Program
{
    static void Main(string[] args)
    {
        string a;
        string b;
        string result;

        // Get user input

        // Print length of a+b
    }
}
```

> Hint : Remember to use google :)

Exercise n°2: Introduction to classes

You did a great job for now, but what you did could have been done in C, or any other languages. The difference with C#, is you can implement ✨classes✨.

Classes are used to define *Objects*.

>> "But what's an object ????"

Let's take an example, with a *player* in a video game.

A player has health points, xp, a username, etc. In other words, individual players all have the same attributes, but theses attributes are not the same for each player.

Here is an example of a class:

```
class Player
{
    public string username;
    public int health;
    public double xp;
}
```

What you have to do:

Complete the following code to:

- Create a new player called "Francis", with 20 health points, and 10.0 xp.
- Print the attributes of the player: Username, health, xp.

```
using System;

class Player
{
    public Player(string username, int health, double xp)
    {
        this._username = username;
        this._health = health;
        this._xp = xp;
    }

    private string _username;
    private int _health;
    private double _xp;
```

```
}

class Program
{
    static void Main()
    {
        // Create Player here
    }
}
```

> Hint : If you are struggling, maybe there is something missing in the Player class, linked to "private"...