



2014/202015
SMI-S3
Travaux Dirigés de Langage C
Série N°4



Exercice 1

Ecrire un programme qui lit la dimension N d'un tableau T du type **int** (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Ranger ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide. Afficher le tableau résultant.

Idée: Echanger les éléments du tableau à l'aide de deux indices qui parcourent le tableau en commençant respectivement au début et à la fin du tableau et qui se rencontrent en son milieu.

Exercice 2

Ecrire un programme qui lit la dimension N d'un tableau T du type **int** (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau TPOS et toutes les valeurs strictement négatives dans un troisième tableau TNEG. Afficher les tableaux TPOS et TNEG.

Exercice 3

Ecrire un programme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Exercice 4

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type **int** (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

Exercice 5

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type **int** (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de chaque ligne et de chaque colonne en n'utilisant qu'une variable d'aide pour la somme.



2014/202015

SMI-S3

Corrigé des Travaux Dirigés de Langage C Série N°4



Solution 1

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int T[50];           /* tableau donné */
    int N;               /* dimension */
    int I,J;             /* indices courants */
    int AIDE;            /* pour l'échange */

    /* Saisie des données */
    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N );
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &T[I]);
    }
    /* Affichage du tableau */
    printf("Tableau donné : \n");
    for (I=0; I<N; I++)
        printf("%d ", T[I]);
    printf("\n");
    /* Inverser le tableau */
    for (I=0, J=N-1 ; I<J ; I++,J--)
        /* Echange de T[I] et T[J] */
        {
            AIDE = T[I];
            T[I] = T[J];
            T[J] = AIDE;
        }
    /* Edition des résultats */
    printf("Tableau résultat : \n");
    for (I=0; I<N; I++)
        printf("%d ", T[I]);
    printf("\n");
    return 0;
}
```

Solution 2

```
#include <stdio.h>
main()
{
    /* Déclarations */
    /* Les tableaux et leurs dimensions */
    int T[50], TPOS[50], TNEG[50];
    int N, NPOS, NNEG;
    int I; /* indice courant */
}
```

```

/* Saisie des données */
printf("Dimension du tableau (max.50) : ");
scanf("%d", &N );
for (I=0; I<N; I++)
{
    printf("Élément %d : ", I);
    scanf("%d", &T[I]);
}
/* Affichage du tableau */
printf("Tableau donné :\n");
for (I=0; I<N; I++)
    printf("%d ", T[I]);
printf("\n");
/* Initialisation des dimensions de TPOS et TNEG */
NPOS=0;
NNEG=0;
/* Transfer des données */
for (I=0; I<N; I++)
{
    if (T[I]>0)
    {
        TPOS[NPOS]=T[I];
        NPOS++;
    }
    if (T[I]<0)
    {
        TNEG[NNEG]=T[I];
        NNEG++;
    }
}
/* Edition du résultat */
printf("Tableau TPOS :\n");
for (I=0; I<NPOS; I++)
    printf("%d ", TPOS[I]);
printf("\n");
printf("Tableau TNEG :\n");
for (I=0; I<NNEG; I++)
    printf("%d ", TNEG[I]);
printf("\n");
return 0;
}

```

Solution 3

```

#include <stdio.h>
main()
{
    /* Déclarations */
    int U[50], V[50];          /* tableaux donnés */
    int N;                     /* dimension */
    int I;                      /* indice courant */
    long PS;                    /* produit scalaire */
    /* Saisie des données */
    printf("Dimension des tableaux (max.50) : ");
    scanf("%d", &N );
    printf("*** Premier tableau **\n");
    for (I=0; I<N; I++)

```

```

    {
        printf("Elément %d : ", I);
        scanf("%d", &U[I]);
    }
    printf("*** Deuxième tableau **\n");
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &V[I]);
    }
    /* Calcul du produit scalaire */
    for (PS=0, I=0; I<N; I++)
        PS += (long)U[I]*V[I];
    /* Edition du résultat */
    printf("Produit scalaire : %ld\n", PS);
    return 0;
}

```

Solution 4

```

#include <stdio.h>
main()
{
    /* Déclarations */
    int T[50][50];      /* tableau donné */
    int L, C;           /* dimensions */
    int I, J;           /* indices courants */
    long SOM;           /* somme des éléments - type long à cause */
                        /* de la grandeur prévisible du résultat. */
    /* Saisie des données */
    printf("Nombre de lignes (max.50) : ");
    scanf("%d", &L );
    printf("Nombre de colonnes (max.50) : ");
    scanf("%d", &C );
    for (I=0; I<L; I++)
        for (J=0; J<C; J++)
        {
            printf("Element[%d][%d] : ",I,J);
            scanf("%d", &T[I][J]);
        }
    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (I=0; I<L; I++)
    {
        for (J=0; J<C; J++)
            printf("%7d", T[I][J]);
        printf("\n");
    }
    /* Calcul de la somme */
    for (SOM=0, I=0; I<L; I++)
        for (J=0; J<C; J++)
            SOM += T[I][J];
    /* Edition du résultat */
    printf("Somme des éléments : %ld\n", SOM);
    return 0;
}

```

Solution 5

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int T[50][50]; /* tableau donné */
    int L, C;      /* dimensions */
    int I, J;      /* indices courants */
    long SOM;      /* somme des éléments - type long à cause */
                  /* de la grandeur prévisible des résultats. */
    /* Saisie des données */
    printf("Nombre de lignes (max.50) : ");
    scanf("%d", &L );
    printf("Nombre de colonnes (max.50) : ");
    scanf("%d", &C );
    for (I=0; I<L; I++)
        for (J=0; J<C; J++)
        {
            printf("Elément[%d][%d] : ",I,J);
            scanf("%d", &T[I][J]);
        }
    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (I=0; I<L; I++)
    {
        for (J=0; J<C; J++)
            printf("%7d", T[I][J]);
        printf("\n");
    }
    /* Calcul et affichage de la somme des lignes */
    for (I=0; I<L; I++)
    {
        for (SOM=0, J=0; J<C; J++)
            SOM += T[I][J];
        printf("Somme - ligne %d : %ld\n",I,SOM);
    }
    /* Calcul et affichage de la somme des colonnes */
    for (J=0; J<C; J++)
    {
        for (SOM=0, I=0; I<L; I++)
            SOM += T[I][J];
        printf("Somme - colonne %d : %ld\n",J,SOM);
    }
    return 0;
}
```