Centre Universitaire de Mila
Institut des sciences et de la technologie
2eme année L MD informatique
Module A.S.D.D.1

Série de TD N 02 : listes chainées

Exercice 01 :

Soit L une liste chainée d'entiers.

1. Donner la déclaration de cette liste.
2. Définir les opérations primitives (Crée_maillon, est_vide, premier, reste, ajouter, longueur) sur les listes chainées.
3. Ecrire les opérations suivantes sur la liste L.
   a) **saisir (L, n)** : permettant de saisir n entiers et les ajouter dans la liste vide L.
   b) **afficher (L)** : permettant d'affiche les éléments de la liste L.
   c) **somme (L)** : qui retourne la somme des éléments de la liste L.
   d) **min (L)** : qui retourne le minimum de la liste L.
   e) **miroir (L)** : qui retourne une liste contenant les mêmes éléments de L mais dans l'ordre inverse.
4. Ecrire l'algorithme permettant de saisir une liste de N entiers et ensuite affiche ses éléments, sa longueur, la somme de ses éléments, son minimum, et son miroir. L'algorithme doit libérer tout l'espace mémoire réservé par la liste L.

Exercice 02

Écrire les opérations récursives suivantes sur les listes chainées d'entiers.

1. **Afficher_pairs (L)** permettant d'afficher tous les éléments pairs de L.
2. **Appartient (L, x)** permettant de chercher un élément dans la liste L.
3. **Somme_neg (L)** qui retourne la somme des éléments négatifs de L.
4. **Max (L)** qui retourne le plus grand élément de L.
5. **Est_triée** (L) permettant de vérifier si une liste est triée dans un ordre croissant ou non.

Exercice 03: écrire une fonction (récursive et itérative) Fusion (L1, L2) qui prend deux listes triées dans l'ordre croissant et retourne une liste triée, dans le même ordre, contenant les deux listes.

Exercice 04: écrire une version itérative et une autre récursive pour les fonctions suivantes sur les listes chainées d'entiers.

1. Insérer (pos, x, L) : permettant d'insérer x dans la position pos.
2. Suprimer_pos (L, pos), permettant de supprimer l'élément existant dans la position pos.

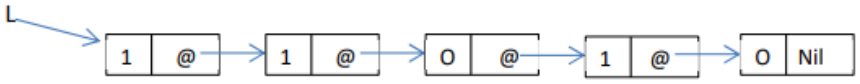
Exercice 05: Soit L une liste chainée d'étudiants pour le module ASDD3. Où chaque étudiant est représenté par son numéro, son nom, son prénom, une note TD, une note TP et une note contrôle.

1. Faire le découpage modulaire nécessaire et écrire l'algorithme permettant de saisir une liste de N étudiants et ensuite affiche la liste des étudiants qui ont obtenu le module.
2. Ecrire la fonction Ajourné (L) permettant de créer une liste contenant les étudiants qui n'ont pas obtenu le module.

Exercice 6 : supplémentaire

On convient de représenter un nombre binaire b1,b2,..., bn, où chaque bi est un 0 ou un 1, par une liste linéaire chainée où chaque élément de la liste contient un bi.

Voici un exemple d'une liste L contenant la représentation binaire du nombre décimal 26 :



1. Ecrire une fonction Egale (L1, L2) permettant de tester l'égalité de deux nombres entiers représentées par deux listes chainées L1, L2 (il faut faire la comparaison en binaire).

2. Ecrire une fonction Somme (L1, L2) admettant en entrée deux listes de taille quelconque et qui renvoie une nouvelle liste l3 contenant l'addition binaire de ces deux listes.

3. Ecrire une procédure Addition () permettant de lire deux nombres entiers et qui affiche le résultat de l'addition en décimale en faisant l'addition en binaire de ces deux nombres.

   Exemple: l'addition de 27 (1 1 0 1 1) et 9 (1 0 0 1) égale 36 (1 0 0 1 0 0)

$$\begin{array}{r} {}^{1}1\,1\,{}^{1}0\,{}^{1}1\,1 \\ 1\,0\,0\,1 \\ \hline 1\,0\,0\,1\,0\,0 \end{array}$$

Exercice 7 : supplémentaire

Nous considérons la construction d'une liste des nombres premiers inférieurs ou égaux à un entier n donnée. Pour construire cette liste, on commence, dans une première phase, par y ajouter tous les entiers de 2 à n en commençant par le plus grand et en terminant par le plus petit qui se trouvera a la tête de la liste. On considère ensuite successivement les éléments de la liste dans l'ordre croissant en on supprime tous leurs multiples stricts. Écrire le programme et les fonctions nécessaires.