Année universitaire 2023/2024 **Niveau : MPI**

Algorithmique et structures de données II

TD2 : Listes

Exercice 1

Les listes on été traitées en cours a base de l'approche chainée. Nous désirons ici donner leur implémentation à base de tableaux. **Les traitements demandés seront réalisés avec le langage C**.

1. Donner la structure de données qui représente une liste a base de tableaux.
2. Donner la fonction **Créer_Liste** qui crée une liste vide.
3. Donner la fonction **Adjq_Liste** qui correspond à l'ajout d'un élément à la fin de la liste.
4. Donner la fonction **Adjt_Liste** qui rajoute un élément au début de la liste.
5. Donner la fonction **Supprimer_dernier_Liste** qui supprime le dernier élément.
6. Donner la fonction **Supprimer_premier_Liste** qui supprime le premier élément.

Exercice 2

On considère les listes **chainées**. **Les traitements demandés seront réalisés avec le langage C**.

1. Donner une fonction récursive qui affiche les valeurs d'une liste chainée et retourne sa longueur.
2. Donner une fonction itérative qui permet la suppression d'un élément suivant une position donnée.
3. Donner une fonction itérative qui permet l'ajout d'un élément suivant une position donnée.
4. Donner une fonction itérative qui réalise la concaténation de deux listes.
5. Donner une fonction itérative qui inverse une liste.
6. Donner une fonction récursive qui inverse une liste.

Exercice 3

On considère une liste chainée contenant des entiers ordonnés par ordre croissant et que l'on veut la maintenir ordonnée.

1. Ecrire en C la fonction **Insertion** qui permet d'insérer un élément a la bonne position (la liste reste ordonnée après l'opération).
2. Ecrire en C la fonction **Elimination** qui élimine les éléments redondants d'une liste.

Exercice 4

On considère que les listes chainées sont représentées par la structure de données suivante :

```
typedef struct el
{int val;
struct el *precedent;
struct el *suivant;
} element;

typedef struct liste
{ element *debut;
element *fin;
int taille;
}Liste;
```

Les traitements demandés seront réalisés avec le langage C.

1. Donner la fonction qui permet de générer une liste vide
2. Donner la fonction qui permet de tester si une liste est vide
3. Donner la fonction qui permet d'ajouter en tête un élément dans la liste
4. Donner la fonction qui permet de supprimer le dernier élément de la liste

Exercice 5

Soient L1 et L2 deux listes linéaires chaînées monodirectionnelles.

1. Ecrire une fonction en C qui construit la liste La = L1 - L2 contenant tous les éléments appartenant à L1 et n'appartenant pas à L2
2. Ecrire une fonction en C qui construit la liste Lb = L1 + L2 contenant tous les éléments appartenant à L1 et à L2

Exercice 6

On considère deux ensembles d'entiers A={a1,a2,…,an} et B={b1,b2,…,bm}. Chaque ensemble est stocké dans une liste chaînée.
On définit la différence symétrique de deux ensembles que l'on note par D comme suit :

$$\mathbf{D(A,B) = (A \cup B) \setminus (A \cap B)}$$

Le résultat doit être stocké dans une liste chaînée.

1. Ecrire en C une fonction itérative calculant la différence symétrique de deux ensembles représentés par deux listes chaînées.
2. Ecrire en C une fonction récursive calculant la différence symétrique de deux ensembles représentés par deux listes chaînées.