Programmation et Algorithmique II

Lab 6: La Pile

### Exercice 1:

Implémenter en C le type abstrait Pile en utilisant le principe de liste chaînée.

1. Utiliser le fichier .h en-dessous pour implémenter les fonctions primitives de la pile.
2. Tester cette implémentation dans un programme.

Supposant que le fichier .h est le suivant :

#ifndef PILES

#define PILES

#include <stdio.h>

#include <stdlib.h>

typedef float Element;

struct SCellule {

Element info;

struct SCellule \*psuiv;

};

typedef struct SCellule \*Cellule;

struct SPile{

struct SCellule \*sommet;

int nbElements;

};

typedef struct SPile \*Pile;

Pile pileVide(); // creer une pile vide (initialiser)

Pile pileAjouter(Pile p, Element e); //ajouter un élément au sommet de la pile

Pile pileSupprimer(Pile p); //supprimer l’élément au sommet de la pile

Element pileSommet(Pile p); //renvoyer l’élément au sommet de la pile

#endif

### Exercice 2:

On se donne une pile P1 contenant des entiers positifs.

1. Ecrire un algorithme pour déplacer les entiers de P1 dans une pile P2 de façon à avoir dans P2 tous les nombres pairs en dessous des nombres impairs.

Exemple :

1. Ecrire un algorithme pour copier dans P2 les nombres pairs contenus dans P1. Le contenu de P1 après exécution de l’algorithme doit être identique à celui avant exécution. Les nombres pairs dans P2 doivent être dans l’ordre où ils apparaissent dans P1.

Entree :

|  |
| --- |
| 4 |
| 3 |
| 2 |
| 1 |

Sortie :

|  |
| --- |
| 3 |
| 1 |
| 4 |
| 2 |

### Exercice 3:

Un fichier texte peut contenir des parenthèses ( ), des crochets [ ], et des accolades { }. Ces éléments peuvent être imbriqués les uns dans les autres (exemple : {a(bc[d])[{e f }(g)]} ) Écrire une fonction qui parcourt le fichier texte et détermine si le fichier est correctement parenthésé, c’est-à-dire si toutes les parenthèses, crochets, etc. sont bien refermés par un caractère du même type, et si les parenthèses, crochets

et accolades sont correctement imbriqués. Exemple de fichier incorrect : ({)}. Exemple correct : {(())}

### Exercice 4:

Implémenter une fonction, qui peut convertir une expression infixe à postfixe. L’expression est entrée par une chaîne de caractère.

Implémenter une fonction, qui évalue une expression postfixe, entré par une chaine de caractère.

### Exercice 5 (optionelle):

Le but de l’exercice est d’implémenter un jeu de bataille automatique. Chaque carte sera représentée une structure contenant un int allant de 0 à 13, et un char entre 0 et 3 représentants la couleur (pique, carreau, trèfle ou coeur).

On supposera qu’au début du jeu, toutes les 52 cartes sont rangées dans un ordre aléatoire dans une pile. On représentera le jeux de cartes de chaque joueur par une pile.

1. Proposer une structure de données pour représenter le talon et les jeux des joueurs. On supposera qu’il y a 2 joueurs.
2. Écrire une fonction de distribution qui donne la moitié des cartes à chaque joueur.
3. Écrire une fonction permettant de jouer un coup : chaque joueur pose une carte, la plus grosse carte remporte le pli. En cas d’égalité des cartes, chaque joueur repose une carte et ainsi de suite jusqu’à ce qu’un des joueurs remporte le pli.
4. Écrire une fonction qui effectue l’ensemble du jeu et donne le joueur gagnant.