Université Sultan Moulay Slimane **Année Universitaire 2015/2016**
Faculté des Sciences et Techniques
Béni Mellal
Département d'Informatique

Correction TD4 : Structures de données en langage C
Parcours MIPC

```
#include <stdio.h>
#include <string.h>

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

typedef struct Nombre {
      int Npaquets;
      Element *Debut;
      Element *Fin;} GrandNombre;

GrandNombre Nombre1, Nombre2, Nombre3;
char ch1[30], ch2[30];
```

initialisation (GrandNombre *Nb) {Nb->Npaquets=0;Nb->Debut=NULL;Nb->Fin=NULL;}

```
int ins_dans_liste_vide(GrandNombre *Nb,int i) {
Element *element;
if ((element = (Element *) malloc(sizeof(Element)))==NULL) return 0;
element-> info = i ;
element-> suivant = NULL;
element-> precedent = NULL;
Nb->Debut = element; Nb->Fin = element;
Nb->Npaquets++;
return 1;
}
```

```
int ins_fin_liste (GrandNombre *Nb,int i) {
Element *element;
if ((element = (Element *) malloc(sizeof(Element)))==NULL) return 0;
element-> info = i ;
element-> suivant = NULL;
Nb->Fin->suivant = element;
element->precedent=Nb->Fin;
Nb->Fin = element;
Nb->Npaquets++;
return 1;
}
```

```
Convertir(GrandNombre *Nb, char *chaine) {
int Long, reste, i, val, N, t;
char *ch4, ch_reste[4],*ch;
Long= strlen(chaine);
N=Long/4;
for (i=0; i<N; i++) {
                val=Long - 4*(i+1);
                ch=&chaine[val];
                ch4=(char *)malloc(4*sizeof(char));
                strncpy(ch4, ch, 4);
                t=atoi(ch4);
                if (Nb->Npaquets==0) ins_dans_liste_vide(Nb,t);
                else ins_fin_liste (Nb,t);
}
reste=Long%4;
if (reste!=0) {
            ch=&chaine[0];
            ch4=(char *)malloc(reste*sizeof(char));
            strncpy(ch_reste, ch, reste);
            t=atoi(ch_reste);
            if (Nb->Npaquets==0) ins_dans_liste_vide(Nb,t);
            else ins_fin_liste (Nb,t);
           }
}

Somme(GrandNombre Nb1, GrandNombre Nb2, GrandNombre *Nb3) {
int s, ret, t, i;
ret=0;
Element *element1, *element2 ;
element1=Nb1.Debut; element2=Nb2.Debut;
for (i=1; i<=Nb1.Npaquets;i++) {
                        s = element1->info + element2->info + ret;
                        t = s%10000;
                        if (Nb3->Npaquets==0) ins_dans_liste_vide(Nb3,t);
                        else ins_fin_liste (Nb3,t);
                        ret= s/10000;
                        element1=element1->suivant;
                        element2=element2->suivant; }
for (i=Nb1.Npaquets+1; i<=Nb2.Npaquets; i++) {
                                s = element2->info + ret;
                                t = s%10000;
                                if (Nb3->Npaquets==0) ins_dans_liste_vide(Nb3,t);
                                else ins_fin_liste (Nb3,t);
                                ret= s/10000;
                                element2=element2->suivant; }
if (ret !=0) ins_fin_liste(Nb3,ret);
}
```

```
afficher_envers(GrandNombre Nb) {
char ch[4];
int i,s;
Element *element;
element=Nb.Fin;
printf("\ affichage \n" );
sprintf(ch,"%d",element->info); printf("%s",ch);
element=element->precedent;
while (element != NULL) {
                s=element->info;
                if (s<10) sprintf(ch,"000%d", s);
                else if (s<100) sprintf(ch,"00%d", s);
                     else if (s<1000) sprintf(ch,"0%d", s);
                          else sprintf(ch,"%d", s);
                printf("%s",ch);
                element=element->precedent;
}
printf("\nfin affichage\n");
}

main() {
printf("Donner un très grand nombre : "); gets(ch1);
printf("Donner un très grand nombre : "); gets(ch2);
initialisation (&Nombre1); Convertir(&Nombre1, ch1);
initialisation (&Nombre2); Convertir(&Nombre2, ch2);
initialisation (&Nombre3);
if (Nombre1.Npaquets<=Nombre2.Npaquets) somme(Nombre1, Nombre2, &Nombre3);
else somme(Nombre2, Nombre1, &Nombre3);
printf(" la somme =\n");
afficher_envers(Nombre3);
printf("apres \n");
system("pause");
}
```