Université Sultan Moulay Slimane **Année Universitaire 2019-2020**
Faculté des Sciences et Techniques
Béni Mellal
Département d'Informatique

TP4 : Structures de données en langage C
Parcours MIPC

L'objectif de ce problème est d'écrire un programme qui permet de calculer la somme de deux grands nombres entiers. Pour réaliser cet objectif on doit lire une chaîne de caractères représentant un grand nombre entier et le charger par paquets de 4 chiffres dans une liste doublement chaînée (LDC).
Pour définir un élément de la liste le type ***struct*** sera utilisé. L'élément de la liste contiendra un champ ***info*** de type entier qui représente les 4 chiffres du paquet et deux pointeurs ***suivant*** et ***precedent.*** Les pointeurs *sont* de même type que l'élément de la liste. Le pointeur ***suivant*** permettra l'accès vers le successeur et le pointeur ***precedent*** permettra l'accès vers le prédécesseur.

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

Chaque nombre est donc représenté par une LDC et pour avoir le contrôle de la liste, il est préférable de sauvegarder les éléments suivants :

- Le pointeur ***debut*** contiendra l'adresse du premier élément de la liste.
- Le pointeur ***fin*** contiendra l'adresse du dernier élément de la liste.
- La variable ***Npaquets*** contient le nombre d'éléments (nombre de paquets).

Pour simplifier l'écriture du programme et pour ne pas avoir plusieurs variables globales dans le programme, une autre structure doit être ajoutée pour représenter les deux nombres à additionner (**Nombre1** et **Nombre2**) et le grand nombre somme (**Nombre3**).

```
typedef struct Nombre {
                Element *debut;
                Element *fin;
                int Npaquets;
        } GrandNombre;
```

Les variables globales du programme peuvent être :

GrandNombre Nombre1, Nombre2, Nombre3 ;

char ch1[30], ch2[30] ; /* ch1 chaîne représente le premier grand nombre*/

/* ch2 chaîne représente le deuxième grand nombre*/

1- Ecrire la fonction qui permet d'initialiser les pointeurs **debut** et **fin** à **NULL** et le **Npaquets** avec la valeur 0 d'un grand nombre.
Prototype de la fonction : **initialisation(GrandNombre *Nb);**

2- Ecrire la fonction qui permet d'insérer un élément dans une liste vide.
Prototype de la fonction : **int ins_liste_vide(GrandNombre *Nb, int info);**

3- Ecrire la fonction qui permet d'insérer un élément à la fin de la liste.
Prototype de la fonction : **int ins_liste_fin(GrandNombre *Nb, int info);**

4- Ecrire la fonction qui permet de charger une chaîne de caractères par paquets de 4 chiffres dans la liste.
Prototype de la fonction **: convertir (GrandNombre *Nb, char *chaine);**

5- Ecrire la fonction qui permet de faire la somme de deux grands nombres.
Prototype de la fonction **:**
somme(GrandNombre Nb1, GrandNombre Nb2, GrandNombre *Somme3);

6- Ecrire la fonction qui permet d'afficher le grand nombre correctement.
Prototype de la fonction **: afficher_envers(GrandNombre Nb);**

7- Ecrire la fonction **main()** pour tester les différentes fonctions.