

Université Cadi Ayad- Marrakech

Faculté des Sciences – Semlalia

Département Informatique

TD/TP4

Programmation en Langage C SMI3

Exercice 1 :

```
#include<stdio.h>

#include<stdlib.h>

void Afficher_Carre(int *p,int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("%d ",p[i]*p[i]);
    printf("\n");
}

void Carre(int a)
{
    printf("%d ",a*a);
}

int main()
{
    int t[100];
    int i,n;
    printf("Entrez le nombre des elements du tableau:\n");
    scanf("%d",&n);
    printf("Entrez les elements du tableau:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&t[i]);
```

```

    }

    printf("En utilisant la fonction Afficher_Carre:\n");

    Afficher_Carre(t,n);

    printf("En utilisant la fonction Carre:\n");

    for(i=0;i<n;i++)

        Carre(t[i]);

    printf("\n");

    return 0;
}

```

Exercice 2 :

```

#include <stdio.h>
main()
{
    /* Prototypes des fonctions appelées */
    long SOMME_MATRICE (int *MAT, int L, int C, int CMAX);
    void LIRE_DIM      (int *L, int LMAX, int *C, int CMAX);
    void LIRE_MATRICE  (int *MAT, int L, int C, int CMAX);
    void ECRIRE_MATRICE (int *MAT, int L, int C, int CMAX);

    /* Variables locales */
    int M[30][30]; /* Matrice d'entiers */
    int L, C;      /* Dimensions de la matrice */
    /* Traitements */
    LIRE_DIM (&L, 30, &C, 30);
    LIRE_MATRICE ( (int*)M, L,C,30);
    printf("Matrice donnée : \n");
    ECRIRE_MATRICE ( (int*)M, L,C,30);
    printf("Somme des éléments de la matrice : %ld\n",
           SOMME_MATRICE( (int*)M, L,C,30));
    return (0);
}

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);
}

```

```

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);
        }
}

void ECRIRE_MATRICE (int *MAT, int L, int C, int CMAX)
{
    /* Variables locales */
    int I,J;

    /* Affichage des composantes de la matrice */
    for (I=0; I<L; I++)
    {
        for (J=0; J<C; J++)
            printf("%7d", *(MAT + I*CMAX + J));
        printf("\n");
    }
}

long SOMME_MATRICE(int *MAT, int L, int C, int CMAX)
{
    /* Variables locales */
    int I,J;
    long SOMME = 0;
    /* Calcul de la somme */
    for (I=0; I<L; I++)
        for (J=0; J<C; J++)
            SOMME += *(MAT + I*CMAX + J);
    return SOMME;
}

```

Exercice 3:

```

#include <stdio.h>
int acker (int m, int n)
{
    if ( (m<0) || (n<0) )
        return(0) ;
    else if (m==0)
        return (n+1) ;
    else if (n==0)
        return ( acker(m-1,1) ) ;
    else
        return acker ( m-1, acker(m,n-1) ) ;
}

main()
{
    int acker (int, int) ;
    int m, n ;
    printf ("donnez m et n : ") ;
    scanf ("%d %d", &m, &n) ;
    printf ("acker ( %d,%d) = %d", m, n, acker(m,n) ) ;
}

```