

TP 3

Exercise 1

Quelle modification faut-il apporter au programme suivant pour qu'il devienne correct :

```
1 #include <iostream>
2 using namespace std ;
3 int fct (int r);
4 main()
5 {
6     int fct (int r);
7     int n,p=5;
8     n = fct (p) ;
9     cout << "p = " << p << " n = " << n ;
10 }
11 int fct (int r)
12 {
13     return 2*r ;
14 }
```

Exercise 2

Écrire :

- une fonction, nommée f1, se contentant d'afficher " bonjour " (elle ne possédera aucun argument, ni valeur de retour);
- 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;
- 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 (on ne fera aucune hypothèse sur les emplacements relatifs des différentes fonctions composant le fichier source).

Exercise 3

Écrire une fonction qui reçoit en arguments 2 nombres flottants et un caractère, et qui fournit un résultat correspondant à l'une des 4 opérations appliquées à ses deux premiers arguments, en fonction de la valeur du dernier, à savoir : addition pour le caractère +, soustraction pour -, multiplication pour * et division pour / (tout autre caractère que l'un des 4 cités sera interprété comme une addition). On ne tiendra pas compte des risques de division par zéro.

Écrire un petit programme (main) utilisant cette fonction pour effectuer les 4 opérations sur les 2 nombres fournis en donnée.

Exercise 4

Transformer le programme (fonction + main) écrit dans l'exercice précédent de manière que la fonction ne dispose plus que de 2 arguments, le caractère indiquant la nature de l'opération à effectuer étant précisé, cette fois, à l'aide d'une variable globale.

Exercise 5

Quels résultats fournira ce programme :

```
1  #include <iostream>
2  using namespace std ;
3  int n=10, q=2 ;
4  main()
5  {
6      int fct (int) ;
7      void f (void) ;
8      int n=0, p=5 ;
9      n = fct(p) ;
10     cout << "A : dans main, n = " << n << " p = " << p
11         << " q = " << q << "\n" ;
12     f() ;
13 }
14 int fct (int p)
15 {
16     int q ;
17     q=2*p+n;
18     cout<<"B:dansfct, n="<<n<<"p="<<p
19         << " q = " << q << "\n" ;
20     return q ;
21 }
22 void f (void)
23 {
24     int p = q * n ;
25     cout<<"C:dansf, n="<<n<<"p="<<p
26     << " q = " << q << "\n" ;
27 }
```

Exercise 6

Écrire une fonction, sans argument ni valeur de retour, qui se contente d'afficher, à chaque appel, le nombre total de fois où elle a été appelée sous la forme :
appel numéro 3

Exercise 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

Exercise 8

Écrire une fonction permettant d'ajouter une valeur fournie en argument à une variable fournie également en argument. Par exemple, l'appel (n et p étant entiers) :

ajouter (2*p+1, n);

ajoutera la valeur de l'expression 2*p+1 à la variable n. Écrire un petit programme de test de la fonction.

Exercise 9

1. Transformer le programme suivant pour que la fonction fct devienne une fonction en ligne.

```
1  #include <iostream>
2  using namespace std ;
3  main()
4  {   int fct (char, int) ;
5  // déclaration (prototype) de fct
6  int n = 150, p ;
7  char c = 's' ;
8  p = fct ( c , n) ;
9  cout << "fct ('" << c << "', " << n << ") vaut : " << p ;
10 }
11 int fct (char c, int n)
12 {   int res ;
13     if (c == 'a')      res = n + c ;
14     else if (c == 's') res = n - c ;
15     else               res = n * c ;
16     return res ;
17 // définition de fct
18 }
```

2. Comment faudrait-il procéder si l'on souhaitait que la fonction fct soit compilée séparément ?

Exercise 10

Soient les déclarations suivantes :

```
1  int fct (int) ;           // fonction I
2  int fct (float) ;        // fonction II
3  void fct (int, float) ;  // fonction III
4  void fct (float, int) ;  // fonction IV
5  int n, p ;
6  float x, y ;
7  char c ;
8  double z ;
```

Les appels suivants sont-ils corrects et, si oui, quelles seront les fonctions effectivement appelées et les conversions éventuellement mises en place ?

- 1 a. fct (n) ;
- 2 b. fct (x) ;
- 3 c. fct (n, x) ;
- 4 d. fct (x, n) ;
- 5 e. fct (c) ;
- 6 f. fct (n, p) ;
- 7 g. fct (n, c) ;
- 8 h. fct (n, z) ;
- 9 i. fct (z, z) ;

Bon courage !!!