

## ÉCOLE SUPÉRIEURE POLYTECHNIQUE

## DÉPARTEMENT INFORMATIQUE

DIC1 && Licence GL  
Le Langage C  
Fiche de TD/TP N°2

**Exercice 1**

Qu'affiche chacune de ces expressions, si l'une d'elle n'affiche rien alors mettre « rien ».

Considérant que  $a = 15$ ,  $b = 4$  et  $c = 7$ .

- a) `printf( "%d", a % b );`
- b) `printf( "%d", a % c + b );`
- c) `printf( "b=" );`
- d) `printf( "a = 15" );`
- e) `printf( "%d = a + b", a + b );`
- f) `c = a + b;`
- g) `scanf( "%d%d", &a, &b );`
- h) `// printf( "Now a and b changes to %d and %d", a, b );`
- i) `printf( "\n" );`

**Exercice 2**

Déterminer l'ordre d'évaluation des opérations dans les expressions ci-dessous, et donner la valeur de x.

- a)  $x = 8 + 15 * (6 - 2) - 1$
- b)  $x = 5 \% 5 + 5 * 5 - 5 / 5$
- c)  $x = (5 * 7 * (5 + (7 * 5 / (7))))$

**Exercice 3**

Écrire un programme qui demande à l'utilisateur de saisir deux nombres puis affiche leur somme, leur produit, leur quotient, leur reste, ainsi que leur différence.

**Exercice 4**

Écrire un programme qui demande trois nombres entiers puis affiche leur somme, leur moyenne, leur produit, le plus petit et le plus grand de ces trois nombres.

**Exercice 5**

Écrire un programme qui demande un nombre entier composé de 5 chiffres puis affiche chacun de ces 5 chiffres séparé par une virgule.

**Exercice 6**

Écrire le programme qui donne le résultat ci-dessous :

number	square	cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

### Exercice 7

Ecrire un programme qui demande 10 nombres et détermine le plus grand, le deuxième plus grand, le plus petit, et le deuxième plus petit

### Exercice 8

Que produit le code ci-après (sans l'exécuter dans un premier temps, puis l'exécuter et comparer) :

```
1  #include <stdio.h>
2  int main( void )
3
4  {
5      int x = 1;
6      int y = 0;
7      while ( x <= 5 ) { // loop 5 times
8          total+= x * x; // perform calculation
9          printf( "%d\n", x*x);
10         ++x; // increment counter x
11     } // end while
12     printf( "Total is %d\n", total); // display result
13 }
```

### Exercice 9

Que produit le code ci-après (sans l'exécuter dans un premier temps, puis l'exécuter et comparer) :

```
1  #include <stdio.h>
2
3  int main( void )
4  {
5      int outer_count = 1; // initialize count
6      while ( outer_count <= 10 ) { // loop 10 times
7          int inner_count = 1;
8          while ( inner_count <= outer_count ) {
9              printf( "*" );
10             inner_count++;
11         } // end inner while
12         printf( "\n" );
13         outer_count++;
14     } // end outer while
15 }
```

### Exercice 10

Déterminer la sortie de chacun de ces bouts de code en considérant dans un premier temps que  $x = 9$  et  $y = 11$ , puis que  $x = 11$  et  $y = 9$ .

```
a) if ( x < 10 )
if ( y > 10 )
puts( "*****" );
else
puts( "#####" );
puts( "$$$$" );
b) if ( x < 10 ) {
if ( y > 10 )
puts( "*****" );
}
else {
puts( "#####" );
puts( "$$$$" );
}
```

### Exercice 11

Modifier le code ci-dessous pour produire la sortie proposée. Vous pourrez l'indenter et ne devez rien changer à

part ajouter des { et }.

```
if (y == 8)
if (x == 5)
puts( "@@@@@" );
else
puts( "#####" );
puts( "$$$$" );
puts( "&&&&&" );
```

a) Considérant que  $x = 5$  et  $y = 8$ , cette sortie est produite.

```
@@@@@
$$$$$
&&&&&
```

b) Considérant que  $x = 5$  et  $y = 8$ , cette sortie est produite.

```
@@@@@
```

c) Considérant que  $x = 5$  et  $y = 8$ , cette sortie est produite.

```
@@@@@
&&&&&
```

d) Considérant que  $x = 5$  et  $y = 7$ , cette sortie est produite.

```
#####
$$$$$
&&&&&
```

#### Exercice 12

Un nombre est un nombre d'Armstrong s'il est égal à la somme des cubes des chiffres qui le composent.

Par exemple, 153 est un nombre d'Armstrong  $153 = 1^3 + 5^3 + 3^3$ .

#### Exercice 13

Ecrire un programme qui affiche de manière ininterrompue les puissances du nombre 3, c-à-d 3, 9, 27, 92, 273, etc. Que se passe-t-il lorsque le programme s'exécute ?

#### Exercice 14

- 1) Ecrire un algorithme cryptographique qui chiffre chaque donnée (un nombre entier à 4 chiffres) comme suit :
  - a) Ajouter 7 à chaque chiffre, puis le diviser par 10, et récupérer le reste de la division et le remplacer par le chiffre initial.
  - b) Ensuite échanger le 1<sup>e</sup> chiffre par le 3<sup>e</sup>, et le 2<sup>e</sup> par le 4<sup>e</sup>.
- 2) Ecrire un algorithme qui effectue l'opération inverse, c-à-d déchiffrer un nombre chiffré avec la méthode de la question 1).