



**2014/202015**  
**SMI-S3**  
**Travaux Dirigés de Langage C**  
**Série N°6**



**Exercice 6.1 :** Ecrire un programme se servant d'une fonction F pour afficher la table de valeurs de la fonction définie par

$$f(x) = \sin(x) + \ln(x) - \sqrt{x}$$

où x est un entier compris entre 1 et 10.

**Exercice 6.2 :**

La fonction LIRE\_TAB à trois paramètres TAB, N et NMAX lit la dimension N et les composantes d'un tableau TAB du type **int**. La dimension N doit être inférieure à NMAX. Implémenter la fonction LIRE\_TAB en choisissant bien le type des paramètres.

**Exemple:** Pour un appel par

**LIRE\_TAB(T, &N, 10);**

la fonction se comportera comme suit:

**Dimension du tableau (max.10): 11**

**Dimension du tableau (max.10): 4**

**Elément[0] : 43**

**Elément[1] : 55**

**Elément[2] : 67**

**Elément[3] : 79**

**Exercice 6.3**

Ecrire la fonction ECRIRE\_TAB à deux paramètres TAB et N qui affiche N composantes du tableau TAB du type **int**.

**Exemple:** Le tableau T lu dans l'exemple ci-dessus sera affiché par l'appel :

**ECRIRE\_TAB(T, N);**

et sera présenté comme suit:

**43 55 67 79**

**Exercice 6.4**

Ecrire la fonction SOMME\_TAB qui calcule la somme des N éléments d'un tableau TAB du type **int**. N et TAB sont fournis comme paramètres; la somme est retournée comme résultat du type **long**.

**Exercice 6.5**

A l'aide des fonctions des exercices précédents, écrire un programme qui lit un tableau A d'une dimension inférieure ou égale à 100 et affiche le tableau et la somme des éléments du tableau.

**Exercice 6.6**

a) Ecrire la fonction LIRE\_DIM à quatre paramètres L, LMAX, C, CMAX qui lit les dimensions L et C d'une matrice à deux dimensions. Les dimensions L et C doivent être inférieures à LMAX respectivement CMAX.

b) Ecrire la fonction LIRE\_MATRICE à quatre paramètres MAT, L, C, et CMAX qui lit les composantes d'une matrice MAT du type **int** et de dimensions L et C.

Implémenter les fonctions en choisissant bien le type des paramètres et utiliser un dialogue semblable à celui de LIRE\_TAB.



**2014/202015**  
**SMI-S3**  
**Corrigé des Travaux Dirigés de Langage C**  
**Série N°6**



**Exercice 6.1**

```
#include <stdio.h>
#include <math.h>
main() {
    /* Prototypes des fonctions appelées */
    double F(int X);
    /* Variables locales */
    int I;
    /* Traitements */
    printf("\tX\tF(X)\n");
    for (I=1 ; I<=10 ; I++)
        printf("\t%d\t%f\n", I, F(I));
    return 0;
}
double F(int X)
{
    return sin(X)+log(X)-sqrt(X);
}
```

**Exercice 6.2**

```
void LIRE_TAB (int *TAB, int *N, int NMAX)
{
    /* Variables locales */
    int I;
    /* Saisie de la dimension du tableau */
    do
    {
        printf("Dimension du tableau (max.%d) : ", NMAX);
        scanf("%d", N);          /* Attention: écrire N et non &N ! */
    }
    while (*N<0 || *N>NMAX);
    /* Saisie des composantes du tableau */
    for (I=0; I<*N; I++)
    {
        printf("Elément[%d] : ", I);
        scanf("%d", TAB+I);
    }
}
```

**Exercice 6.3**

```
void ECRIRE_TAB (int *TAB, int N)
{
    /* Affichage des composantes du tableau */
    while(N)
    {
        printf("%d ", *TAB);
        TAB++;
        N--;
    }
    printf("\n");
}
```

#### Exercice 6.4

```
long SOMME_TAB(int *TAB, int N)
{
    /* Variables locales */
    long SOMME = 0;
    /* Calcul de la somme */
    while(N)
    {
        SOMME += *TAB;
        TAB++;
        N--;
    }
    return SOMME;
}
```

#### Exercice 6.5

```
#include <stdio.h>
main()
{
    /* Prototypes des fonctions appelées */
    void LIRE_TAB (int *TAB, int *N, int NMAX);
    void ECRIRE_TAB (int *TAB, int N);
    long SOMME_TAB(int *TAB, int N);
    /* Variables locales */
    int T[100];          /* Tableau d'entiers */
    int DIM;             /* Dimension du tableau */
    /* Traitements */
    LIRE_TAB (T, &DIM, 100);
    printf("Tableau donné : \n");
    ECRIRE_TAB (T, DIM);
    printf("Somme des éléments du tableau : %ld\n", SOMME_TAB(T, DIM) );
    return 0;
}

void LIRE_TAB (int *TAB, int *N, int NMAX)
{
    ...
}
void ECRIRE_TAB (int *TAB, int N)
{
    ...
}
long SOMME_TAB(int *TAB, int N)
{
    ...
}
```

#### Exercice 6.6

```
void LIRE_DIM (int *L, int LMAX, int *C, int CMAX)
{
    /* Saisie des dimensions de la matrice */
    do
    {
        printf("Nombre de lignes de la matrice (max.%d) : ",LMAX);
        scanf("%d", L);
    }
    while (*L<0 || *L>LMAX);
    do
    {
        printf("Nombre de colonnes de la matrice (max.%d) : ",CMAX);
```

```

    scanf("%d", C);
}
while (*C<0 || *C>CMAX);
}

```

b) Ecrire la fonction LIRE\_MATRICE à quatre paramètres MAT, L, C, et CMAX qui lit les composantes d'une matrice MAT du type int et de dimensions L et C.

```

void LIRE_MATRICE (int *MAT, int L, int C, int CMAX)
{
    /* Variables locales */
    int I,J;
    /* Saisie des composantes de la matrice */
    for (I=0; I<L; I++)
        for (J=0; J<C; J++)
        {
            printf("Elément[%d][%d] : ", I, J);
            scanf("%d", MAT + I*CMAX + J);
        }
}

```