

---

## TP1 : Du C au C++

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

---

Créer un projet TP1 de type "C++ executable" dans lequel il est demandé de créer pour chaque exercice un fichier ".cpp" contenant les fonctions et un programme principal "main()"

### Exercice 1

Créer et tester un programme C++ qui demande à l'utilisateur de saisir deux valeurs (val1 et val2) et puis calcule leur somme et affiche le résultat comme suit :

*"val1 + val2 = resultat"*

Si le résultat est positif le message suivant est aussi affiché *"Positif ? true"* sinon l'affichage sera *"Positif ? false ou nul"*. "false" et "true" doivent être le contenu de deux variables différentes.

### Exercice 2

Créer et tester une fonction C++ qui :

- demande à l'utilisateur d'entrer une taille (n) d'un tableau (utiliser le mot-clé `auto` pour déclarer "n")
- alloue un espace mémoire pour ce tableau si la taille est positive

- c. demande à l'utilisateur d'entrer "n" valeurs et les affiche
- d. calcule la moyenne des "n" valeurs et l'affiche
- e. libère la mémoire allouée par le tableau en fin de programme

### Exercice 3

Ecrire et tester 3 fonctions qui ont le même nom à savoir `max` et qui calculent le maximum des variables données comme arguments :

1. Prototype de la première fonction `max(int n)` dans ce cas c'est la valeur de "n" qui va être retournée
2. Prototype de la deuxième fonction `max(int m, int n)`
3. Prototype de la troisième fonction `max(int m, int n, int r)`

### Exercice 4

Programmer et tester en C++ deux fonctions permettant d'échanger les contenus de 2 variables de type int fournies en argument :

- **1ère version** : la transmission des variables concernées est faite par adresse (valable en C et C++)
- **2ème version** : la transmission des variables concernées est faite par référence (valable en C++)

### Exercice 5

1. Écrire une fonction `sum(n)` qui renvoie la somme des entiers de 1 à n.

Exemple d'utilisation :

```
int i = 6;  
int s = sum(i); // s contient 21
```

2. Modifier le prototype de la fonction pour qu'elle effectue la somme de  $k$  à  $n$ . Si  $k$  n'est pas précisé, la valeur  $k = 1$  sera utilisée par défaut.

Exemple d'utilisation :

```
int i = 6;  
int s1 = sum2(i); // s1 contient 21  
int s2 = sum2(i, 4); // s2 contient 15
```

3. Modifier la fonction pour qu'elle ne renvoie plus rien et stocke le résultat dans un entier donné en argument.

Exemple d'utilisation :

```
int i = 6;  
int s = 0;  
sum3(i, s); // s contient 21  
sum3(i, s, 4); // s contient 15
```

4. Modifier la fonction pour qu'elle ne modifie pas  $s$  si  $k$  a une valeur incorrecte.

Exemple d'utilisation :

```
int i = 6;  
int s = 0;  
sum4(i, s, 42); // s contient 0  
sum4(i, s, 4); // s contient 15
```

5. Dans certains cas, on ne peut pas savoir si une modification a bien eu lieu ou non. Par exemple :

```
int i = 6;  
int s = 15;  
sum5(i, s, 42); // s contient 15 (non modifié)  
sum5(i, s, 4);  // s contient 15 (modifié)
```

Modifier l'implémentation de `sum()` pour que l'on puisse distinguer ces deux cas.