



First, do some loop





Exercice - 1

- 📌 Écrire un programme qui comptera de 1 à 10 en utilisant une `for` loop.
- 📌 Écrire un programme qui comptera de 10 à 1 en utilisant une `while` loop.



Exercice - 2

📌 Écrire un programme qui demandera à l'utilisateur un input qui soit un nombre.

Il calculera la somme des nombres de 0 au nombre choisi par l'utilisateur et le retournera dans le terminal.

Voici les premières lignes de votre code :



Exercise - 2

```
int userNumber = Convert.ToInt32( Console.ReadLine());

if (userNumber <= 0) {
    Console.WriteLine("Please enter a positive number");
}
```



Tester Exercice - 2



Entrée : 1

Retour : 1



Entrée : 3

Retour : 6



Entrée : 1024

Retour : 524800



Entrée : 65432111

Retour : 187871080



Exercice - 3

Écrire un programme qui génère un nombre aléatoire entre 1 et 100, et demande à l'utilisateur de trouver le nombre choisi.

Le programme doit utiliser une boucle `while` pour continuer à demander à l'utilisateur d'essayer de trouver le bon nombre.

À chaque réponse de l'utilisateur, le programme devra nous dire si le nombre à deviner est plus petit ou plus grand. Il s'arrêtera si l'utilisateur trouve le bon nombre.

Voici un peu de code pour vous aider :



Exercice - 3

```
int randomNum = new Random().Next(1, 100); // genere un nombre aleatoire entre 1 et 100  
  
int UserResponce = Convert.ToInt32(Console.ReadLine()); // demande a l'utilisateur un input qui sera  
// converti en nombre
```



Exercice - 4

Réécrire le même programme que dans l'exercice 3.

Cette fois utilisez une loop `for` et l'utilisateur n'aura que 7 essais pour deviner le nombre.

Dans le cas où il n'y arrive pas, le programme renverra le nombre à trouver et se stoppera.



Exercice - 5

Écrivez un programme qui calcule la factorielle d'un nombre entré par l'utilisateur. Le programme doit utiliser une boucle `for` pour calculer la factorielle.

Par exemple, si l'utilisateur entre 5, le programme doit produire 120 ($5! = 5 \times 4 \times 3 \times 2 \times 1$).



Exercice - 6

Écrivez un programme qui calcule la somme des nombres pairs compris entre 1 et un nombre entré par l'utilisateur.

Le programme doit utiliser une boucle `while` pour calculer la somme.

Par exemple, si l'utilisateur entre 10, le programme doit produire 30 ($2 + 4 + 6 + 8 + 10$).



Exercice - 7

Écrire un programme qui renvoie le n-ième élément de la suite de Fibonacci, cette elements sera donner par l'utilisateur a l'entrée du programme.

La suite commence par 0, 1, 1, 2, 3, 5, etc.



Exercice - 8

Écrivez un programme qui demande à l'utilisateur de saisir une série de nombres séparés par des virgules, puis calculer la moyenne de ces nombres.

Le programme doit utiliser une boucle `for` ou `foreach` pour itérer sur les nombres de la chaîne d'entrée, et calculer la moyenne.

Par exemple, si l'utilisateur saisit "1, 2, 3, 4, 5", le programme doit produire $3 \left((1 + 2 + 3 + 4 + 5) / 5 \right)$.

Tips below :



Exercice - 8 Tips

- 📌 Les string possèdent une méthode pour séparer une chaîne de caractère en tableau de chaînes de caractères :

```
string exemple = "bonjour les amiches, j'espert que tout ce passe bien";  
string[] tableau_de_mot = exemple.Split(' '); // je split sur un espace ici
```

- 📌 Les tableaux ont aussi de méthodes, on peut notamment connaître la taille d'un tableau grâce à l'une de ces méthodes.



Exercice - 9

Écrivez un programme qui demande à l'utilisateur de saisir une série de nombres séparés par des virgules, et renvoie la liste de nombres triés.

L'objectif étant d'utiliser une des boucle(s) et non une méthode d'Array.



Exercice - 10

From string to int.

Écrivez un programme qui demande a l'utilisateur de saisir un nombre entre -32768 et 32767.

Transformer ce nombre en binaire et affichez le.

Ici on exprimera le chiffre sous forme de deux octets (2 fois 8 bits).

Le premier bit sert a définir le signe.



Exemple - 10

- 📌 -1 : 1000 0000 0000 0001
- 📌 -10 : 1000 0000 0000 1010
- 📌 1024 : 0000 0100 0000 0000
- 📌 32767 : 0111 1111 1111 1111