SEQUENCE 5– TABLEAUX

SEANCE 7 - DECOUVERTE TABLEAUX A 2 DIMENSIONS

OBJECTIFS

- Manipuler un tableau à 2 dimensions

A LIRE :

```
int[,] tabNotes = new int[,] { { 75, 80, 56, 89 }, { 98, 84, 85, 7 }, { 65, 88, 76, 56 } };
Console.WriteLine(tabNotes.Length);   // 12
Console.WriteLine(tabNotes.Rank);  //2
Console.WriteLine(tabNotes.GetLength(0)); //3 car c'est d'abord un tableau de 3 éléments
Console.WriteLine(tabNotes.GetLength(1)); //4 car chaque élément est constitué de 4 éléments
Console.WriteLine(tabNotes[0, 3]); //89
Console.WriteLine(tabNotes[2, 1]);   //88
Console.WriteLine(tabNotes[1, 1]);   // ?????
```

0			
0	1	2	3
75	80	56	89

1			
0	1	2	3
98	84	85	7

2			
0	1	2	3
65	88	76	56

Avec une représentation plus classique, on peut dire que ce tableau est composée de 3 lignes et 4 colonnes. Rem : Ce genre de représentation n'est pas possible au-delà de 2 dimensions.

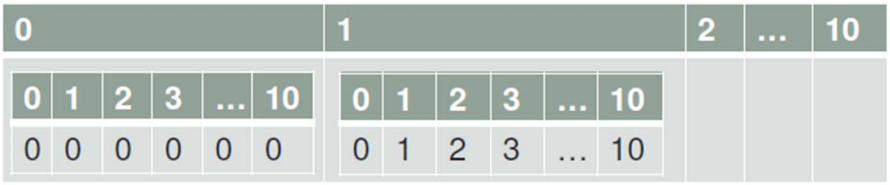
	0	1	2	3
0	75	80	56	89
1	98	84	85	7
2	65	88	76	56

On peut aussi écrire afin de visualiser plus facilement les lignes et les colonnes.

```
int[,] tabNotes = new int[,] {
                              { 75, 80, 56, 89 },
                              { 98, 84, 85, 7 },
                              { 65, 88, 76, 56 } };
```

Exo 1 : INITIALISATION ET AFFICHAGE CASE PAR CASE – TABLES DE MULTIPLICATION

1. Au sein du répertoire « **Sequence_5_Tableaux** » , créez un projet « **Exo1_TablesDeMultiplication**» type console dans une solution, nommée « **Seance7_DecouverteTab2Dim» . Ajoutez le code ci-dessous :**




```
int[,] tables = new int[11, 11];
for (int i = 0; i < tables.GetLength(0); i++)
{
    for (int j = 0; j < tables.GetLength(1); j++)
        tables[i, j] = i * j;
}
```

2. Affichez toutes les tables comme l'exemple donné :

```
TABLE DE  0
---------------
0x0 = 0
0x1 = 0
0x2 = 0
0x3 = 0
0x4 = 0
0x5 = 0
0x6 = 0
0x7 = 0
0x8 = 0
0x9 = 0
0x10 = 0
---------------
TABLE DE  1
---------------
1x0 = 0
1x1 = 1
1x2 = 2
1x3 = 3
1x4 = 4
1x5 = 5
1x6 = 6
1x7 = 7
1x8 = 8
1x9 = 9
1x10 = 10
---------------
TABLE DE  2
---------------
```

3. Programmez un mode entrainement, tirez 2 nombres au hasard, vérifiez la réponse donnée par l'utilisateur. Faites le recommencer en cas d'erreur : il a droit à 3 essais. Après le programme lui donne la bonne réponse.

```
------------------------------
 ENTRAINEMENT
------------------------------
7x8 =  ?  (Q pour quitter)
Essai 1 /3
48
Erreur. Reessaie :
Essai 2 /3
54
Erreur. Reessaie :
Essai 3 /3
23
7x8 : 56
------------------------------
6x3 =  ?  (Q pour quitter)
Essai 1 /3
18
------------------------------
5x7 =  ?  (Q pour quitter)
Essai 1 /3
```

EXO 2 : MATRICE

1. Ajoutez un projet « Exo2_Matrices ».

Dans cet exercice, on travaillera avec des matrices carrées.

Transposée :

$$M \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad {}^{t}M \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Addition :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} + \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{pmatrix}$$

Multiplication :

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

2. Veuillez définir les méthodes :

- `public static void Affiche(int[,] tab)`
- `public static int[,] Additionne(int[,] tab1, int[,] tab2)`
- `public static int [,] Transpose (int[,] tab )`
- `public static int[,] Multiplie(int[,] tab1, int[,] tab2)`

3. Améliorez votre code :
   a. Pour les méthodes Additionne, Transpose et Multiplie : lancez des exceptions si les matrices passées en paramètre ne sont pas carrées ou pas de la même taille

   b. Pour la méthode Affiche : faites une surcharge pour afficher les nombres contenus dans une matrice sur un nombre de caractère spécifié en paramètre afin d'assurer un alignement ! Ex : ci-dessous, les nombres sont alignés sur 3 caractères

```
  2   4   6   8  10
 12  14  16  18  20
 22  24  26  28  30
```