

TP 2 : Représentation contigüe : Piles

Exercice 1

L'objectif : Réaliser une pile simple en tableau en utilisant la structure suivante :

```
typedef int element;
typedef struct {
    element elements[MAX_PILE];
    int iSommet; /*
} pile;
```

1.1: Mettre en place les fichiers .h et .c nécessaires pour la réalisation en implémentant les fonctions suivantes:

```
pile pile_vide ();
int est_vide ( pile p );
element sommet ( pile p );
pile empiler ( pile p, element e );
pile depiler ( pile p );
```

1.2. Faire un jeu de données dans le main.c avec MAX_PILE = 6 :

- empiler quatre entiers (10, 20,30,40)
- afficher les éléments de la pile
- afficher l'indice du sommet et le contenu du sommet de la pile, quel est le résultat obtenu? le sommet de la pile est 40, l'indice affiché est 3.
- empiler les trois entiers (50, 60, 70) , quelle est la valeur du sommet?
- dépiler six éléments et réafficher le sommet et son indice.
- Dépiler un autre élément et réafficher le sommet et son indice.

Exercice 2

Ecrire un programme qui teste si une expression est bien parenthésée.

$(a+b)$ et $()$ sont des expressions bien parenthésées.

$((a+b)$ et $)a+b$ ne sont pas bien parenthésées.

Le programme main initialise l'expression avec un tableau de caractères :

Ex : `char a[100] = { '(', 'a', '+', 'b', ')', '\0' }`

1. Décrire l'algorithme et mettre en place les fichiers .h et .c nécessaires pour la réalisation.

2. Implémenter dans le main.c le programme qui permet de vérifier si l'expression est bien parenthésée et qui affiche le résultat.