Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Béni-Mellal
Département d'Informatique

Année universitaire 2015/2016

Correction TP2
Structures de données en langage C
Parcours MIPC

```
#include <stdio.h>
#define Max 30

typedef struct {
   int Temps;
   char Distance[6];
   char Termine[4];
}Score;

typedef struct {
  int   Code;
  char *Nom_Prenom;
  char *Date_Naissance;
  Score Competetion;
}Athelete;

typedef struct element {
  Athelete jou;
  struct  element *suivant;
} Noeud;

typedef struct  {
  Noeud *Debut;
  Noeud * Fin;
  int Nb_Athletes;
} LSC;
```

initialisation(LSC *L) {L->Debut=NULL;L->Fin=NULL; L->Nb_Athletes=0;}

Noeud * saisir(){

```
Noeud *e;
if ((e=(Noeud *)malloc(sizeof(Noeud)))==NULL) return NULL;
if ((e->jou.Nom_Prenom=(char *)malloc(40*sizeof(char)))==NULL) return NULL;
if ((e->jou.Date_Naissance=(char *)malloc(40*sizeof(char)))==NULL) return NULL;
printf("Donner le nom et prenom de l athelete  : "); gets(e->jou.Nom_Prenom);
printf("Donner sa date de Naissance  :"); gets(e->jou.Date_Naissance);
printf("Donner le code du joueur   :"); scanf("%d",&e->jou.Code);
printf("Donner le temps effectue   : "); scanf("%d",&e->jou.Competetion.Temps);
printf("Donner la Distance de la course   : "); gets(e->jou.Competetion.Distance);
printf("Est ce que l'Athlète à terminer la course oui/non : ");
gets(e->jou.Competetion.Termine);
return e;
}
```

affiche(LSC L){

```
Noeud *courant;
if (L.Debut==NULL) printf("pas d'Athlete à afficher \n");
else {
     courant=L.Debut;
     while(courant!=NULL) {
        printf("Nom_Prenom : %s \n",courant->jou.Nom_Prenom);
        printf("Date_Naissance : %s\n", courant->jou.Date_Naissance);
        printf("Temps : %d \n",courant->jou.Competetion.Temps);
        printf("Distance : %s \n",courant->jou.Competetion.Distance);
        printf("Termine course : %s\n",courant->jou.Competetion.Termine);
        courant=courant->suivant;
    }
    printf("Fin liste\n");
  }
}
```

```
int Supprimer_Athletes_Non_Termine(LSC *L){
Noeud * courant, *avant ;
if (L->Debut==NULL) {printf("pas d athelte a supprimer \n");return 0;}
courant=L->Debut; avant=L->Debut;
while (courant !=NULL) {
    if (strcmp(courant->jou.Competetion.Termine,"non")==0) {
        if (courant==L->Debut) {
                L->Debut=L->Debut->suivant;
                if (L->Debut==NULL) L->Fin=NULL;
                }
        else {
           avant->suivant=courant->suivant;
           if  (avant->suivant=NULL) L->Fin=avant;
        }
        L->Nb_Athletes--;
        free(courant->jou.Nom_Prenom);
        free(courant->jou.Date_Naissance);
        free(courant);
     }
    avant=courant;
    courant=courant->suivant;
}
return 1;
}

Noeud *preparerNoeud(Noeud *p){
Noeud *e;
if ((e=(Noeud *)malloc(sizeof(Noeud)))==NULL) return NULL;
if ((e->jou.Nom_Prenom=(char *)malloc(40*sizeof(char)))==NULL) return NULL;
if ((e->jou.Date_Naissance=(char *)malloc(40*sizeof(char)))==NULL) return NULL;
strcpy(e->jou.Nom_Prenom,p->jou.Nom_Prenom);
strcpy(e->jou.Date_Naissance,p->jou.Date_Naissance);
e->jou.Code=p->jou.Code;
e->jou.Competetion.Temps=p->jou.Competetion.Temps;
strcpy(e->jou.Competetion.Distance,p->jou.Competetion.Distance);
```

```
strcpy(e->jou.Competetion.Termine,p->jou.Competetion.Termine);
e->suivant=NULL;
return e;
}

Divise_Deux_Listes(LSC L,LSC *L400, LSC *L3000) {
Noeud *s, *p;
initialisation(L400);
initialisation(L3000);
if (L.Debut==NULL) printf(" rien à diviser \n ");
else {
    p=L.Debut;
    while(p!=NULL){
            s=preparerNoeud(p);
            if (strcmp(p->jou.Competetion.Distance,"400_m")==0)
              if (L400->Nb_Athletes==0) {
                    L400->Debut=s; L400->Fin=s;
                    s->suivant=NULL; L400->Nb_Athletes=1;
                    }
              else {
                  s->suivant=L400->Debut;
                  L400->Debut=s;
                  L400->Nb_Athletes ++;
                  }
             else
              if (L3000->Nb_Athletes==0) {
                    L3000->Debut=s; L3000->Fin=s;
                    s->suivant=NULL; L3000->Nb_Athletes=1;
                    }
              else {
                  s->suivant=L3000->Debut;
                  L3000->Debut=s;
                  L3000->Nb_Athletes ++;
                  }
    }
```

```
   }
}
```

trier(LSC *L){

```
Noeud *p , *q, *aux;
aux=(Noeud *)malloc(sizeof(Noeud));
aux->jou.Nom_Prenom=(char *)malloc(40*sizeof(char));
aux->jou.Date_Naissance=(char *)malloc(40*sizeof(char));
p=L->Debut ;
int en_ordre = 0;
while(en_ordre==0) {
   en_ordre = 1;
   while(p->suivant!=NULL) {
        q=p->suivant;
        while(q!=NULL){
            if(p->jou.Competetion.Temps > q->jou.Competetion.Temps) {
              *aux=*p; *p=*q; *q=*aux; en_ordre=0;}
             q=q->suivant;
        }
        p=p->suivant;
   }
}
}
```