

## TP3 – Syntaxe de base du C++ : les tableaux

---

### **Exercice 1 : allocation statique / allocation dynamique**

On veut créer un tableau d'entier, le remplir avec des valeurs entières puis on affichera le nombre de valeurs entières positives. Pensez à diviser le problème en fonctions.

Observez votre code et optimisez le si c'est possible en termes de temps d'exécution.

### **Exercice 2 :**

Écrire une fonction *remplir* qui permet de réserver et de remplir un tableau de N entiers.

Écrire une fonction *recherche* qui recherche une valeur entière V dans un tableau. La fonction doit rechercher si V se trouve dans le tableau et afficher « V se trouve dans le tableau à la position i » ou « V ne se trouve pas dans le tableau ».

Écrire un programme principale *main* qui exploite les fonctions précédemment définies. (Donnez une version avec les tableaux bruts, puis une version avec l'objet `std::array` )

Reprendre le même problème et le résoudre à l'aide du conteneur `vector`.

### **Exercice 3 :**

Ecrire un programme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

*Exemple :*

$U=\{1,2,3\}$  et  $V=\{4,5,6\}$  donc  $U \cdot V = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$

### **Exercice 4 : fonctions retournant plusieurs valeurs (`std::pair`, `std::tuple`)**

Ecrire une fonction *mini\_maxi* qui retourne la plus grande et la plus petite valeur dans un tableau d'entiers A.

Dans la fonction *main*, afficher les valeurs retournées par la fonction *mini\_maxi*. Pour remplir le tableau utiliser les fonctions définies dans les exercices précédents.

**Exercice 5 : tableau dynamique / map**

Soit la structure ville définie par un nom et une position (longitude et latitude). On veut gérer l'ensemble des villes de la Tunisie et créer un programme qui recherche la ville par son nom et affiche sa position.

Donnez une version avec les vector puis une autre version avec unordered\_map. Comparer les performances de chaque méthode.

**Exercice 6 : bibliothèque**

Reprendre les fonctions de l'exercice précédent et penser à ajouter d'autres fonctions sur les tableaux (afficher, trier, ...) puis construire une bibliothèque nommée « tableaux.h » qu'on pourra utiliser à chaque fois qu'on a affaire aux tableaux.