

---

## SÉANCE 4

---



### Objectif

Le but de cette séance est de manipuler les fonctions et le passage de paramètres.



### Exercices

#### ✎ Exercice 1 (Plus petit et plus grand éléments d'un tableau d'entiers)

Pour toutes les fonctions ci-dessous, on suppose que le tableau contient au moins un élément.

Vous veillerez à **tester les fonctions au fur et à mesure** à l'aide d'une fonction principale.

1. Écrivez la fonction d'en-tête :  
`void AfficherTab(int Tab[], int NbElt)`  
qui affiche tous les entiers d'un tableau sur la même ligne, avec un espace entre chaque entier (et un retour à la ligne après l'affichage).  
Paramètres :
  - `Tab` : tableau de `int` ;
  - `NbElt` : nombre d'éléments du tableau `Tab`.
2. Écrivez la fonction d'en-tête :  
`int Min(int Tab[], int NbElt)`  
qui retourne le plus petit élément d'un tableau d'entiers.  
Paramètres : identiques à la fonction précédente.
3. Écrivez la fonction d'en-tête :  
`int Max(int Tab[], int NbElt)`  
qui retourne le plus grand élément d'un tableau d'entiers.  
Paramètres : identiques à la fonction précédente.
4. Écrivez la fonction d'en-tête :  
`void MinMax(int Tab[], int NbElt, int *pMin, int *pMax)`  
qui délivre en sortie le plus petit et le plus grand éléments d'un tableau d'entiers.  
Paramètres :
  - `Tab` : tableau de `int` ;
  - `NbElt` : nombre d'éléments du tableau `Tab` ;
  - `pMin` : adresse où doit être stocké le plus petit élément ;
  - `pMax` : adresse où doit être stocké le plus grand élément.
5. Écrivez la fonction d'en-tête :  
`int SupprimerVal(int Tab[], int *pNbElt, int Val)`  
qui supprime toutes les occurrences d'une valeur dans un tableau d'entiers. Le nombre de suppressions sera retourné par la fonction et le nombre d'éléments du tableau sera mis à jour.  
Paramètres :
  - `Tab` : tableau de `int` ;
  - `pNbElt` : adresse pointant vers le nombre d'éléments du tableau ;
  - `Val` : valeur à supprimer.

## ☞ Exercice 2 (Conversion de durées)

L'objectif de cet exercice consiste à écrire deux fonctions permettant de convertir une durée exprimée en secondes en la même durée exprimée en heures, minutes et secondes. Par exemple, 3666 secondes représentent 1 heure 1 minute et 6 secondes.

Pour tester vos deux fonctions, écrivez une fonction principale qui demande à l'utilisateur de taper une durée en seconde, la lit, appelle chacune des deux fonctions et affiche les résultats. Aucune entrée au clavier ni sortie à l'écran ne sera donc réalisée par les fonctions de conversion.

Les informations en entrée et en sortie des deux fonctions doivent passer par leurs paramètres selon le schéma suivant :

1. la première fonction doit prendre en entrée la durée en secondes et délivrer en sortie le nombre d'heures, le nombre de minutes et le nombre de secondes (4 paramètres en tout) ;
2. la seconde fonction doit prendre en entrée la durée en secondes et délivrer en sortie le résultat sous la forme d'un tableau à 3 cases, la première contenant le nombre d'heures, la deuxième le nombre de minutes et la troisième le nombre de secondes (2 paramètres en tout).



## Pour aller plus loin

Écrivez une variante de la fonction `MinMax()` de l'exercice 1 qui, au lieu de délivrer en sortie le plus petit et le plus grand éléments, délivre en sortie les **adresses** du plus petit et plus grand éléments dans le tableau passé en paramètre (aide : pour cela il vous faudra commencer par modifier l'en-tête de la fonction).