UNIVERSITE DE JIJEL

FACULTE DES SCIENCES EXACTES ET INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Année universitaire 2020-2021

**Matière: Algorithmique et Structures de Données 3**

# TP N° 0

## Exercice 1:

Soit T un tableau de n entiers.

```
void    LireTab (int n, int * T) ; //  void    LireTab (int n, int  T[]) ;
int   IMinTab (int n, int * T) ; //  int    IMinTab (int n, int  T[]) ;
int   NbrOccMin (int n, int * T) ; //  int    NbrOccMin (int n, int  T[]) ;
void   AfficherTab (int n, int * T) ; //  void    AfficherTab (int n, int  T[]) ;
```

1. Ecrivez une fonction LireTab qui permet de lire les éléments du tableau T.
2. Ecrivez une fonction IMinTab qui renvoie l'indice de la plus petite valeur dans le tableau T.
3. Ecrivez une fonction NbrOccMin qui renvoie le nombre d'occurrences de la plus petite valeur dans le tableau T.
4. Ecrivez une fonction AfficherTab qui permet d'afficher les éléments du tableau T.
5. Ecrivez un programme en langage C qui permet :
   - De lire un tableau Tab de n entiers,
   - D'afficher la valeur et le nombre d'occurrences du plus petit élément dans le tableau Tab,
   - D'afficher les éléments du tableau Tab.

## Exercice 2:

Soit L une liste d'entiers.

```
typedef struct  ElementListe {
        int  Val ;
        struct  ElementListe * Suiv ;
} ElementListe ;
 typedef  ElementListe * Liste ;

int Longueur (Liste L) ;
void InsererDebut (int X, Liste * L) ;
void InsererFin (int X, Liste * L) ;
void SupprimerPremier (Liste * L) ;
void SupprimerDernier (Liste * L) ;
void AfficherListe (Liste L) ;
```

1. Ecrivez une fonction Longueur qui calcule le nombre des éléments de la liste L.
2. Ecrivez une fonction InsererDebut qui insère une valeur X au début de la liste L.
3. Ecrivez une fonction InsererFin qui insère une valeur X à la fin de la liste L.
4. Ecrivez une fonction SupprimerPremier qui supprime le premier élément dans la liste L.

5. Ecrivez une fonction SupprimerDernier qui supprime le dernier élément dans la liste L.
6. Ecrivez une fonction AfficherListe qui affiche les éléments de la liste L.
7. Ecrivez un programme en langage C qui permet :
   - De lire N nombres entiers. Les nombres pairs seront stockés dans une liste **LPairs** dans le même ordre de leurs lectures et les nombres impairs seront stockés dans une liste **LImpairs** dans l'ordre inverse de leurs lectures.
   - D'afficher la longueur des listes LPairs et de LImpairs,
   - D'afficher les éléments des listes LPairs et de LImpairs,
   - De supprimer le premier et le dernier élément des listes LPairs et de LImpairs,
   - D'afficher la longueur des listes LPairs et de LImpairs,
   - D'afficher les éléments des listes LPairs et de LImpairs,

---

- Allocation mémoire avec la fonction **malloc()** :

```
type * ptr = (type *) malloc(sizeof(type));
```

- libération de la mémoire avec la fonction **free()** :

```
free(ptr) ;
```

Exemple1:

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

int main(){
int * P ;

      /* Allocation d'un emplacement mémoire qui sera pointé par le
         pointeur P */

P = (int *) malloc(sizeof(int));

      /* Accès à l'emplacement mémoire pointé par le pointeur P */

*P = 10 ;
printf("le contenu de l'emplacement mémoire pointé par P est %d ", *P);

     /* Libération de l'emplacement mémoire pointé par le pointeur P */

free(P) ;

return 0 ;

}
```

Exemple2:

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

typedef struct  Point {
            float  Abs ;
            float  Ord;
} Point ;

int main(){
Point * P ;

       /* Allocation d'un emplacement mémoire qui sera pointé par le
         pointeur P */

P = (Point *) malloc(sizeof(Point));

      /* Accès à l'emplacement mémoire pointé par le pointeur P */

printf ("Donnez l'abscisse du point:\n") ;
scanf("%f",&((*P).Abs)) ; // scanf("%f",&(P->Abs)) ;
printf ("Donnez l'ordonée du point:\n") ;
scanf("%f",&((*P).Abs)) ; // scanf("%f",&(P->Ord))
printf("les coordonnées du point sont %f et %f \n ", P->Abs, P->Ord) ;

     /* libération de l'emplacement mémoire pointé par le pointeur P */

free(P) ;

return 0 ;
}
```