

Chapitre 2 :

Corrigé du TP (Série n° 2)

Les fonctions et les procédures

Exercice 1

- 1. But de l'exercice:** comparaison entre un programme avec et sans procédures et fonctions
- Donner la solution à une (petite) partie bien définie d'un problème plus complexe.
 - Meilleure lisibilité
 - Diminution du risque d'erreurs
 - Possibilité de tests sélectifs
 - Dissimulation des méthodes : connaître l'effet seulement sans les détails.
 - Réutilisation de modules déjà existants
 - Simplicité de l'entretien
 - Favorisation du travail en équipe.

2. Solution1:

Sans l'utilisation des fonctions et des procédures.

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  main ()
4  {
5  float A[100][100], B[100][100], Matsom[100][100], Matdif[100][100];
6  int i,j,M,N;
7  printf ("ce programme calcule la somme et la difference de deux matrices A et B \n dont la
      taille est inferieur à 100. \n Les deux matrices doivent avoir les memes dimensions
      \n\n");
8  printf ("Donner la taille des matrices A, B \n");
9  printf (" Nombre de lignes : ");
10 scanf ("%d", &N);
11 printf (" \n nombre de colonnes : ");
12 scanf ("%d", &M);

13 printf ("\n Entrer les elements de la matrice A");
14 for (i=0; i<N; i++)
15 {
16     for (j=0; j<M; j++)
17     {
18         printf ("\n Donner l'element (%d, %d) : ", i, j);
19         scanf ("%f", &A[i][j]);
20     }
21 }
22 printf ("\n Entrer les elements de la matrice B");
23 for (i=0; i<N; i++)
24 {
25     for (j=0; j<M ; j++)
```

```

26     {
27         printf ( "\n Donner l'element (%d, %d) : " , i , j );
28         scanf ( "%f" , &B[i][j]);
29     }
30 }
31 for(i=0;i<N; i++)
32 {
33     for (j=0;j<M; j++)
34     {
35         Matsom[i][j]=A[i][j]+B[i][j];
36         Matdif[i][j]=A[i][j]- B[i][j];
37     }
38 }
39 system ( "cls" );
40 printf ( "\n La matrice A est \n" );
41     for ( i = 0 ; i < N ; i++ )
42     {
43         for (j=0 ; j < M ; j ++ )
44             printf ( "%12f" , A[i][j]);
45         printf ( "\n" );
46     }
47 Printf ( "\n La matrice B est \n" );
48     for ( i=0 ; i<N ; i++ )
49     {
50         for (j=0 ; j<M ; j++ )
51             printf ( "%12f" , B[i][j]);
52         printf ( "\n" );
53     }
54 printf ( "\n La matrice somme ( A + B ) est \n" );
55     for ( i=0 ; i<N ;i++ )
56     {
57         for (j=0 ; j<M ; j++ )
58             printf ( "%12f" , Matsom[i][j]);
59         printf ( "\n" );
60     }
61 printf ( "\n la matrice Difference ( A- B ) est \n" );
62     for ( i=0 ; i<N ; i++ )
63     {
64         for (j=0 ; j<M ; j++ )
65             printf ( "%12f" , Matdif [i][j]);
66         printf ( "\n" );
67     }
68 }

```

3.Solution 2 : Avec des fonctions et des procédures

```

1     #include<stdio.h>
2     #include<stdlib.h>
3
4     int i,j;
5     void liremat ( float x[][100], int n, int m )
6     {
7         for( i=0; i<n ; i++)
8             for (j=0 ; j<m ; j++)

```

```

    {
        Printf ( "\n donner l'element (%d, %d) : ", i, j );
        Scanf ( "%f", &x[i][j]);
    }
}

void afficheMAT ( float x[][100],int n , int m)
{
    for (i=0 ; i<n ; i++)
    {
        for (j=0 ; j<m ; j++)
            printf ( "%12f", x[i][j]);
        printf ( "\n");
    }
}

3     main()
4     { float A[100][100], B[100][100], Matsom[100][100], Matdif[100][100];
5       int M,N;
6       Printf ("ce programme calcule la somme et la difference de deux matrice A et B \n dont
la
        taille est inferieur à 100. \n Les deux matrices doivent avoir les memes dimensions.
        \n\n");
7       printf ( "Donner la taille des matrices A, B \n");
8       printf ( " Nombre de lignes : ");
9       scanf ( "%d", & N);
10      printf ( " \n nombre de colonnes : ");
11      scanf ( "%d", & M);
12      printf ( "\n Entrer les elements de la matrice A");
13      lireMAT ( A, N, M);
14      printf ( "\n Entrer les elements de la matrice B");
15      lireMAT( B, N,M);

