

# TP C++ 3 : Structures

M1 Mathématiques Appliquées

2019-2020



## Exercice 1 : Structures et fonctions

On considère la structure suivante qui permet d'encoder les nombres complexes :

```
struct complexe{  
    float re_part, im_part;  
};
```

On demande d'implémenter et de tester les deux fonctions suivantes :

- Une fonction prenant deux structures `complexe` en argument et retournant (par valeur) une structure `complexe` contenant la somme des deux nombres complexes;
- Une fonction prenant en argument un réel (flottant) ainsi qu'un pointeur sur une structure de type `complexe`, qui modifie le complexe contenu dans la structure en le multipliant par le réel.

## Exercice 2 : Structure, fonction et tableau

En utilisant le code développé dans l'exercice précédent, écrire et tester une fonction prenant en entrée un tableau de structures de type `complexe`, ainsi que sa taille, et retournant une structure de type `complexe` correspondant à la somme des éléments de ce tableau.

## Exercice 3 : Structure contenant des tableaux

On considère le type structure `enreg` défini comme suit :

```
const int NMOIS = 12;  
  
struct enreg{  
    int stock;  
    float prix;  
    int ventes[NMOIS];  
};
```

Écrire une fonction qui affecte la valeur zéro aux champs `stock` et `ventes` d'une valeur de type structure `enreg` passée en argument. Cette fonction ne devra pas renvoyer d'argument en sortie.

Écrire un programme de tests qui permette de visualiser l'effet de cette fonction.

## Exercice 4 : Structure de structure

On définit deux structures `date` et `personne` :

```
const int LMAX_NOM=30;
```

```
struct date{
    int jour;
    int mois;
    int annee;
};
```

```
struct personne{
    char nom[LMAX_NOM];
    date date_inscr;
};
```

- a) Écrire alors une fonction `inscr_personne` prenant en argument une structure de type `personne` et remplissant les différents champs de cette structure en demandant à l'utilisateur d'en entrer les valeurs.
- b) Modifier la structure `personne` de sorte à ce que le tableau `nom` soit alloué dynamiquement. Modifier ensuite la fonction `inscr_personne` afin que celle-ci lise un nom de taille maximum `LMAX_NOM`, puis alloue dynamiquement le champ `nom` de sorte à ce qu'il soit de taille minimale.