INSEA
Institut National des Statistiques et d'Economie Appliquée
Pr. ELHARI Kaoutar

Année Universitaire 2019-2020
Structures de Données
S2 : DES-DS-RO

TP 5 : Les listes doublement chaînées

Le but de cet exercice est de manipuler les listes doublement chaînées. A la différence des listes simplement chaînées vues en cours, un nœud d'une liste doublement chaînée possède en plus un pointeur sur l'élément qui le précède.

Une liste doublement chaînée peut être représentée de la façon suivante :

```
typedef struct noeud_S *Noeud;

typedef struct noeud_S {
       int info;
       Noeud suivant;
       Noeud precedant;
} noeud;

typedef struct {
       Noeud tete;
       Noeud queue;
       int nbr_noeud;
} liste_DC;

typedef liste_DC *liste;
```

- Ecrire une fonction qui permet de créer une liste vide. le contenu de la tête et de la queue est l'entier 0. Son prototype est : liste creationListeVide()
- Ecrire une fonction { liste ajouterDebut(liste l, int data)} qui permet d'ajouter un élément au début d'une liste.
- Ecrire une fonction { liste ajouterFin(liste l, int data)} qui permet d'ajouter un élément à la fin d'une liste.
- Ecrire une fonction {void afficher(liste l)} qui permet d'afficher les éléments d'une liste.
- Ecrire une fonction {void afficherInverse(liste l)} qui permet d'afficher d'une manière inversée les éléments d'une liste.
- Ecrire une fonction {void chercherNoeud(liste l, int data)} qui permet de rechercher un élément dans une liste et d'afficher sa position s'il est trouvé.
- Ecrire une fonction { liste supprimer(liste l, int data)} qui permet de supprimer un élément donné d'une liste.