Module/ElÃl'ment : POO-C++ Nom de l'enseignant : Youssef Baddi

TP 2

Exercise 1

Quelles erreurs ont été commises dans chacun des groupes d'instructions suivants :

1.
```
int a, b;
if (a<b) cout << "ascendant"
else cout << "non ascendant" ;
```

2.
```
int n ;
...
switch (2*n+1)
{ case 1 : cout << "petit" ;
case n : cout << "moyen" ;
}
```

3.
```
const int LIMITE=100
int n ;
...
switch (n)
{
  case LIMITE-1 : cout << "un peu moins" ;
  case LIMITE : cout << "juste" ;
  case LIMITE+1 : cout << "un peu plus" ;

}
```

Exercise 2

Soit le programme suivant :

```
#include <iostream>
main()
{   int n;
  cin >> n ;
  switch (n) {
  case 0 :
    cout << "Nul\n" ;
  case 1 :
  case 2 :
    cout << "Petit\n" ;
    break ;
  case 3 :
  case 4 :
  case 5 : cout << "Moyen\n" ;
  default : cout << "Grand\n" ;
  }
}
```

Quels résultats affiche-t-il lorsqu'on lui fournit en donnée :

— 0
— 1
— 4
— 10
— -5

Exercise 3

Soit le petit programme suivant :

```
#include <iostream> using namespace std ; main()
{ int i,n,som;
      som = 0 ;
      for (i=0 ; i<4 ; i++)
         { cout << "donnez un entier " ;
           cin >> n ;
           som += n ;
         }
      cout << "Somme : " << som ;
}
```

Écrire un programme réalisant exactement la même chose, en employant, à la place de l'instruction for :

— une instruction while,
— une instruction do ... while.

Exercise 4

Quels résultats fournit le programme suivant :

```
#include <iostream>
using namespace std ;
main()
  {   int n=0 ;
      do
        { if (n%2==0)
           {   cout << n << " est pair\n" ;
         n += 3 ;
         continue ;
       }
           if (n%3==0)
            {   cout << n << " est multiple de 3\n" ;
                  n += 5 ;
              }
           if (n%5==0)
            {   cout << n << " est multiple de 5\n" ;
              break ;
            }
     n += 1 ;
   }
   while (1) ;
 }
```

Exercise 5

Quels résultats fournit le programme suivant :

```
#include <iostream>
using namespace std ;
main()
{ int n,p;

    n=0 ;
    while (n<=5) n++ ;
      cout << "A : n = " << n << "\n" ;

    n=p=0 ;
    while (n<=8) n += p++ ;
      cout << "B : n = " << n << "\n" ;

    n=p=0 ;
    while (n<=8) n += ++p ;
      cout << "C : n = " << n << "\n" ;

  n=p=0 ;
  while (p<=5) n+= p++ ; cout<<"D:n="<<n<<"\n" ;

    n=p=0 ;
    while (p<=5) n+= ++p ;
      cout << "E : n = " << n << "\n" ;
}
```

Exercise 6

Quels résultats fournit le programme suivant :

```
#include <iostream>
using namespace std ;

main()
{ int n,p;
  n=p=0 ;
  while (n<5) n+=2 ;
  p++ ;
  cout<<"A:n="<<n<<"p="<<p<< "\n";

    n=p=0 ;
    while (n<5) { n+=2 ; p++ ; }
      cout << "B : n = " << n << " p = " << p << "\n" ;
}
```

Exercise 7

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

Rappelons que la fonction sqrt fournit la racine carrée (double) de la valeur (double) qu'on lui donne en argument.

Exercise 8

Calculer la somme des n premiers termes de la ń série harmonique ż, c'est-à-dire la somme :

$$1 + 1/2 + 1/3 + 1/4 + ... + 1/n$$

La valeur de n sera lue en donnée.

