

Université des Lagunes
L2
Travaux Dirigés-Travaux Pratiques
C++

Exercice 1 : Soient les déclarations :

- char c = '\x05' ;
- int n = 5 ;
- long p = 1000 ;
- float x = 1.25 ;
- double z = 5.5 ;

Quels sont le type et la valeur de chacune des expressions suivantes :

- n + c + p
- 2 * x + c
- (char) n + c
- (float) z + n / 2

Exercice 2 : Soit le programme suivant :

```
#include <iostream>
using namespace std ;
main()
{
    int n=10, p=5, q=10, r ;

    r = n == (p = q) ;
    cout << "A : n = " << n << " p = " << p << " q = " << q
         << " r = " << r << "\n" ;

    n = p = q = 5 ;
    n += p += q ;
    cout << "B : n = " << n << " p = " << p << " q = " << q << "\n" ;

    q = n < p ? n++ : p++ ;
    cout << "C : n = " << n << " p = " << p << " q = " << q << "\n" ;

    q = n > p ? n++ : p++ ;
    cout << "D : n = " << n << " p = " << p << " q = " << q << "\n" ;
}
```

Pour chaque valeur des variables n, p et q, quels sont les résultats fournis ?

Exercice 3 : Écrire un programme qui calcule les racines carrées de nombres fournis en donnée. Il s'arrêtera lorsqu'on lui fournira la valeur 0. Il refusera les valeurs négatives. Son exécution se présentera ainsi :

```

donnez un nombre positif : 2
sa racine carrée est : 1.414214e+00
donnez un nombre positif : -1
svp positif
donnez un nombre positif : 5
sa racine carrée est : 2.236068e+00
donnez un nombre positif : 0

```

Exercice 4 : Calculer la somme des n premiers termes de la « série harmonique », c'est-à-dire

la somme : $1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$

NB : La valeur de n sera lue en donnée.

Exercice 5 : Écrire un programme qui détermine la nième valeur un (n étant fourni en donnée) de la « suite de Fibonacci » définie comme suit :

$$u_1 = 1$$

$$u_2 = 1$$

$$u_n = u_{n-1} + u_{n-2} \text{ pour } n > 2$$

Exercice 6 : Écrire :

- 1- une fonction, nommée f1, se contentant d'afficher « bonjour » (elle ne possédera aucun argument, ni valeur de retour) ;
- 2- une fonction, nommée f2, qui affiche « bonjour » un nombre de fois égal à la valeur reçue en argument (int) et qui ne renvoie aucune valeur ;
- 3- une fonction, nommée f3, qui fait la même chose que f2, mais qui, de plus, renvoie la valeur (int) 0.

Écrire un petit programme appelant successivement chacune de ces 3 fonctions, après les avoir convenablement déclarées.

Exercice 7 : Écrire 2 fonctions à un argument entier et une valeur de retour entière permettant de préciser si l'argument reçu est multiple de 2 (pour la première fonction) ou multiple de 3 (pour la seconde fonction).

Utiliser ces deux fonctions dans un petit programme qui lit un nombre entier et qui précise s'il est pair, multiple de 3 et/ou multiple de 6, comme dans cet exemple (il y a deux exécutions) :

```

donnez un entier : 9
il est multiple de 3

-----
donnez un entier : 12
il est pair
il est multiple de 3
il est divisible par 6

```

Exercice 8 : Écrire un programme qui lit 10 nombres entiers dans un tableau avant d'en rechercher le plus grand et le plus petit.

Exercice 9 : Écrire un programme permettant de calculer la somme des éléments d'une matrice $M_{3,4}$ dont les éléments sont entrés par l'utilisateur.

Exercice 10 : Réaliser une classe Etudiant permettant de manipuler un Etudiant inscrit à l'Université des Lagunes. On s'intéressera à l'étudiant qui suit un cours et a des notes dans la matière dans laquelle il a suivi le cours.

Écrire un programme déclarant un étudiant, affichant le cours qu'il a suivi et sa note pour ce cours.