

TP N°: 1

Exercice1 : -----

Soit la structure pile (qui respecte le principe de fonctionnement **LIFO** : *Last Input First Output*) contenant deux champs (un tableau des entiers **tab** et un entier **sommet**) :

```
#define max 100
typedef struct{
    int tab[max] ;
    int sommet;
} Pile;
```

On se donne trois piles **P1**, **P2** et **P3**. La pile **P1** contient une suite de nombres entiers positifs.

- 1) Ecrire les fonctions suivantes:
 - a) *InitialiserPile(Pile *P)* : permet d'initialiser la pile **P**.
 - b) *int PileVide(Pile P)* : retourne 1 si la pile **P** est vide et 0 sinon.
 - c) *int PilePleine(Pile P)* : retourne 1 si la pile **P** est pleine et 0 sinon.
 - d) *void PilePush(Pile *P, int e)* : empile l'élément **e** dans la pile **P**.
 - e) *int PilePop(Pile *P)* : retire et retourne un élément de la pile **P**.
 - f) *int PileSommet(Pile P)* : retourne le sommet de la pile **P**.
- 2) Ecrire une fonction *Deplacer()* pour déplacer tous les éléments de **P1** dans **P2** de façon à avoir dans **P2** tous les nombres pairs au-dessus des nombres impairs. Penser à utiliser **P3** comme pile auxiliaire.
- 3) Écrire la fonction **main** pour tester les différentes fonctions.

Exercice2 : -----

Une file est une liste linéaire qui respecte le principe de fonctionnement **FIFO** (*First Input First Output*). En se basant sur la gestion par décalage, on considère que la file est représentée par une structure comme suit :

```
typedef struct{
    int tab[100] ;
    int ar; // arrière ou queue de la file
} File;
```

On se propose de trier les éléments d'une file par ordre décroissant. Dans ce traitement, il faut utiliser trois files : **F1**, **F2** et **F3**.

F1 contient les entiers à trier, **F2** et **F3** sont des files auxiliaires.

- 1) Donner une implémentation des fonctions :
 - a) *void InitialiserFile(File *F)* : initialise la file **F**.
 - b) *int FileVide(File F)* : retourne 1 si la file **F** est vide et 0 sinon.
 - c) *int FilePleine(File F)* : retourne 1 si la file **F** est pleine et 0 sinon.
 - d) *void Enfiler(File *F, int e)* : insère l'élément **e** dans la file **F**.
 - e) *int Defiler(File *F)* : extrait et retourne un élément de la file **F**.
 - f) *int Retirer_max()* : retire l'élément de valeur maximale de la file **F1**. Penser à utiliser la file auxiliaire **F2**.
 - g) *Void Trier()* : trie la file **F1** dans l'ordre décroissant. Penser à utiliser la fonction précédente *Retirer_max()* et la file auxiliaire **F3**.
- 2) Ecrire la fonction **main** pour tester les différentes fonctions.