

# TP C++ : Structures répétitives

Dans ce nouveau TP nous allons utiliser les structures répétitives vues en cours telles que : `while()`, `do ... while` et `for()`.

Vous trouverez sur classroom, un dossier appelé "TP03\_structuresRepetitives" qui est le projet CLion à compléter. Téléchargez le zip, dézippez-le, ouvrez le dossier projet avec CLion et commencez à coder.

## 1 Somme infinie ?

Dans cet exercice, il s'agit de faire la somme des entiers naturels, soit la somme des nombres entiers positifs : 1,2,3,4 ...

Votre programme demandera à l'utilisateur de saisir un entier `n` qui représentera le nombre d'entiers naturels à additionner et stockera le résultat dans la variable `somme` avant de l'afficher sur la console.

Par exemple avec `n=5` :

$$somme = 1 + 2 + 3 + 4 + 5$$

1. En C++, déclarez et initialisez à 0 les variables qui contiendront la somme des entiers `somme` et le nombre d'entiers voulu `n`.
2. Afficher sur la console la demande d'un entier à l'utilisateur et gérez la capture de cet entier au clavier pour le stocker dans `n`.
3. Créez une structure `for()`, qui permettra de tourner dans la boucle le nombre de fois désigné par `n`.
4. Dans la boucle `for` trouvez l'instruction permettant d'additionner la somme précédente avec l'entier actuel et de mettre le résultat dans `somme`.
5. Pour terminer affichez `somme` à l'écran.

## 2 Multiples et non multiples

Combien y a-t-il d'entiers entre 1 et 20000 qui sont multiples de 7, mais pas multiples de 9 ? (le nombre à trouver est 2540 !)

1. En C++ déclarez et initialisez à 0 la variable de comptage `comptage` qui contiendra l'entier recherché.
2. Créez une structure `for()`, qui permettra de tourner dans la boucle le nombre de fois désiré.
3. Dans la boucle `for` ajouter une structure `if` afin de faire le test demander.
4. Si le test est positif incrémentez la variable de codage `comptage`.

### 3 Carnet de notes

Afin de vérifier que Pronote ne s'est pas trompé, Alice souhaiterait avoir une application de calcul de moyenne. Aidez-là.

1. En C++ déclarez et initialisez à 0 le nombre de variables dont vous avez besoin.
2. Créez une structure `while()`, qui permettra de redemander à l'utilisateur de rentrer une note tant qu'il le souhaite.
3. Dans la boucle, afficher sur la console la demande d'une note à l'utilisateur et gérez la capture de ce nombre au clavier pour le stocker dans une variable. Faites de même pour demander le coefficient de cette note.
4. Toujours dans la boucle, faites les calculs dont vous avez besoin pour calculer la moyenne.
5. A la fin, affichez la moyenne calculée.

### 4 Calcul de factorielle

La factorielle d'un nombre est le produit de tous les entiers à partir de 1 jusqu'à ce nombre. La factorielle peut seulement être définie pour des **entiers positifs**.

La factorielle d'un nombre négatif n'existe pas et la **factorielle de 0 est 1**.

Par exemple : la factorielle de 5 est définie par l'expression mathématique  $5!$  et vaut :

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

Créez un programme qui demande à l'utilisateur de saisir un entier  $n$ , qui vérifiera si  $n \geq 0$  et calculera sa factorielle, dans le cas contraire un message d'erreur sera affiché et il sera demandé à l'utilisateur de saisir un nouveau nombre. Le résultat du calcul sera affiché sur la console.

### 5 Suite de Fibonacci

On considère la suite de nombres entiers dont les deux premiers termes sont :

$$F_0 = 1 \quad F_1 = 1$$

Les termes suivants de cette suite sont construits de la manière suivante :

$$F_n = F_{n-1} + F_{n-2}$$

Écrire un code qui à partir d'un entier  $n \geq 0$  affiche le  $n$ -ième terme de la suite.

Par exemple selon la valeur de  $n$  :

$$F_3 \rightarrow 3$$

$$F_{11} \rightarrow 89$$

$$F_{16} \rightarrow 987$$