

- 4.1) Écrire un programme en C# qui initialise un tableau de 10 variables de type byte avec les chiffres de 0 à 9 (0, 1, 2, 3, 4, 5, 6, 7, 8 et 9).

```
byte[] Chiffres = new byte[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

- 4.2) Écrire un programme en C# qui demande à l'utilisateur 5 nombres. Il affiche finalement les nombres entrés. L'utilisation d'un tableau est obligatoire.

```
short[] Nombres = new short[5];
for (int i = 0; i < Nombres.Length; i++) {
    Console.Write("Entrer nombre #{0}: ", i + 1);
    Nombres[i] = short.Parse(Console.ReadLine());
}
Console.Write("Les nombres sont : ");
foreach (int nombre in Nombres) //un autre format de for => foreach
{ //pour chaque nombre du tableau ; au lieu d'utiliser une variable
  //itérateur/compteur , elle utilise une variable qui va récupérer
  //le contenu de la case courante du tableau (et non pas uniquement
  //la position comme la variable i de la première boucle for fait.
  //dans ce type de boucle on ne peut pas modifier le pas d'incrément
  //car c'est toujours +1
    Console.Write(nombre + " ");
}
```

- 4.3) Reprendre l'algorithme précédent et affichez plutôt la moyenne des nombres entrés.

```
short[] Nombres = new short[5];
int somme = 0;
for (int i = 0; i < Nombres.Length; i++) {
    Console.Write("Entrer nombre #{0}: ", i + 1);
    Nombres[i] = short.Parse(Console.ReadLine());
}
foreach (int nombre in Nombres) //un autre format de for => foreach
{ //pour chaque nombre du tableau ; au lieu d'utiliser une variable
  //itérateur/compteur , elle utilise une variable qui va récupérer
  //le contenu de la case courante du tableau (et non pas uniquement
  //la position comme la variable i de la première boucle for fait.
  //dans ce type de boucle on ne peut pas modifier le pas d'incrément
  //car c'est toujours +1
    somme += nombre;
}
Console.Write("La moyenne est de {0}", somme/5);
```

- 4.4) Écrire un programme C# qui demande à l'utilisateur 5 nombres. Il affiche ensuite la plus grande valeur entrée en précisant quelle **position** réelle (humaine) elle occupe dans le tableau.

```
short[] Nombres = new short[5];
for (byte i = 0; i < Nombres.Length; i++) {
    Console.Write("Entrer nombre #{0}: ", i + 1);
    Nombres[i] = short.Parse(Console.ReadLine());
}

int max = Nombres[0]; // on suppose que le premier nombre du tableau est le maximum
byte positionMax = 0; // la position de Nombres[0] qu'on suppose que c'est notre max
for (byte i = 1; i < Nombres.Length; i++) { // dans ce cas j'ai besoin de ce format de for car j'aurai besoin du compteur i pour me donner la position
    // je commence à parcourir mon tableau à partir de la deuxième case qui a
    // la position 1 car on va comparer la valeur de la première case (position 0)
    // avec les prochaines cases 1, 2, 3, et 4 pour chercher le maximum
    if (Nombres[i] > max) { // si la case courante est plus grande que Nombres[0]
        max = Nombres[i]; // ceci est notre nouvel maximum alors
        positionMax = i; // et voici sa position
    }
}

Console.Write("Maximum = {0}, position = {1}", max, positionMax + 1);
```

- 4.5) Écrire un programme C# qui remplit aléatoirement de chiffres (compris entre 0 et 9) deux tableaux de départ de 8 éléments. Il crée ensuite un troisième tableau composé de la multiplication des éléments des deux tableaux de même indice et affiche les résultats à l'écran comme suit :

```
byte[] Nombres1 = new byte[8];
byte[] Nombres2 = new byte[8];
byte[] NombresProduit = new byte[8];
Console.WriteLine("---> Remplissage des deux tableaux : Tableau1 & Tableau2");//on peut remplir chaque tableau tout seule dans une boucle à part
for (byte i = 0; i < Nombres1.Length; i++)
{
    do
    {
        Console.Write("Tableau 1 => Entrer nombre #{0}: ", i + 1);
        Nombres1[i] = byte.Parse(Console.ReadLine());
    } while (Nombres1[i] < 0 || Nombres1[i] > 9);
    do
    {
        Console.Write("Tableau 2 => Entrer nombre #{0}: ", i + 1);
        Nombres2[i] = byte.Parse(Console.ReadLine());
    } while (Nombres2[i] < 0 || Nombres2[i] > 9);
}
//Affichage tableau Nombres1
Console.Write("Tableau 1 : [");
for (int j = 0; j < Nombres1.Length - 1; j++) Console.Write($"{Nombres1[j]}, ");
Console.Write($"{Nombres1[Nombres1.Length-1]}\n");
//Affichage tableau Nombres2
Console.Write("Tableau 2 : [");
for (int j = 0; j < Nombres2.Length - 1; j++) Console.Write($"{Nombres2[j]}, ");
Console.Write($"{Nombres2[Nombres2.Length - 1]}\n");
//Affichage tableau NombreProduit qui est la multiplication des tableaux Nombres1 et Nombres2 (on calcul et on affiche en même temps)
Console.Write("Résultats : [");
for (int j = 0; j < NombresProduit.Length - 1; j++) Console.Write($"{Nombres2[j] * Nombres1[j]}, ");
Console.Write($"{Nombres2[Nombres2.Length - 1] * Nombres1[Nombres2.Length - 1]}\n");
```

- 4.6) Écrire un programme C# qui demande à l'utilisateur 9 notes à l'utilisateur (sur 100, non nécessaire de valider). Il retourne ensuite le nombre de notes supérieures à la moyenne de la classe. L'affichage doit être identique à celui-ci.