16      for (i=0;i<N; i++)
17      {
18          for (j=0 ; j<M ; j++)
19          {
20              Matsom[i][j]=A[i][j]+B[i][j];
21              Matdif[i][j]=A[i][j] - B[i][j];
22          }
23      }

24
25      printf ( "\n La matrice A est \n");
26      system ("cls");
27      afficheMAT(A, N, M);
28      printf ( "\n La matrice B est \n" );
29      afficheMAT (B, N, M);
30      printf ( "\n La matrice Difference (A- B) est \n");
31      afficheMAT (Matdif, N, M);
32      printf ( "\n La matrice somme (A + B) est \n");
33      afficheMAT ( Matsom, N, M);
34      }

```

Exercice 2

But de l'exercice : Notion de la fonction récursive, variables locales et variable globales

```
void convert(int ,int );
main()
{
    int n, base;
    printf ( "Donner un entier \n" );
    scanf ( "%d",&n );
    printf ( "Choisir une base: 2, 8, 16 \n " );
    scanf ("%d" ,&base);
    if ( base==2 ) { printf ("La valeur de %d en binaire est : ", n ); convert ( n , base); }
    if (base==8) { printf ( "La valeur de %d en octal est : ", n); convert(n , base); }
    if(base==16) { printf ( "La valeur de %d en hexadecimal est : ", n); convert(n , base); }
}

void convert ( int x , int y )
{
    int z, r=0;
    if (x!= 0)
    {
        convert ( x/y , y ); z = x % y ;
        if ( z == 10 ){ r = 1 ; printf ( "A" );}
        if ( z == 11 ){ r = 1 ; printf ( "B" );}
        if ( z == 12 ){ r = 1 ; printf ( "C" );}
        if ( z == 13 ){ r = 1 ; printf ( "D" );}
        if ( z == 14 ){ r = 1 ; printf ( "E" );}
        if ( z == 15 ){ r = 1 ; printf ( "F " );}
        if ( r == 0 ) printf ( "%d" , z );
    }
}
```

Exercice 3

But de l'exercice : Eviter la répétition du même code s'il est utilisé plusieurs fois dans le même programme, faire la comparaison entre la fonction récursive et la fonction itérative et la procédure qui appelle une autre fonction. La solution est donnée pour 3 vecteurs.

```
#include<stdio.h>
#include<stdlib.h>
int i;
void lirevecteur ( int x [], int n )
{ for (i=0 ; i<n ; i++)
    { printf ( "\n Donner l'element (%d) : ", i+1);
      scanf ("%d" ,&x[i]);
    }
}
void affiche ( int tab[], int n)
{
    for ( i=0;i<n;i++)
    printf ( "tab[%d] = %d \n" , i , tab[i] );
    printf ( "\n" );
}
```

```

}

bool tablo_ego (int t1[], int t2[], int n, int i) /* fonction récursive*/
{
    bool result;
    if (i == n-1) result = (t1[i]==t2[i]);
    else result = (t1[i]==t2[i]) && tablo_ego (t1, t2, n, i+1);
    return result;
}

void compare_rec (int t1[], int t2[], int n)
{
    if (tablo_ego (t1, t2, n, 0)) printf ("sont egaux.\n");
    else printf ("sont differents.\n");
}

void compare_iter (int t1[], int t2[], int n)
{
    int i;
    bool pas_fini=1;
    for (i=0; i<n && pas_fini; i++)
        if (t1[i]!=t2[i]) pas_fini=0;
    if (pas_fini) printf ("sont egaux.\n");
    else printf ("sont differents.\n");
}

main ()
{int N, A[30], B[30], C[30];
    printf ("Donner la longueur du vecteur : ");
    scanf ("%d", &N);
    printf ("Le tableau A\n"); lirevecteur (A, N); printf ("\n");
    printf ("Le tableau B\n"); lirevecteur (B, N); printf ("\n");
    printf ("Le tableau C\n"); lirevecteur (C, N); printf ("\n");
system ("cls");
    printf ("Voici le tableau A\n"); affiche (A, N);
    printf ("Voici le tableau B\n"); affiche (B, N);
    printf ("Voici le tableau C\n"); affiche (C, N);

printf ("Version recursive : \n");
    printf (" Les deux tableaux A et B "); compare_rec (A, B, N);
    printf (" Les deux tableaux A et C "); compare_rec (A, C, N);
    printf (" Les deux tableaux C et B "); compare_rec (C, B, N);

printf ("\n Version iterative : \n");
    printf (" Les deux tableaux A et B "); compare_iter (A, B, N);
    printf (" Les deux tableaux A et C "); compare_iter (A, C, N);
    printf (" Les deux tableaux C et B "); compare_iter (C, B, N);
}

```