

Travaux Dirigés de Programmation II
[TD n°3 : Structures linéaires en C]

Objectifs : - Initier à la manipulation/implémentation de structures linéaires en C ;
- Introduire des applications des piles et/ou listes (implémentées par chaînage).

Exercice 1

Le but de cet exercice est d'implémenter en C, une pile de nombres entiers sous forme de liste chaînée (on parlera de pile chaînée). Pour cela :

- a)- Définir en *langage C* le type **PILE**, qui permet de représenter une pile sous forme d'une pile chaînée.
- b)- Ecrire la fonction, **est_vide**, permettant de créer une pile initialement vide.
- c)- Ecrire la fonction, **pile_vide**, qui permet de vérifier si une pile est vide ou non.
- d)- Ecrire la fonction, **sommet**, qui retourne l'élément au sommet d'une pile non vide.
- e)- Ecrire la fonction, **empiler**, qui permet d'ajouter un élément donné au sommet d'une pile.
- f)- Ecrire la fonction, **depiler**, qui supprimer l'élément du sommet d'une pile non vide.
- g)- Ecrire la fonction, **vider**, qui permet de réinitialiser à zéro une pile d'entiers. On notera que l'espace mémoire occupé par la pile devra être complètement libéré.

Exercice 2

On désire évaluer une expression arithmétique en notation postfixée, en utilisant comme opérateurs uniquement les quatre opérateurs **+**, **-**, ***** et **/**. On supposera que cette expression manipule uniquement des chiffres (les nombres sont des chiffres **0, 1, 2, ..., 9**), autrement dit un nombre entier est un seul chiffre.

On supposera aussi que l'expression est donnée sans qu'il y ait des séparateurs (pas d'espaces entre les opérateurs et les valeurs).

Par exemple, $36*25+-$ (qui est l'équivalent en infixé de $(3*6)-(2+5)$)

- a)- Ecrire la fonction, **int evaluer_postfixe(char *exp)**, permettant d'évaluer une expression postfixée donnée. **Indication** : utiliser une pile de nombre entiers.
- b)- Ecrire un programme C pour tester la fonction définie ci-dessus.

Exercice 3

On considère une liste d'étudiants. On caractérise un étudiant par son matricule (*entier*), son nom (*chaîne de 20 caractères au plus*), son prénom (*chaîne de 20 caractères au plus*), sa date de naissance (*3 entiers désignant le jour, le mois et l'année*) et sa moyenne générale (*réel*).

La liste d'étudiants sera représentée par une liste simplement chaînée d'étudiants.

- a)- Donner en *langage C* la déclaration complète du type **Liste_Etud**.
- b)- Ecrire la fonction, **admis**, qui permet d'afficher tous les étudiants ayant une moyenne **>= 10**.
- c)- Ecrire la fonction récursive, **nombre_etud_rec**, qui calcule le nombre d'étudiants dans une liste d'étudiants.
- d)- Refaire la question c)- en supposant que la liste d'étudiants est une liste chaînée circulaire.

Exercice 4

- a)- Ecrire la fonction, **tableau_vers_liste**, qui prend en paramètres un tableau et son nombre d'éléments, et qui crée une liste chaînée dont les éléments sont les mêmes que ceux du tableau.
- b)- Ecrire la fonction, **liste_vers_tableau**, qui fait le travail inverse de la fonction définie en a)-.

Exercice 5

Ecrire la fonction, **supprime_occur1**, qui prend en paramètres une liste chaînée et une donnée, et qui supprime la première occurrence de cette donnée dans la liste.

Exercice 6 (facultatif)

Refaire l'exercice 5 en supposant une liste doublement chaînée.