SÉRIE N° 6 : ALLOCATION DYNAMIQUE DE LA MÉMOIRE

EXERCICE 1: QCM

1. Quelles sont les étapes à suivre dans l'ordre lorsqu'on alloue dynamiquement de la mémoire ?

- malloc, vérification de la validité du pointeur, utilisation de la mémoire, free
- malloc, utilisation de la mémoire, free, vérification de la validité du pointeur
- free, vérification de la validité du pointeur, malloc, utilisation de la mémoire

2. Que se passe-t-il si je fais l'opération suivante ?

```
malloc(25*sizeof(int));
```

- Cela réserve de la mémoire pour un int de 25 octets
- Cela réserve de la mémoire pour un tableau d'int de 25 cases
- Cela réserve de la mémoire pour un float de 25 octets
- Cela réserve de la mémoire pour un tableau de float de 25 cases

EXERCICE 2:

Ecrire une fonction de saisie des éléments d'un tableau de réels. Le nombre d'éléments du tableau sera donné en argument de la fonction qui retournera l'adresse du tableau. Ce tableau devra être alloué dynamiquement.
Une gestion des erreurs permettra de renvoyer le pointeur NULL en cas d'erreur d'allocation mémoire.

EXERCICE 3:

Ecrire la fonction **void lib_tab(float *T,int n)** qui affiche puis libère la mémoire d'un tableau **T** alloué dynamiquement.

EXERCICE 4:

Ecrire d'une manière dynamique la fonction **char* concatener(char *s,char *t)** qui permet de rajouter la chaîne t à la fin de la chaîne s et de mettre le résultat dans une troisième chaîne de caractères **res** .
La fonction retourne la nouvelle chaîne de caractères **res.**

EXERCICE 5:

1. Définir la structure **Complexe**.
2. Ecrire la fonction permettant de saisir un nombre complexe.
3. Ecrire la fonction permettant d'afficher un nombre complexe.
4. Ecrire la fonction permettant de créer dynamiquement un tableau T pour stocker N nombres complexes (N sera donné en argument de la fonction).
5. Ecrire la fonction permettant de remplir le tableau T
6. Ecrire la fonction permettant d'afficher le tableau T.

Tester ces fonctions dans un petit programme.