

## Séance 4 : Jeudi 07/02/2019

## Codage des exercices TD2 / TP1

1. Écrire l'algorithme d'un programme permettant de simuler le fonctionnement d'une calculatrice simple (+, -, \*, /). Dans cet exercice, l'utilisateur saisira les deux opérandes, l'opérateur et le programme lui affichera le résultat correspondant. Dans le cas d'une division, on vérifiera bien que le dénominateur est non nul !

```

Début

Nb1, Nb2 : reels
Op : caractère
Afficher('Donnez le premier nombre :')
Saisir(Nb1)
Afficher('Donnez le deuxième nombre :')
Saisir(Nb2)
Afficher('Donnez le symbole de l'opération :')
Saisir(Op)
Selon Op
  '+' : Afficher (Nb1 + Nb2 )
  '-' : Afficher (Nb1 - Nb2 )
  '*' : Afficher (Nb1 * Nb2)
  '/' : Si (Nb2 = 0) Alors Afficher ('Opération impossible')
                                             Sinon

Afficher( Nb1 / Nb2)
      Fin si
Autrement : Afficher ('Opération inexistante')
Fin Selon
Fin

```

## Traduction C/C++

```

#include<iostream>
using namespace std;

int main (void)
{
    float nb1, nb2, res;
    char op ;
    cout<<"Donnez deux reels"<<endl;
    cin>>nb1;
    cin>>nb2; // cin>>nb1>>nb2;
    cout<<"Donnez l'operateur"<<endl;
    cin>>op ;
    switch (op)
    {
        case '+' : res = nb1+nb2 ; cout<<"La somme est :
"<<res<<endl;
            break;
        case '-' : cout<<"la difference est "<<nb1-nb2<<endl;
            break;
        case '*' : cout<<"le produit est "<<nb1*nb2<<endl;
            break;
    }
}

```

```

        case '/' : if (nb2 != 0) cout<<"la division est
"<<nb1/nb2<<endl;
                    else cout<<"Division par 0 impossible "<<endl;
        break;
        default : cout<<"l'operateur propose n'existe pas"<<endl;
    }
    return 0;
}

```

## 2. Premiers pas, premières étoiles

1. Écrivez un programme permettant d'afficher N étoiles sur une ligne ; N étant demandé à l'utilisateur.

Combien d'étoiles voulez-vous afficher ?	Résultat :
5	*****

```

int main (void)
{
    int n, i;
    cout<<"donnez le nombre d'etoiles"<<endl;
    cin>>n;
    for (i=1;i<=n;i=i+1)
    {
        cout<<"*";
    }
    return 0;
}

```

2. Écrivez un programme permettant d'afficher un rectangle de N \* M étoiles ; N et M étant demandés à l'utilisateur.

Quelles sont les dimensions du rectangle : longueur ?	Résultat :
4	****
Largeur	****
3	****

```

int main (void)
{
    int n, i, m, j;
    cout<<"donnez le nombre d'etoiles sur une ligne"<<endl;
    cin>>n;
    cout<<"donnez le nombre lignes"<<endl;
    cin>>m;
    for (j=1;j<=m;j=j+1)
    {
        for (i=1;i<=n;i=i+1)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    return 0;
}

```

3. Écrivez un programme permettant d'afficher un triangle de hauteur N ; N étant demandé à l'utilisateur et représentant à la fois le nombre maximum d'étoiles sur la dernière ligne et le nombre de lignes.

Quelle est la hauteur du triangle ?	Résultat :
4	* ** *** ****

```
int main (void)
{
    int n, i, m, j;
    cout<<"donnez la hauteur du triangle"<<endl;
    cin>>m;
    for (j=1;j<=m;j=j+1)
    {
        for (i=1;i<=j;i=i+1)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    return 0;
}
```

4. et maintenant le même triangle à l'envers !!!

