

TD 1 : Se familiariser avec le langage C++

Exercice 1

Soient les déclarations :

```
char c = '\x01' ;  
short int p = 10 ;
```

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

<code>p + 3</code>	<code>/* 1 */</code>
<code>c + 1</code>	<code>/* 2 */</code>
<code>p + c</code>	<code>/* 3 */</code>
<code>3 * p + 5 * c</code>	<code>/* 4 */</code>

Exercice 2

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 :

<code>n + c + p</code>	<code>/* 1 */</code>
<code>2 * x + c</code>	<code>/* 2 */</code>
<code>(char) n + c</code>	<code>/* 3 */</code>
<code>(float) z + n / 2</code>	<code>/* 4 */</code>

Exercice 3

Soient les déclarations suivantes :

```
int n = 5, p = 9 ;  
int q ;  
float x ;
```

Quelle est la valeur affectée aux différentes variables concernées par chacune des instructions suivantes ?

<code>q = n < p ;</code>	<code>/* 1 */</code>
<code>q = n == p ;</code>	<code>/* 2 */</code>
<code>q = p % n + p > n ;</code>	<code>/* 3 */</code>
<code>x = p / n ;</code>	<code>/* 4 */</code>
<code>x = (float) p / n ;</code>	<code>/* 5 */</code>
<code>x = (p + 0.5) / n ;</code>	<code>/* 6 */</code>
<code>x = (int) (p + 0.5) / n ;</code>	<code>/* 7 */</code>
<code>q = n * (p > n ? n : p) ;</code>	<code>/* 8 */</code>
<code>q = n * (p < n ? n : p) ;</code>	<code>/* 9 */</code>

Exercice 4

Quels résultats fournit le programme suivant :

```
#include <iostream>
using namespace std ;
main () {
    int i, j, n ;
    i = 0 ;
    n = i++ ;
    cout << "A : i = " << i << " n = " << n << "\n" ;

    i = 10 ;
    n = ++ i ;
    cout << "B : i = " << i << " n = " << n << "\n" ;

    i = 20 ;
    j = 5 ;
    n = i++ * ++ j ;
    cout << "C : i = " << i << " j = " << j << " n = " << n << "\n" ;

    i = 15 ;
    n = i += 3 ;
    cout << "D : i = " << i << " n = " << n << "\n" ; i = 3 ;

    j = 5 ;
    n = i *= --j ;
    cout << "E : i = " << i << " j = " << j << " n = " << n << "\n" ;
}
```

Exercice 5

Quels résultats fournira ce programme :

```
#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" ;
}
```

Exercice 6

Quels résultats fournira ce programme :

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

    /* cas 1 */
    n = 5 ; p = 2 ;
    q = n++ > p || p++ != 3 ;
    cout << "A : n = " << n << " p = " << p << " q = " << q << "\n" ;

    /* cas 2 */
    n = 5 ; p = 2 ;
    q = n++ < p || p++ != 3 ;
    cout << "B : n = " << n << " p = " << p << " q = " << q << "\n" ;

    /* cas 3 */
    n = 5 ; p = 2 ;
    q = ++n == 3 && ++p == 3 ;
    cout << "C : n = " << n << " p = " << p << " q = " << q << "\n" ;

    /* cas 4 */
    n = 5 ; p = 2 ;
    q = ++n == 6 && ++p == 3 ;
    cout << "D : n = " << n << " p = " << p << " q = " << q << "\n" ;
}
```

Exercice 7

Soit le programme suivant :

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

- a. 0
- b. 1
- c. 4
- d. 10
- e. -5

Exercice 8

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) ;
}
```

Combien de fois la boucle **do-while** s'exécute ?

Exercice 9

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

$$1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$$

La valeur de n sera saisie par l'utilisateur.

Exercice 10

Écrire un programme qui détermine la ⁿ^{ème} valeur **u_n** (n étant fourni en donnée) de la « **suite de Fibonacci** » définie comme suit :

$$\begin{aligned} u_1 &= 1 \\ u_2 &= 1 \\ u_n &= u_{n-1} + u_{n-2} \text{ pour } n > 2 \end{aligned}$$

Exercice 11

Afficher un triangle isocèle formé d'étoiles. La hauteur du triangle (c'est-à-dire le nombre de lignes) sera saisie par l'utilisateur, 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

```

      *
     ***
    *****
   ********
  *********
 *****
*****
*****
*****
*****
*****
*****
*****
*****

```

Exercice 12

Écrire un programme qui affiche la « **table de multiplication** » 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 de la bibliothèque **iomanip**.

Exercice 13 (Jeu de dés)

On souhaite écrire un algorithme pour simuler un jeu de lancer de dés à deux joueurs. Voici les règles :

- A chaque tour les deux joueurs peuvent choisir de lancer un ou deux dés. Le score est la somme cumulée des lancers de tous les tours de jeu.
- Avant de lancer le dé, le joueur choisit s'il veut lancer 1 ou 2 dés. S'il choisit de lancer deux dés, son score augmentera du résultat de la somme des deux dés. Cependant si les deux dés ont le même résultat son score sera diminué du résultat de la somme des dés.
- Si à l'issue d'un tour, un joueur a le score de 30 ou au-dessus, il remporte la partie.

Q1. Comment simuler un lancer de dés en informatique ?

Q2. Écrire une fonction **throwDice()** permettant de communiquer au reste du programme le résultat d'un lancer de dés.

Q3. Écrire une fonction **playerTurn()** qui correspond au tour d'un seul joueur. Cette fonction va renvoyer l'augmentation ou la diminution du score du joueur à l'issue de ce tour. Elle doit :

- Afficher le message "**Voulez-vous lancer 1 ou 2 dés ?**" et récupérer la réponse de l'utilisateur. Il faudra s'assurer que l'utilisateur donne une valeur cohérente avant de passer à la suite.
- En fonction du choix du joueur simuler le lancer de dés en utilisant la fonction **throwDice()**.
- En fonction du résultat obtenu (cf les règles du jeu), renvoyer au reste du programme l'évolution du score du joueur.

Q4. Dans le programme principal **main()**, écrire le programme permettant le déroulement du jeu :

- Définir et initialiser les variables **score1** et **score2**, respectivement les scores du joueur 1 et 2.
- Le jeu continue jusqu'à ce que le score d'un joueur atteigne 30 à la fin d'un tour. Lors d'un tour, le joueur 1 joue puis le joueur 2 et le score de chaque joueur est affiché. Écrire l'algorithme permettant le déroulement complet d'une partie.
- Lorsque la partie se termine, afficher le joueur gagnant. Il est possible d'obtenir un match nul si les deux joueurs ont un score dépassant 30 à la fin du dernier tour.