

TP1 Structures de données en langage C

Parcours MIPC

-2 séances-

Exercice 1 : *pile et élément avec priorité*

On suppose que tout élément est muni d'une **priorité strictement positive** représentée par un **entier** qui doit être précisé au moment de l'ajout de l'élément dans la **pile**. La fonction **retirer** doit **chercher l'élément le plus prioritaire**. On s'intéresse à une implémentation efficace des opérations (**ajouter** et **retirer**).

Chaque élément de la pile sera défini par la structure de donnée suivante :

```
typedef struct elt {  
    int  priorite ;    // priorite représente la priorité de l'élément de type entier  
    char donnee ;     // donnée porte une valeur de type char  
} element
```

- 1- Donner les structures de données et les variables globales à déclarer pour manipuler la pile.
- 2- Donner une implémentation des fonctions **ajouter** et **retirer** dans les deux cas suivants :
 - a- la fonction **ajouter** range les éléments dans un ordre quelconque. Ce qui implique que la fonction **retirer** doit parcourir la pile pour extraire les éléments jusqu'à elle arrive à l'élément prioritaire puis elle doit rendre les éléments retirés dans le même ordre avant l'extraction.
 - b- la fonction **ajouter** maintient les éléments par ordre de priorité croissante.
- 3- Ecrire la fonction **main** pour tester les différentes fonctions.

Exercice 2 : Gérer une vaisselle sale

Nous voulons modéliser une file constituée de piles d'assiettes sales (5 piles au maximum) dans une cuisine de restaurant. Chaque pile d'assiettes (10 assiettes au maximum) posée au fur et à mesure qu'elles arrivent en cuisine dans une file. Le plongeur nettoie les assiettes en les prenant, une par une, sur le dessus de la première pile stockée.

- En se basant sur les structures de données suivantes :

```
#include<stdio.h>
#define Max_File 5
#define Max_Assiettes 10

// Implémentation statique des piles
typedef struct {
    int PA[Max_Assiettes];
    int Nbre_Assiettes; //(Sommet)
} Pile_Assiettes;
```

```
// Implémentation statique et par décalage de la file
typedef Pile_Assiettes File[Max_File];
int ar;
File F;
```

- Ecrire la fonction permettant d'ajouter une pile d'assiettes.
- Ecrire la fonction permettant au plongeur de retirer une assiette pour la nettoyer.
- Ajouter la fonction **main** qui permet d'appeler les différentes fonctions.

Une serveuse, à des comportements très étranges, amène les assiettes en cuisine, mais lorsqu'elle trouve la file pleine, elle commence de les ajouter sur les piles pour les rendre à leurs maximums (10 assiettes). Dans le cas où la serveuse se trouve à la fin avec un reste d'assiettes, elle les jette à la poubelle.

- Ajouter une fonction qui permet de simuler le travail de la serveuse et de rendre le nombre d'assiettes jetées à la poubelle.
- Ecrire la fonction permettant d'afficher l'état de la file soit avant ou après le passage de la serveuse.
- Réécrire la fonction **main** qui permet de simuler le travail de la serveuse folle.