

Nom, prénom : _____ Groupe : _____

TP 10 : Tableaux à 1 dimension

Préparation

1. Une matrice est une représentation sous forme de tableau de plusieurs nombres. Voici par exemple une matrice 3x3 :

$$\begin{bmatrix} 12 & 5 & 4 \\ 1 & -5 & 4 \\ 7 & 8 & 10 \end{bmatrix}$$

Il est possible d'additionner deux matrices en additionnant les termes un à un :

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} = \begin{bmatrix} a+a' & b+b' \\ c+c' & d+d' \end{bmatrix}$$

Par exemple :

$$\begin{array}{c} A \\ \begin{bmatrix} 12 & 5 & 4 \\ 1 & -5 & 4 \\ 7 & 8 & 10 \end{bmatrix} \end{array} + \begin{array}{c} B \\ \begin{bmatrix} 2 & 7 & 1 \\ 8 & 5 & -7 \\ 2 & 8 & -4 \end{bmatrix} \end{array} = \begin{array}{c} C \\ \begin{bmatrix} 14 & 12 & 5 \\ 9 & 0 & -3 \\ 9 & 16 & 6 \end{bmatrix} \end{array}$$

Nous allons programmer ce calcul en C. La matrice A est mémorisée dans un tableau de la façon suivante :

```
int A[3][3]={12,5,4,1,-5,4,7,8,10};
```

Ecrire la déclaration de la deuxième matrice B :

Ecrire les instructions permettant de réaliser l'addition des deux matrices (A et B) et de mémoriser le résultat dans une matrice C :

2. Une grille de morpion est mémorisée dans un tableau à deux dimensions :

```
char grille [3][3];
```

Voici un exemple de déclaration avec la grille associée :

```
char grille [3][3]={'O',' ','O',' ','X',' ',' ',' ','X'};
```

O		O
	X	
		X

Dessiner la grille correspondant à la déclaration suivante :

```
char grille [3][3]={'X','O','O',' ','X','O','X',' ','X'};
```


Donner la déclaration correspondant à la grille suivante :

X		O
O	O	X
X	X	

```
char grille [3][3]=
```

Dans la grille ci-dessous, donner les indices de la croix et du rond :

	X	
O		

Indices de la croix : grille [][]

Indices du rond : grille [][]

Travail pratique

Exercice 1 : Compléter et tester le programme fourni sur le document réponse afin qu'il affiche la somme des deux matrices.

Exercice 2 : Ecrire un programme principal qui déclare une grille de morpion conformément à l'exercice de la préparation.

a. Ajouter à ce programme les instructions permettant de tester si la grille comporte une solution gagnante en diagonale. Si cette solution existe, le programme affiche qui est le vainqueur (ronds ou croix).

b. Ajouter à ce programme les instructions permettant de tester si la grille comporte une solution gagnante verticale. Si cette solution existe, le programme affiche qui est le vainqueur (ronds ou croix).

Exercice 3 : On fournit les deux fonctions suivantes qui permettent respectivement de charger et d'enregistrer une image au format BMP comportant 320 x 200 pixels en 16 millions de couleurs.

```
char Read_BMP_320x200 (char *filename, unsigned char *pImage)
{
    FILE *file = fopen(filename,"rb");
    if (file==NULL) return (-1);
    if (fseek (file,54,0)!=0) return (-2);
    if (fread (pImage,3,320*200,file)!=320*200) return (-3);
    fclose (file);
    return (0);
}

char Write_BMP_320x200 (char *filename,unsigned char *pImage)
{
    FILE *file = fopen(filename,"rb+");
    char Header[54]={66,77,54,-18,2,0,0,0,
                    0,0,54,0,0,0,40,0,
                    0,0,64,1,0,0,-56,0,
                    0,0,1,0,24,0,0,0,
                    0,0,0,-18,2,0,0,0,
                    0,0,0,0,0,0,0,0,
                    0,0,0,0,0,0};

    if (file==NULL) return (-1);
    if (fwrite (Header,1,54,file)!=54) return (-2);
    if (fwrite (pImage,3,64000,file)!=320*200) return (-3);
    fclose (file);
    return (0);
}
```

Voici un exemple d'utilisation de ces fonctions :

```
void main(void)
{
    unsigned char Im[200][320][3];
    int x,y,c;

    // Chargement de l'image

    if (Read_BMP_320x200("logoIUT.bmp",&Im[0][0][0])!=0)
    {
        printf ("Erreur lors de l'ouverture");
        exit (0);
    }

    // Modification de l'image
    for (x=0;x<320;x++)
        for (y=0;y<100;y++)
            Im[y][x][0]=255;

    // Sauvegarde de l'image

    if (Write_BMP_320x200("logoIUT2.bmp",&Im[0][0][0])!=0)
    {
        printf ("Erreur lors de l'écriture");
        exit (0);
    }
}
```

- a. Que fait ce programme ?
- b. Modifier le programme de façon à ce qu'il enregistre une image toute blanche.
- c. Modifier le programme de façon à ce qu'il enregistre une image toute noire.
- d. Modifier le programme de façon à ce qu'il enregistre une image toute rouge.
- e. Modifier le programme de façon à ce qu'il effectue un miroir vertical de l'image originale.

Compte-rendu à compléter

Exercice 1 :

```
void main()
{
    // Déclaration des variables

    int A[3][3]={-7,2,9,9,-5,1,7,7,-3};           // Matrice A
    int B[3][3]={65,4,3,9,12,3,1,1,82};           // Matrice B
                                                    // Matrice C

    // Calcul de la somme

    // Affichage du résultat

    for (x=0;x<3;x++)                             // Pour les lignes
    {
        for (y=0;y<3;y++)                         // Pour les colonnes
            printf ("%d\t",C[x][y]);              // Affiche la valeur
        printf ("\n");                             // Ligne suivante
    }
}
```

Exercice 2 :

a. Solution gagnante en diagonale :

b. Solution gagnante verticale :

Exercice 3 :

a. Explication :

b. Image blanche :

```
                                // Modification de l'image
for (x=0;x<320;x++)
    for (y=0;y<100;y++)
    {

    }

}
```

d. Image rouge :

```
                                // Modification de l'image
for (x=0;x<320;x++)
    for (y=0;y<100;y++)
    {

    }

}
```

e. Miroir vertical :