

Séance N°2

Boucles et structures de contrôle

Pour réaliser un traitement **sous condition**, il existe l'instruction **if ... else**. Pour réaliser plusieurs fois le même traitement (faire des **traitements itératifs**), il y a 3 types de boucles en C++: **for**, **while** et **do ... while**.

Traitement conditionnel : **if ... else**

Syntaxe :

```
if(condition)
{
// instructions à effectuer si la condition est vraie,
// séparées par des « ; »
}
else
{
// instructions à effectuer si la condition est fausse,
// séparées par des « ; »
}
```

Opérateurs conditionnels:

Plus petit que	<
Plus grand que	>
Plus petit ou égal à	<=
Plus grand ou égal à	>=
Égal à	==
Différent de (inégal)	!=
ET logique	&&
OU logique	

Ex 1 - Testez le programme :

```
void main()
{
int a;
cout<<"donner a : ";
cin>>a;
int b;
cout<<"donner b : ";
cin>>b;
if(a>b) {
cout<<"a est plus grand que b..."<<endl;
}
else {
cout<<"a est inferieur ou egal a b..."<<endl;
}
}
```

Boucle itérative : **for**

Syntaxe :

```
for (initialisation de la variable de boucle ; condition de fin de boucle ; incrémentation)
{
// instructions à répéter, séparées par des « ; »
}
```

Ex 2 - Testez le programme :

```
//boucle for
#include <iostream>
using namespace std;
void main()
{
for(int i=0; i<6; i++)
    {
        int carre = i*i;
        cout<<carre<<endl;
    }
}
```

Question : combien de passages dans la boucle ou **nombre d'itérations** sont réalisées ?

Question : à quoi correspond i++ ? que se passe-t-il si on remplace i++ par i+2 ?

Boucle itérative : while

Permet de répéter un groupe d'instructions tant qu'une condition est vérifiée (vraie).

Syntaxe :

```
while(condition)
{
// instructions à répéter, séparées par des « ; »
}
```

Ex 3 - Testez le programme :

```
#include <iostream>
using namespace std;
void main()
{
int a = 4;
int b = 7;
while(a<=b)
    {
        cout<<a<<endl;
        a++;
    }
}
```

Question : que se passe-t-il si on place en commentaire l'instruction //a++ ?

Question : réactiver l'instruction a++ ; et inverser les valeurs : a=7 et b=4, que se passe-t-il ?

Boucle itérative : do ... while

Permet d'exécuter au moins une fois un groupe d'instructions, puis de répéter tant qu'une condition est vraie.

Syntaxe :

```
do
{
// Instructions à répéter, séparées par des « ; »
}
while(condition);
```

Ex 4 - Testez le programme :

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

```

void main()
{
int a = 7;
int b = 4;
do
    {
    cout<<a<<endl;
    a--;
    }
while(a>b);
}

```

TRAVAIL PERSONNEL

Pour mettre en application ce que j'ai appris, je fais les exercices suivants :

Exercice 1

Écrire un programme pour calculer les **racines** d'une équation du second degré : $ax^2 + bx + c = 0$.

Le programme devra

- demander à l'utilisateur de saisir les 3 **coefficients** (a, b, c) du polynôme,
- calculer le **discriminant**, et
- suivant la valeur du discriminant, afficher la valeur des deux racines réelles, de la racine double, ou des deux racines complexes sous la forme $a + i b$.

La fonction *sqrt* permet de calculer la racine carrée (square root) ; elle nécessite la directive `#include <cmath>` en tête de programme.

Exercice 2

Écrire un programme pour calculer la **somme** et la **moyenne** de N valeurs.

Le programme devra préalablement demander à l'utilisateur : 'Combien de valeurs sont à saisir ?' pour initialiser la variable N.

Exercice 3

Modifier le programme précédent en supprimant la variable N : calculer la somme et moyenne des valeurs saisies **tant que** la valeur saisie est différente de -1.

BILAN PERSONNEL

Ce que j'ai appris aujourd'hui : (à compléter)

Vocabulaire informatique : (à compléter)