Université Moulay Ismail
Faculté des Sciences et Techniques
Département d'Informatique, Errachidia

A.U. : 2019-2020
Structures de Données en C, Filière MIP, S4
Préparé par Pr. B. BOUDA

# T.D. N°2

**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. Écrire 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**) Écrire la fonction CreerElement( ) qui crée un élément vide de la liste.
   - **d**) Écrire la fonction CreerListe( ) qui crée une liste vide.
   - **e**) Écrire la fonction CreerListeOrdre( ) qui crée la liste en respectant l'ordre de lecture des entiers au fur et à mesure de la création de la liste.
   - **f**) Écrire la fonction CreerListeCroissante( ) qui crée la liste en rangeant les entiers par ordre croissant au fur et à mesure de la création de la liste.

On veut maintenant développer une fonction qui permet de concaténer les deux listes crées précédemment et en effectuer quelques operations.

2. A la suite du programme précèdent :
   - **a**) Écrire 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 Concatenation( ) qui parcourt les deux listes et insère respectivement, leur éléments dans une autre liste vide.
   - **c**) Écrire la fonction RechercheElem( ) qui cherche une valeur (ValR) dans de la liste concaténée.

N.B. : On vous 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* Concatenation(Liste* Li1, Liste* Li2)
- int RechercheElem(Liste* Li, int ValR)