Quelle est la hauteur du triangle ?	Résultat :
4	**** *** ** *

```
int main (void)
{
    int n, i, m, j;
    cout<<"donnez la hauteur du triangle"<<endl;
    cin>>m;
    for (j=1;j<=m;j=j+1)
    {
        for (i=1;i<=m+1-j;i=i+1)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    return 0;
}
```

3. Écrire un algorithme permettant de calculer la somme des n premières puissances de 2.  
Exemple : valeur saisie : 6 → résultat : 63 (= 1 + 2 + 4 + 8 + 16 + 32).  
Outil :  $2^i$  s'écrira en algorithmique : puissance (2, i)

```

Début
Variables : n, somme, i : entier
    Afficher('donnez la valeur de n')
    Saisir(n)
somme ← 0
    Pour i allant de 0 à n-1 par pas de 1 faire
        somme ← somme + puissance(2,i)
    Fin Pour
Afficher (somme)
Fin

```

#### Traduction en C/C++

```

int main (void)
{
    int i,n,res;
    cout<<"Donnez un entier"<<endl;
    cin>>n;
    res=0;
    for (i=0;i<n;i=i+1)
    {
        res = res + pow(2,i);
    }
    cout<<"le resultat est : "<<res<<endl;
    return 0;
}

```

- 4- Écrire un algorithme permettant de calculer la somme des n premiers nombres impairs. Exemple : valeur saisie : 6 → résultat : 36 (= 1 + 3 + 5 + 7 + 9 + 11)  
 Quel lien pouvez-vous établir entre la valeur obtenue et le nombre n ?

```

Début
Variables : n, somme, i : entier
    Afficher('donnez la valeur de n')
    Saisir(n)
somme ← 0
    Pour i allant de 1 à 2*n par pas de 2 faire
        somme ← somme + i
    Fin Pour
Afficher (somme)
Fin

```

Exemple pour n=5  
 Somme = 1+3+5+7+9 = 25 = 5<sup>2</sup>  
 Donc la somme des n premiers nombres impairs est égale au carré de n.

#### Traduction en C/C++

```

int main (void)
{
    int i,n,res;
    cout<<"Donnez un entier"<<endl;
    cin>>n;
    res=0;
    for (i=1;i<=n;i=i+1)// for (i=1;i<2*n;i++)
    {

```

```

        res = res + (2*i-1); //res += i
    }
    cout<<"le resultat est : "<<res<<endl;
    return 0;
}

```

- 5- Écrire un algorithme permettant de lire 20 nombres entiers au clavier. Si le nombre x saisi est pair, on affiche la valeur  $(x / 2)$  sinon on affiche  $(3*x + 1)$ . Attention, on ne mémoriser pas les 20 valeurs saisies.

Puisque l'on connaît le nombre de passages dans la boucle, on utilise la boucle **pour** :

Début

Variables : nbre, i : entier

Pour i allant de 1 à 20 par pas de 1 faire

    Afficher ('Entrez un nombre')

    Saisir(nbre)

    Si (nbre modulo 2) = 0                   Alors                   nbre ← nbre / 2  
   Sinon                    nbre ← 3\*nbre + 1

        FinSi

        Afficher(nbre)

FinPour

Fin

### Traduction en C/C++

```

#include<iostream>
using namespace std;
int main (void)
{
    int i,n;
    for (i=1 ;i<=20 ;i++)
    {
        cout<<"Donnez un entier"<<endl;
        cin>>n;
        if (n%2 ==0) cout<<n/2 ;
        else cout<<(3*n) + 1 ;
    }
    return 0;
}

```

## Cours 3 : fonctions et procédures

=> télécharger le fichier 2019\_SUR\_CM3.pdf

## Exemple d'application du cours

Reprendre les instructions de dessin d'un triangle et en faire une procédure. Utiliser la procédure dans le main pour visualiser le résultat.

```
void dessin_triangle (int h)
{
    int i,j;
    for (i=1;i<=h; i++)
    {
        for (j=1;j<=i; j++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
}

int main (void)
{
    int h;
    cout<<"donnez un entier"<<endl;
    cin>>h;
    dessin_triangle(h);

    return 0;
}
```