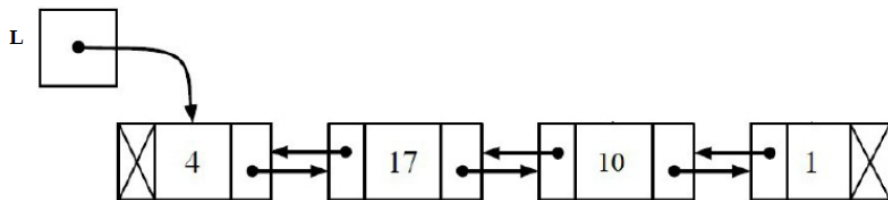


TD 5

LISTE DOUBLEMENT CHAÎNÉE

Exercice 1 :

Pour représenter une liste d'entiers (4; 17; 10; 1), on peut utiliser une liste doublement chaînée. Dans cette liste chaque cellule contient un entier et deux pointeurs, un contenant l'adresse de la cellule précédente et l'autre contenant l'adresse de la cellule suivante :



Q1. Donnez la syntaxe qui permet de définir cette liste doublement chaînée.

```
typedef struct cellule{
.....
.....
.....
}Dliste;
```

Q2. Ecrire une fonction d'entête `Dliste* initListe(Dliste *L)` qui permet d'initialiser une liste doublement chaînée L.

Q3. Ecrire une fonction d'entête `Dliste* InsertDebut(Dliste *L, int x)` qui insère, en tête de la liste doublement chaînée L un entier x.

Q4. Ecrire une fonction d'entête `Dliste* InsertFin(Dliste *L, int x)` qui insère, à la fin de la liste doublement chaînée L un entier x.

Q5. Ecrire une fonction d'entête `Dliste* SupprimDebut(Dliste*L)` qui supprime le premier élément d'une liste chaînée.

Q6. Ecrire une fonction d'entête `Dliste* SupprimFin(Dliste*L)` qui supprime le dernier élément d'une liste chaînée.

Q7. Ecrire une fonction d'entête `Dliste* SupprimElement(Dliste*L, int x)` qui supprime la première occurrence d'un entier n dans une liste doublement chaînée L.

Q8. Ecrire une fonction d'entête `Dliste* Vider(Dliste*L)` qui supprime tous les éléments de L.

Q9. Ecrire une fonction d'entête `void AffListe(Dliste * L)` qui affiche les éléments de la liste doublement chaînée passée en paramètre.

Q10. Ecrire une fonction d'entête `int Existe(Dliste * L,int x)` qui retourne 1 si x existe dans la liste L ou 0 sinon.

Exercice 2

On considère une liste doublement chaînée définie par :

```
//Définir le type cellule
typedef struct cellule{
    info info;
    struct cellule *preced;
    struct cellule *suiv;
}tCellule;
//Définir le type liste
typedef struct listeDchainee{
    tCellule *debut
    tCellule *fin
}Dliste;
```

Q1. Reprendre toute les questions de l'exercice 1.

Exercice 3

On considère que L1 et L2 sont deux listes doublement chaînées

Q1. Faites un dessin qui montre comment vous allez ajouter L2 à la fin de L1.

Q2. Ecrivez ensuite la fonction ajouterEnFin(L1;L2).

Q3. Faites la même chose pour ajouter L2 au début de L1. On nommera la fonction ajouterAuDbut(L1,L2).

Q4. Faites une fonction qui supprime de L1 tous les éléments dont la donnée associée est impaire (on utilisera la fonction donnéeEstPaire(x)) et qui ajoute tous ceux de L2 dont la donnée associée est paire.

Exercice 4 : “Autres fonctions ”

Q1. Écrivez une fonction vérifiant si une liste d'entiers doublement chaînée est triée.

Q2. Donnez une fonction renvoyant l'adresse de la dernière cellule d'une liste doublement chaînée non vide.

Q3. Écrivez une fonction qui modifie une liste doublement chaînée non vide en déplaçant son premier élément en dernière position.

Q4. Écrivez une fonction qui supprime les répétitions d'éléments consécutifs égaux d'une liste doublement chaînée non vide. Plus précisément si la liste contient une séquence d'éléments de même valeur, il ne faut garder qu'un élément de cette séquence. Par exemple la séquence (1,8,8,8,2,2,9,7,9,9) devient (1,8,2,9,7,9).