



Travaux Pratiques N° 1

Les bases de langage C#

EXERCICE 1 :

- ◆ Écrivez un programme qui lit une série de nombres entiers que l'utilisateur peut terminer en entrant 0. Le programme affiche ensuite le plus grand nombre positif et le plus petit nombre négatif parmi tous les nombres entrés par l'utilisateur.
- ◆ Par exemple :
 - Si on entre 1, 2, 3, 0, le maximum positif est 3 et il n'y a pas de minimum négatif.
 - Si on entre -1, -3, 0, alors il n'y a aucun maximum positif, et le minimum négatif est -3.

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 une fonction permettant de calculer la somme des chiffres individuels composants un nombre.
- ◆ Tester le programme dans une fonction main.

EXERCICE 4:

- ◆ On appelle nombres d'Armstrong les nombres entiers tels que la somme des cubes de leurs chiffres (en base 10) est égale au nombre lui-même.
Exemple : $1^3 + 5^3 + 3^3 = 153$.
- ◆ Écrire un programme qui affiche tous les nombres d'Armstrong inférieurs à 1000000.
NB : il faudra utiliser la fonction de l'exercice 2.

EXERCICE 5 :

Un mathématicien nommé Ulam a proposé de générer une séquence de nombres à partir de tout entier positif n ($n > 0$), comme suit:

- Si n est 1, stop.
- Si n est pair, le nombre suivant est $n / 2$.
- Si n est impair, le numéro suivant est $3 * n + 1$.
- Continuez ce processus jusqu'à atteindre 1.

Exemple :

- Si l'utilisateur entre le nombre entier 5, alors la séquence Ulam sera dans ce format:
16 -> 8 -> 4 -> 2 -> 1.



- Si l'utilisateur donne le nombre entier 4, alors la séquence Ulam sera dans ce format:
2 -> 1

Ecrire un programme permettant d'afficher la séquence Ulam d'un entier saisi

EXERCICE 6:

Ecrivez un programme qui prend en entrée une chaîne et un caractère puis supprime ce caractère dans cette chaîne.

Exemple :

- Si la chaîne est "HelloWorld" et le caractère est 'l', le résultat devrait être: HeoWord
- Si la chaîne est "McDonalds" et le caractère est 's', le résultat devrait être: McDonald

EXERCICE 7:

- ◆ Un tableau A de dimension N+1 contient N valeurs entières triées par ordre croissant; la (N+1)ième valeur est indéfinie.
- ◆ Ecrire un programme permettant d'insérer une valeur VAL donnée au clavier dans le tableau A de manière à obtenir un tableau de N+1 valeurs triées.

EXERCICE 8 :

Un numéro ISBN est légal s'il comporte 10 chiffres et $d_1 + 2 * d_2 + 3 * d_3 + \dots + 10 * d_{10}$ est un multiple de 11.

Par exemple le numéro ISBN 0-201-31452-5 est valide car : $1*5 + 2*2 + 3*5 + 4*4 + 5*1 + 6*3 + 7*1 + 8*0 + 9*2 + 10*0 = 88$ et 88 est un multiple de 11.

Ecrire un programme qui vérifie si un nombre est un ISBN.

Exemple :

- 1) Si l'utilisateur saisie 8535902775, alors le code doit afficher "Valide"
- 2) Si l'utilisateur saisie 1843369283, alors le code doit afficher "Non Valide"

EXERCICE 9 :

Ecrivez un programme qui prend en entrée une phrase et met en majuscule le premier caractère de chaque mot de cette phrase sans affecter les autres.

Exemples:

- Si la chaîne est "jon skeet", le programme doit afficher "Jon Skeet".
- Si la chaîne est "old mcdonald", le programme devrait afficher "Old Mcdonald".
- Si la chaîne est "miles o'Brien", le programme devrait afficher "Miles O'Brien".



EXERCICE 10 :

Ecrire un programme en C# permettant de calculer la somme de la série $1 + 11 + 111 + 1111 + \dots$ de n termes.

EXERCICE 11 :

Créez un tableau pouvant contenir dix nombres entiers et remplissez chaque emplacement avec une valeur aléatoire de 1 à 50. Affichez ces valeurs à l'écran, puis demandez à l'utilisateur un entier. Effectuez une recherche dans le tableau et comptez le nombre de fois que l'élément est trouvé.

Exemple d'exécution du programme :

```
Array: 1 2 3 4 5 6 7 7 9 14  
Value to find:12  
12 was found 0 times
```

```
Array: 1 2 3 4 5 6 7 7 9 14  
Value to find:7  
7 was found 2 times
```

EXERCICE 12 :

Ecrire un programme qui calcule la somme de la diagonale d'une matrice de dimensions M et N .