Université Moulay Ismail **AU 2019/2020**
Faculté des Sciences et Techniques Errachidia **Filière MIP/ S4**
Département d'Informatique
Module I143 **Responsable Pr. B. AKSASSE**

Série 2 TD Structures de données en C : les listes chaînées

Exercice 1

On veut implémenter un programme qui lit une suite d'entiers saisis au clavier par un utilisateur et crée la liste simplement chaînée correspondante.

1) **Ecrire un programme en C qui permet de :**
   a. **Définir la structure d'un élément de la liste simplement chaînée**
   b. **Définir la structure de contrôle de la liste simplement chaînée**
   c. **Ecrire la fonction CreerElement( ) qui crée, alloue, initialise et retourne un élément vide de la liste.**
   d. **Ecrire la fonction CreerListe( ) qui crée, alloue, initialise et retourne une liste vide**
   e. **Ecrire la fonction CreerListeOrdre( ) qui crée la liste en respectant l'ordre de lecture des entiers positifs au fur et à mesure de la saisie des éléments. La condition d'arrêt est lorsqu'on saisit un entier négatif.**
   f. **Ecrire la fonction CreerListeCroissante( ) qui crée la liste en rangeant les entiers positifs par ordre croissant au fur et à mesure de la saisie des éléments. La condition d'arrêt est lorsqu'on saisit un entier négatif.**

2) **On veut maintenant développer une fonction qui permet de concaténer les deux listes chaînées créées précédemment et en effectuer quelques opérations. A la suite du programme précédent :**
   a. **Ecrire la fonction AjoutFin( ) qui permet d'insérer une valeur (vaf) à la fin d'une liste chaînée.**
   b. **En utilisant la fonction AjoutFin( ), écrire la fonction ConcatListe( ) qui permet de concaténer les deux listes en insérant leurs éléments dans une autre nouvelle liste vide.**
   c. **Ecrire la fonction RechercheElem( ) qui permet de chercher une valeur (val) dans la liste concaténée.**
   d. **Ecrire la fonction SuppPos( ) qui retourne et supprime un élément de la liste concaténée après une position de suppression désirée (pos).**

N.B. : On donne les prototypes des différentes fonctions :

```
Element* CreerElement( ) ;
Liste* CreerListe( ) ;
void CreerListeOrdre(Liste* Li) ;
void CreerListeCroissante(Liste* Li) ;
void AjoutFin(Liste* Li, int vaf) ;
Liste* ConcatListe (Liste* Li1, Liste* Li2) ;
int RechercheElem(Liste* Li, int val) ;
int SuppPos(Liste*Li, int pos) ;
```