```
short[] Notes = new short[9]; //9 notes
byte NombreMoyenne = 0; //compteur des notes qui sont supérieures de la moyenne
short somme = 0; //la somme des notes pour calculer la moyenne
for (byte i = 0; i < Notes.Length; i++)
{
    do//entrée des notes entre 0 et 100 (notes/100)
    {
        Console.Write("Entrer note #{0} : ", i + 1);
        Notes[i] = byte.Parse(Console.ReadLine());
    } while (Notes[i] < 0 || Notes[i] > 100);
    somme += Notes[i]; //un fois la note est correcte on l'ajoute à la somme
}
//calcul du nombre des notes qui sont supérieures à la moyenne
for (int i = 0; i < Notes.Length; i++) if (Notes[i] > (somme / 9)) NombreMoyenne++;
Console.Write("Nombre de notes > moyenne = {0}", NombreMoyenne);
```

## Défis : Les tableaux avancées

- 4.7) Écrire un programme en C# qui demande à l'utilisateur 20 chiffres (0 à 9 validés par le système c'est à dire que si on entre 10, il faut demander à nouveau) et qui construit ensuite un tableau contenant le nombre d'occurrence pour chaque chiffre de 0 à 9 et l'affiche à l'écran.

```
byte[] TabChiffres = new byte[20];
byte[] TabOccurrences = new byte[10]; // de 0 à 9 => c'est 10 chiffres
for (byte i = 0; i < TabChiffres.Length; i++)
{
    do // entrée des notes entre 0 et 9
    {
        Console.Write("Entrer chiffre : ");
        TabChiffres[i] = byte.Parse(Console.ReadLine());
        if (TabChiffres[i] < 0 || TabChiffres[i] > 9) Console.WriteLine(TabChiffres[i] + " n'est pas un chiffre");
    } while (TabChiffres[i] < 0 || TabChiffres[i] > 9);
    TabOccurrences[TabChiffres[i]]++; // le nombre entré par l'utilisateur est la position de tableau des occurrences:
                                    // exemple TabChiffres[i] = 2 => je vais incrémenter la valeur de TabOccurrences[2]
                                    // bref: TabChiffres[position chiffre] = chiffre, TabOccurrences[chiffre] = nombre de fois(occurrence du chiffre)
}
for (int i = 0; i < TabOccurrences.Length; i++) Console.WriteLine(i + " --> " + TabOccurrences[i]);
```

- 4.8) Écrire un programme en C# qui demande à l'utilisateur une quantité déterminée par vous de nombres à entrer et qui affiche ensuite ces nombres triés de façon décroissante.

```
Console.Write("Combien de nombre vous voulez saisir ? : ");
int nbr = int.Parse(Console.ReadLine()); //récupération de quantité des nombres que l'utilisateur veut saisir => qui sera la taille du tableau
int[] TabNombres = new int[nbr];
int tmp = 0; //variable temporaire qui servira pour la permutation
for (int i = 0; i < TabNombres.Length; i++)
{
    Console.Write("Entrer nombre {0} : ", i+1);
    TabNombres[i] = int.Parse(Console.ReadLine());
}
//tri du tableau décroissant: en comparant la case courante et sa suivante si la case est inférieure à sa suivante on les permute
for(int j = 0; j < TabNombres.Length; j++)//on reprend cela TabNombres.Length fois pour s'assurer que la tableau est complètement trié
{
    for(int i = 0; i<TabNombres.Length-1;i++)
    {
        if (TabNombres[i] <= TabNombres[i+1])
        {
            tmp = TabNombres[i+1];
            TabNombres[i+1] = TabNombres[i];
            TabNombres[i] = tmp;
        }
    }
}
for (int i = 0; i < TabNombres.Length; i++)//Affichage du tableau décroissant
{
    Console.Write(TabNombres[i] + " ");
}
```