GIF-1003
PROGRAMMATION AVANCÉE EN C++

Laboratoire #2

Codage des algorithmes : éléments de syntaxe et sémantique en langage C++

Exercice #1. Traduction du pseudo-code en langage C++

Vous devez traduire fidèlement l'algorithme suivant en langage C++. Tracer systématiquement l'exécution de votre programme avec le débogueur tel que montré dans le laboratoire#1. Vous avez à compléter le fichier cube.cpp présent dans l'archive qui contient cet énoncé. Vous n'avez pas à traduire les assertions ({A : …}), elles devront seulement apparaitre sous forme de commentaires. Servez-vous du document traduction.pdf présent dans l'archive zip, vous y trouverez l'analogue en C++ de toutes les instructions et structures de contrôle en notation algorithmique (pseudo-code).

Algorithme#1. Développement d'un cube en une série de nombres impairs consécutifs.

```
Début
        Demander n
        {A: n est un nombre entier > 0}
        cube ← n*n*n
        s ← 0
        p ← 1
        k ← 1

        Tant que (s ≠ cube)
        début
                si (s>cube) alors
                début
                        s ← s - (2*k-1)
                        k ← k + 1
                fin
                si ( s<cube) alors
                début
                        s ← s + (2*p -1)
                        p ← p + 1
                fin
        fin

        premier ← 2*k -1 {le premier terme de la série}
        dernier ← 2* (p-1) -1 {le dernier terme}
        Afficher cube " = "
        k ← premier

        Tant que (k < dernier)
        début
                Afficher k " + "
                k ← k+2
        fin

        Afficher dernier
{A: affichage du développement du cube de n en une série de nombres impairs consécutifs}
fin
```

Exercice #2. Types de base, opérateurs et expressions, instructions de contrôle

Écrire un programme qui affiche un triangle isocèle formé d'étoiles. La hauteur du triangle (c'est-à-dire le nombre de lignes) sera fournie par l'usager. Le résultat attendu doit être comme l'exemple ci-dessous, soit que la dernière ligne s'affiche sur le bord gauche de l'écran.

```
combien de lignes ? 5
    *
   ***
  *****
 *******
*********
```

Exercice #3. cin : >> vs getline

Créer un nouveau projet C++, implantez, compilez et exécutez le programme suivant :

```
#include <iostream>
#include <string>
using namespace std;
int main (void){
        string phrase;
        cout << "Entrez une phrase comprenant des espaces" << endl;
        cin >> phrase;
        cout << phrase << endl ;
        return 0;
}
```

Que remarquez-vous ?

Utilisez getline à la place de >> en considérant maintenant le code suivant :

```
1   #include <iostream>
2   #include <string>
3   using namespace std;
4   int main (void)
5   {
6      int nombre ;
7      string phrase;
8      cout << "Entrez un nombre" << endl;
9      cin >> nombre ;
10     cout << "Entrez une phrase comprenant des espaces" << endl;
11     char buffer[256];
12     cin.getline(buffer,256);
13     cout << phrase << endl ;
14     return 0;
15  }
```

Compilez et exécutez ce programme. Que remarquez-vous cette fois-ci?
Insérez un appel à cin.ignore() entre les ligne 9 et 10, puis recompilez et exécuter de nouveau. Qu'en concluez-vous quant à l'usage de cin.ignore()?

Exercice #4. Instructions de contrôle

Une compagnie d'électricité veut faire une liste dont chaque ligne contient le numéro de client suivi du nombre de kilowattheures (kwh) utilisé par ce client et une approximation du montant de la facture de ce client.
Un opérateur fournira sur demande au programme les entiers correspondant au numéro de client suivi du nombre de kilowattheures utilisé par ce client. L'opérateur terminera par une sentinelle composée du numéro de client 0 et un entier quelconque pour le nombre de kilowattheures.
Le programme devra indiquer, après avoir obtenu la sentinelle, le nombre de clients traités ainsi qu'une approximation du montant total facturé pour l'ensemble des clients traités.

Remarques :

Il y moins de 32767 clients.
Tous les clients ont des numéros dans [0..32767].
Chaque consommation d'électricité traitée est dans [0..32767].
Le montant total facturé est facilement représentable en mémoire dans des cellules de type double.
La précision de l'ordinateur est amplement suffisante pour la compagnie.
Le tarif est le suivant :

25 $ d'abonnement, quelque soit la consommation.
50 cents par kWh pour les 100 premiers kWh;

35 cents par kWh pour les 150 kWh suivants;
20 cents par kWh pour la fraction de la consommation qui excède 250 kWh.

Exercice#5. Trouvez l'erreur dans chacun des segments de programmes suivants et expliquez comment la corriger.

a)

```
int g (void)
{
        cout << "g de la fonction intérieure" << endl;
        return 0;

        int h (void)
        {
                cout << "h de la fonction intérieure" << endl;
                return 0;
        }
}
```

b)

```
int somme (int x, int y)
{
        int resultat;
        resultat = x + y;
}
```

c)

```
int somme (int n)
{
        if (n ==0)
                return 0;
        else
                 n + somme (n - 1);
}
```

d)

```
void f (double a);
{
        float a;
        cout << a << endl;
}
```

e)

```
void produit (void)
{
        int a, b, c, resultat;
        cout << "Entrez trois entiers: ";
        cin >> a >> b >> c;
        resultat = a * b * c;
        cout << "Le résultat est " << resultat;
        return resultat;
}
```

f)

```
int f();
int main()
{
        std::cout << f();
        return 0;
}
```