Exercise 9

Afficher un triangle isocèle formé d'étoiles. La hauteur du triangle (c'est-à-dire le nombre de lignes) sera fourni en donnée, comme dans l'exemple ci-dessous. On s'arrangera pour que la dernière ligne du triangle s'affiche sur le bord gauche de l'écran.

combien de lignes ? 10

Exercise 10

Afficher toutes les manières possibles d'obtenir un DH avec des pièces de 2 cents, 5 cents et 10 cents. Dire combien de possibilités ont été ainsi trouvées. Les résultats seront affichés comme suit :

- 1 DH = 50 X 2c
- 1DH=45X2c + 2X5c
- 1DH=40X2c + 4X5c

— 1DH=35X2c + 6X5c
— 1DH=30X2c + 8X5c
— 1DH=25X2c + 10X5c
— 1DH=20X2c + 12X5c
— 1DH=15X2c + 14X5c
— 1DH=10X2c + 16X5c
— 1DH= 5X2c + 18X5c
— 1 DH = 20 X 5c
— 1DH=45X2c + 1X10c
— 1DH=40X2c + 2X5c + 1X10c
— 1DH=35X2c + 4X5c + 1X10c
— 1DH=10X2c + 2X5c + 7X10c
— 1DH= 5X2c + 4X5c + 7X10c
— 1DH= 6X5c + 7X10c
— 1DH=10X2c + 8X10c
— 1DH= 5X2c + 2X5c + 8X10c
— 1DH= 4X5c + 8X10c
— 1DH= 5X2c + 9X10c
— 1DH= 2X5c + 9X10c
— 1 DH = 10 X 10c

En tout, il y a 66 façons de faire 1 DH

Rappelons que l'insertion dans le flot cout d'une expression de la forme setw(n), où n est une expression entière, demande de réaliser l'affichage suivant (et uniquement ce dernier) sur n caractères auminimum. L'emploi de setw nécessite l'inclusion du fichier iomanip.

Exercise 11

Écrire un programme qui détermine la nième valeur u_n (n étant fourni en donnée) de la ń suite de Fibonacci ż définie comme suit :

$$u_1 = 1$$

$$u_2 = 1$$

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

Exercise 12

Écrire un programme qui trouve la plus grande et la plus petite valeur d'une succession de notes (nombres entiers entre 0 et 20) fournies en données, ainsi que le nombre de fois où ce maximum et ce minimum ont été attribués. On supposera que les notes, en nombre non connu à l'avance, seront terminées par une valeur négative. On s'astreindra à ne pas utiliser de "tableau". L'exécution du programme pourra se présenter ainsi :
donnez une note (-1 pour finir) : 12
donnez une note (-1 pour finir) : 8
donnez une note (-1 pour finir) : 13
donnez une note (-1 pour finir) : 7
donnez une note (-1 pour finir) : 11
donnez une note (-1 pour finir) : 12
donnez une note (-1 pour finir) : 7
donnez une note (-1 pour finir) : 9
donnez une note (-1 pour finir) : -1
note maximale : 13 attribuée 1 fois

note minimale : 7 attribuée 2 fois

Exercise 13

Écrire un programme qui affiche la ń table de multiplication ż des nombres de 1 à 10, sous la forme suivante :

```
      I   1   2   3   4   5   6   7   8   9  10
---------------------------------------------
   1  I   1   2   3   4   5   6   7   8   9  10
   2  I   2   4   6   8  10  12  14  16  18  20
   3  I   3   6   9  12  15  18  21  24  27  30
   4  I   4   8  12  16  20  24  28  32  36  40
   5  I   5  10  15  20  25  30  35  40  45  50
   6  I   6  12  18  24  30  36  42  48  54  60
   7  I   7  14  21  28  35  42  49  56  63  70
   8  I   8  16  24  32  40  48  56  64  72  80
   9  I   9  18  27  36  45  54  63  72  81  90
  10  I  10  20  30  40  50  60  70  80  90 100
```

Rappelons que l'insertion dans le flot cout d'une expression de la forme setw(n), où n est une expression entière, demande de réaliser l'affichage suivant sur n caractères au minimum. L'emploi de setw nécessite l'inclusion du fichier iomanip.

Bon courage ! ! !