

Les boucles « do... while »

- 3.1) Écrire un programme C# avec une boucle « do... while » et un « switch » qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne. Il affiche "Ajouter une note" pour le choix 1, "Modifier une note" pour le choix 2 et "Supprimer une note" pour le choix 3.

```
Faite votre choix de menu [1, 2 ou 3] :
```

```
9
```

```
1
```

```
Ajouter une note
```

```
Faite votre choix de menu [1, 2 ou 3] :
```

```
2
```

```
Modifier une note
```

```
Faite votre choix de menu [1, 2 ou 3] :
```

```
3
```

```
Supprimer une note
```

- 3.2) Écrire un programme C# qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

```
Entrez un nombre entre 10 et 20 inclusivement :
```

```
90
```

```
Plus petit!
```

```
0
```

```
Plus grand!
```

```
15
```

```
Bravo!
```

3.3) Analysez le programme C# suivant et répondez aux questions suivantes :

```
int x1, x2, result;
Console.WriteLine("Entrez deux entier :");
x1 = int.Parse( Console.ReadLine() );
x2 = int.Parse( Console.ReadLine() );

do {
    result = x1 % x2;
    x1 = x2;
    x2 = result;
} while( result != 0 );

Console.WriteLine($"Le résultat est { x1 }");
```

3.3.1 Quelle est l'instruction qui permet de modifier le résultat du test de sortie de boucle (dans le while) ?

3.3.2 Quel sera le résultat en supposant que l'utilisateur entre les valeurs 30 et 42.
Exécutez le programme à la main (pour vous aider, construisez le tableau de trace des variables).

3.3.3 Quel sera le résultat en supposant que l'utilisateur entre les valeurs 35 et 6.
Exécutez le programme à la main (pour vous aider, construisez le tableau de trace des variables).

3.3.4 Quel est le calcul réalisé par ce programme, à quoi sert ce programme ?

Les boucles « while »

- 3.4) Écrire un programme C# qui demande un nombre à l'utilisateur et calcule sa factorielle.

```
Entrez un nombre entier : 6  
La factorielle de 6 est 720
```

- 3.5) Écrivez un programme de type devinette qui tire un chiffre au hasard entre 0 et 9 et demande à l'utilisateur de deviner ce nombre jusqu'à ce qu'il le trouve. Suivez les étapes suivantes :

1. Tirer le nombre au hasard
2. Demander un nombre à l'utilisateur
3. Tant que le nombre lu est différent du nombre tiré au hasard
 - A) Demande un nombre à l'utilisateur
 - B) Compter le nombre de boucles (essaie) effectuées
4. Afficher un message de réussite ainsi que le nombre d'essais nécessaires
5. N.B. Pour tirer un nombre aléatoire : `new Random().Next(0,9)`

- 3.6) Reprenez le programme précédent (3.5) mais :

1. En faisant tirer au hasard un nombre entre 0 et 50
2. Afficher « trop bas/haut » si la valeur devinée est trop petite/grande

Les boucles « for »

- 3.7) Écrire un programme C# qui demande un nombre de départ qui doit obligatoirement être pair (boucle « do... while ») sinon on redemande un nombre, et qui ensuite affiche les **dix** nombres suivants par saut de 2 (boucle « for »). Par exemple, si l'utilisateur entre le nombre 18, le programme affichera de 18 à 38.

```
Entrez un nombre pair :  
9  
18  
Résultat = [18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38]
```

- 3.8) Écrire un programme C# qui demande successivement 5 nombres entiers positifs à l'utilisateur, et qui lui dit ensuite quel était le plus grand parmi ces 5 nombres.

```
Entrer nombre : 120  
Entrer nombre : 150  
Entrer nombre : 7  
Entrer nombre : 90  
Entrer nombre : 32  
Le plus grand nombre est 150
```

Défis : Les boucle « for » avancées

- 3.9) Écrire un programme C# qui demande un nombre x et affiche les x premiers nombres de cette série [1, 1, 2, 3, 5, 8, 13, ...] (suite de Fibonacci).

```
Entrer le nombre d'éléments de la suite : 9  
Suite de Fibonacci = [1, 1, 2, 3, 5, 8, 13, 21, 34]
```

- 3.10) Écrire un programme C# qui demande un nombre à l'utilisateur (hauteur) et qui affiche un « carré (rectangle) » de cette hauteur avec des « X »

```
Entrer hauteur du carré : 5
XXXXX
XXXXX
XXXXX
XXXXX
XXXXX
```

- 3.11) Écrire un programme C# qui demande un nombre à l'utilisateur (hauteur) et qui affiche une pyramide de cette hauteur avec des « X »

```
Entrer hauteur de la pyramide : 5
  X
 XXX
XXXXX
XXXXXXXXX
XXXXXXXXXX
```