



## Fiche N° 4 : Listes doublement Chainées

### Exercice 1 :

1. Proposer une structure pour déclarer une liste doublement chaînée contenant des éléments entiers.
2. Écrire une fonction d'initialisation d'une liste doublement chaînée.
3. Écrire une fonction pour afficher une Liste doublement Chainée.
4. Écrire une fonction pour ajouter un nouvel élément au début de la liste.
5. Écrire une fonction pour ajouter un nouvel élément à la fin de la liste.
6. Écrire une fonction pour supprimer l'élément du début de la liste.
7. Écrire une fonction pour supprimer l'élément en fin de la liste.
8. Écrire une fonction récursive pour afficher une Liste doublement Chainée.
9. Écrire une fonction récursive pour afficher une Liste doublement Chainée par ordre inverse.

### Exercice 2 :

On dispose d'une liste doublement chaînée L, triée par ordre croissant sur le champ entier:

1. Ecrire une fonction qui permet d'insérer un élément dans L.
2. Ecrire une fonction qui permet de supprimer un élément de valeur v de L.
3. Ecrire une fonction (itérative et récursive) qui permet de fusionner deux listes triées dans une seule.
4. Ecrire un programme qui permet de tester les fonctions précédentes.

### Problème :

Considérez une liste doublement chaînée avec des nœuds contenant chacun un entier. L'objectif est de concevoir un programme pour trouver la sous-séquence continue (contiguous subsequence) de la liste dont la somme est maximale.

Objectifs :

- Implémentez une fonction `findMaxSubsequence` qui prend en entrée une liste doublement chaînée et renvoie la somme maximale d'une sous-séquence continue de cette liste.
- La sous-séquence doit consister en au moins un élément de la liste.
- Si la liste contient uniquement des éléments négatifs, la fonction doit renvoyer la valeur du plus petit élément.

Exemple :

- Pour la liste doublement chaînée {1, -2, 3, 10, -4, 7, 2, -5}, la sous-séquence avec la somme maximale est {3, 10, -4, 7, 2}, et la somme maximale est 18.