

TP2: OPÉRATIONS SUR LES LISTES DOUBLEMENT CHAÎNÉES

Ecrire un programme C qui permet de créer et de lire une liste doublement chaînée. Pour définir un élément de la liste le type struct sera utilisé. Un élément de la liste contient les trois champs suivants :

- valeur de type entier.
- precedent un pointeur de même type qu'un élément de la liste.
- suivant un pointeur de même type qu'un élément de la liste.

```
typedef struct ElementListe{  
    int valeur;  
    struct ElementListe *precedent;  
    Bstruct ElementListe *suivant;} Element;
```

1. Ecrire la fonction qui permet d'insérer un élément dans une liste vide.
2. Ecrire la fonction qui permet d'insérer un élément au début d'une liste.
3. Ecrire la fonction qui permet d'insérer un élément à la fin d'une liste .
4. Ecrire la fonction qui permet d'insérer un élément avant un élément de la liste.
5. Ecrire la fonction qui permet d'insérer un élément après un élément de la liste.
6. Ecrire la fonction qui permet d'afficher la liste (de droite à gauche, et de gauche à droite).
7. Ecrire la fonction qui permet de supprimer ensuite tous les entiers qui sont inférieurs ou égales à une valeur entrée par l'utilisateur.
8. Ecrire la fonction qui permet l'échange (**Element *lis, int p, int q**) qui intervertit les éléments de la liste **lis** doublement chaînée situés aux positions p et q.
9. Ecrire la fonction qui permet **inverse_liste(Element *lis)** qui inverse l'ordre des éléments d'une liste.
10. Ecrire la fonction qui permet de trier la liste en utilisant l'algorithme de le tri à bulles .
11. Ecrire la fonction qui permet d'insérer un entier dans la liste en ordre croissant.