

TP 6 : Les piles

Exercice 1 : Pile statique

On souhaite gérer une pile d'entiers à l'aide d'un **tableau**.

1- Créer une **structure Pile** avec les champs **sommet** (entier) et un tableau d'entiers **tab** de taille **dimension**.

Ajouter les fonctions suivantes pour gérer la pile:

2- **void init_pile (Pile *P)** qui initialise le **sommet** de la pile à 0.

3- **int pile_vide (Pile *P)** qui teste si la pile est vide. Elle retourne 1 dans le cas favorable.

4- **int pile_pleine (Pile *P)** qui teste si la pile est pleine. Elle retourne 1 dans le cas favorable.

5- **int empiler (Pile *P, int x)** qui ajoute l'entier **x** dans la pile. Utiliser la fonction **pile_pleine**. Cette fonction retourne 1 dans la cas d'un empilement réussi et 0 sinon.

6- **void remplir (Pile *P)** qui permet de remplir la pile par **n** entiers. Utiliser la fonction **empiler** précédente pour empiler plusieurs entiers.

7- **int depiler (Pile *P)** qui retire le dernier entier inséré dans la pile. Utiliser la fonction **pile_vide**. Cette fonction retourne l'entier retiré en cas de dépilement réussi et -1 sinon.

8- **void affichage(Pile *P)** qui affiche le contenu de la pile.

9- **int rechercher (Pile *P, int x)** qui recherche l'entier **x** dans la pile. La fonction retourne sa position dans la pile s'il existe et -1 sinon.

10- **void menu(Pile *p)** qui représente un menu textuel qui permet à l'utilisateur de choisir l'opération à réaliser (1) initialiser la pile, 2) remplir la pile, 3) dépiler, 4) afficher la pile, 5) rechercher dans la pile).

11- Ajouter le programme principal qui fait appel au menu.

Exercice 2 : Pile dynamique

Ecrire un programme qui gère une pile d'étudiants à l'aide d'une liste chaînée. Pour cela, créer une 1^{ère} **structure Etudiant** qui contient les champs: **cne** (entier), un **nom** et un **prenom** (pointeur sur un caractère) et une variable **suivant** de type **pointeur** sur la structure **Etudiant**. Aussi, créer une 2^{ème} **structure Pile** avec les champs **sommet** de type **pointeur** sur la structure **Etudiant** et un entier **taille**.

La pile est gérée à l'aide des fonctions suivantes :

1- **void initialiser (Pile *P)** qui initialise le pointeur **sommet** de la pile à NULL et **taille** à 0.

2- **int empiler(Pile *P, int nouveau_cne, char *nouveau_nom, char *nouveau_prenom)** qui ajoute un nouveau étudiant (**nouveau_cne**, **nouveau_nom** et **nouveau_prenom**) à la pile, modifie le pointeur **sommet** et incrémente la **taille**. La fonction retourne -1 si l'empilement a échoué et 0 sinon.

3- **int depiler (Pile *P)** qui retire le dernier étudiant inséré dans la pile, modifie le pointeur **sommet** et décrémente la **taille** de la pile. La fonction retourne -1 si le dépilement a échoué et 0 sinon.

3- **void afficher (Pile *P)** qui affiche le contenu de la pile.

4- Ecrire le programme principal qui fait appel aux fonctions précédentes.

5- Ajouter un menu.