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

Correction TD1 MIPC
Structures de Données en Langage C

Exercice 1:

```
#include<stdio.h>
#define max 50
char p[max]; // déclaration de la  pile
int sommet ;

initialisation() {  sommet=-1; }

void empiler(char x) {
          if (sommet>=max-1) printf("pile pleine");
         else { sommet++; p[sommet]=x; }
}

char depiler () {
char x;
 if (sommet<0)  { printf("pile vide"); return '\0'; }
 else   { x=p[sommet];  sommet--;       return x;}
 }

int pilevide() { return sommet<0;}

int pilepleine() {return sommet>=max-1;}

main() {
  char a,ch[max];
  int i,lg,valide=1;
  printf("une donner une chaine\n"); gets(ch);
  lg=strlen(ch);
  for(i=0;i<lg/2;i++) empiler(ch[i]);
  if(lg%2==0)  for(i=lg/2;((i<lg) && (valide==1));i++) { a=depiler();   if(ch[i]!=a)  valide=0; }
  else for(i=lg/2+1; ((i<lg) && (valide==1));i++) {a=depiler();  if(ch[i]!=a)  valide=0;}
  if (pilevide() && (valide==1)) printf("chaine palindrome  \n");
  else printf("chaine non palindrome \n");
  system("pause");
 }
```

Exercice 2:

```
#include<stdio.h>
#include<stdlib.h>
#define N 20

typedef struct{
      int t[N];
      int sommet;
      } pile;

void Initialiser_Pile(pile *p) {  p->sommet=-1;}

int pile_Vide(pile p) {  return (p.sommet>0) ;}

int depiler(pile *p) {
   int supp;
   if(pile_Vide(*p))  {printf("la pile  est vide\n"); return -1;}
   else { supp=p->t[p->sommet];  p->sommet--;return supp;}
}

void  empiler(pile *p,int elt) {
       if(!pile_Vide(*p)) printf("la pile  est pleine\n");
       else {  p->sommet++;   p->t[p->sommet]; }
}

int Sommet_pile(pile p) {  return (p.sommet) ;}

void afficher(pile p) {
    int i;
    for(i=0;i<p.sommet;i++)  printf("%d\n",p.t[i]);
}

void deplacer(pile p1,pile *p2) {
 pile *p3;
 int m,s;
          Initialiser_Pile(p3);
          while((!pile_Vide(p1))) {
                    m=depiler(&p1);
                    if(m%2==0) empiler(p3,m); else empiler(p2,m);
          }
          While (!pile_Vide(*p3)) { s=depiler(p3);  empiler(p2,s);  }
}

void copier(pile p1,pile *p2) {
 pile *p3;
 int m,s;
          Initialiser_Pile(p3);
          While (!pile_Vide(p1)) {
                    m=depiler(&p1);
```

```
                    if(m%2==0) empiler(p3,m);
        }
        while(p3->sommet>0){
                    s=depiler(p3);
                    empiler(p2,s);
                    }
}
```

Exercice 3 :

```
#include<stdlib.h>
#define N 20

typedef struct elmt {// Les structures de données à utiliser ;
      int p[N];
      int sommet;
      } pile;

 typedef struct {
      int F[N];
      int ar;
 } File;

File f1, f2, f3;  // Les variables globales ;

void InitialiserFile(File *F)  { // fonction initialiser une File ;
        F->ar=-1;
 }

int fileVide(File F) {  // fonction file est Vide ? ;
     return (F.ar==-1);
}

void enfiler(File *F, int val)  { // fonction enfiler ;
if (F->ar==N-1) printf("la file est pleine \n");
else { F->ar++; F->F[F->ar]=val; }
}

int defiler(File *Fi)   { // fonction defiler ;
 int x, t;
 int supp;
 if (fileVide(*Fi)) { printf("la file est vide \n"); return 0; }
 else {   supp=Fi->F[0];
          for (t=1; t<=Fi->ar;t++) Fi->F[t-1]= Fi->F[t] ;
          Fi->ar-- ;
          return supp;
      }
}
```

```
void defilerEnfiler(File *F1,File *F2) {
  enfiler(F2, defiler(F1));
}

void afficher(File Fi)  {   // fonction affichage ;
   int  courant=Fi.ar;
   while(courant!=-1) {  printf(" %d  ", Fi.F[courant]); courant --; }
   printf("\n");
}

void trier(File *F1,File *F2) { // fonction Trier ;
  int min, m, iter, pos, t, s;
  if (fileVide(*F1))  printf("la file est vide \n ") ;
  else
                while (!fileVide(*F1)) {
                    min= defiler(F1);
                    if (!fileVide(*F1)) {
                                    t=0;  m=F1->ar;
                                    while(t<=m) {
                                                    s=defiler(F1);
                                                    if(min>=s) { enfiler(F1,min); min=s; }
                                                    else enfiler(F1,s);
                                                    t++;
                                    }
                     }
                   enfiler(F2,min);
             }
}

main(){              // fonction main ;
     int n;
    InitialiserFile(&f1);
    InitialiserFile(&f2);
    printf("entrer les elements de la file et taper un nombre négative pour terminer :\n");
    while(1){
            scanf("%d",&n);
            if(n<0) break; else enfiler(&f1,n);
    }
   printf("\n**********la file 'avant' le tri est : **********\n");
   afficher(f1) ;
   trier(&f1,&f2);
   printf("\n**********la file 'après' le tri est : **********\n");
   afficher(f2) ;
   system("pause");
}
```