

TD1 : Allocation Dynamique de Mémoire & Enregistrements

Exercice N° 1 :

1. Ecrire une fonction « **float moyenne(float *T, int taille)** » qui calcule et retourne la moyenne d'un tableau **T**.
2. Écrire un programme qui :
 - a) Demande **n** le nombre d'étudiants,
 - b) Alloue dynamiquement un tableau de **n** entiers,
 - c) Saisit les **n** notes et affiche la moyenne,
 - d) Demande **m** (nouveaux étudiants) et **agrandit** le tableau à **n+m**,
 - e) Ré-affiche les notes et la nouvelle moyenne,
 - f) Libère la mémoire avec **free**.

Exercice N° 2 :

On souhaite gérer dynamiquement un tableau d'entiers **trié en ordre croissant**.

1. Écrire un programme qui :
 - demande à l'utilisateur le **nombre initial d'éléments** **n**,
 - alloue dynamiquement un tableau d'entiers de taille **n**,
 - lit les **n** entiers **déjà triés** par l'utilisateur,
 - affiche le tableau.
2. Écrire une fonction qui permet d'insérer un nouvel élément, la fonction doit :
 - augmente la taille du tableau de 1 (avec **realloc**),
 - insère la nouvelle valeur **val** **à la bonne position** pour que le tableau reste trié,
 - met à jour **n**.
3. Dans le programme principal :
 - demander à l'utilisateur une valeur à insérer,
 - appeler la fonction **insérerElement**,
 - afficher le tableau mis à jour (toujours trié).

Exemple d'exécution attendue :

```
Entrez le nombre d'éléments : 5
Entrez les 5 valeurs triées : 2 8 15 23 40
Tableau initial : 2 8 15 23 40
Entrez la valeur à insérer : 18
Tableau après insertion : 2 8 15 18 23 40
```

Exercice N° 3 :

Nous voulons développer une petite application pour gestion des étudiants.

1. Déclarer une structure **Etudiant** pour modéliser un étudiant (nom, age, note).
2. Créer une fonction « **Etudiant* lireEtudiant()** » qui permet de lire et retourner un étudiant.
3. Ecrire une fonction « **void afficherEtudiant(Etudiant etu)** » qui affiche les informations d'un étudiant passé en paramètre.
4. Ecrire une fonction « **float moyenne(Etudiant *et, int n)** » qui calcule et retourne la moyenne des **n** étudiants passés en paramètres.
5. Dans la fonction main(), demander à l'utilisateur le nombre **n** d'étudiants, allouer dynamiquement un tableau de **n** étudiants, lire leurs informations, puis afficher la liste et la moyenne générale.

Exercice N° 4 :

Écrire un programme en C qui :

1. Définit deux structures : une structure **Adresse** contenant : ville, rue, codePostal et une structure **Employe** contenant : nom et un champ adr de type Adresse.
2. Ecrire une fonction « **Employe* lireEmploye** » qui lit les informations d'un employé (nom et son adresse complète).
3. Affiche toutes les informations de l'employé à l'aide d'une fonction « **void afficherEmploye (Employe e)** ».
4. Une fonction **main** pour lire **n** employés et les afficher sur écran.