

Introduction à la programmation

Cours no. 2

Jean-Sébastien Coron

Université du Luxembourg

Exécution conditionnelle

- ▶ Un test permet d'exécuter une ou plusieurs instructions en fonction de l'état de certaines variables.

```
if (i>5)
    printf("La variable i est supérieure à 5");
```

- ▶ Utilisation de `else`.

```
if (i>=0)
    printf("La variable i est positive.");
else printf("La variable i est négative");
```

Exécution conditionnelle

- ▶ Exécution de plusieurs instructions:

```
if (x>0)
{
  y=2*x;
  z=3*x;
}
else
{
  y=x-1;
  z=x-2;
}
```

- ▶ L'instruction `else {...}` est optionnelle.

Opérateurs de test

- ▶ Tests possibles:
 - ▶ Égalité: $a==b$
 - ▶ Différence: $a!=b$
 - ▶ Comparaison stricte: $a<b$
 - ▶ Comparaison large: $a<=b$
- ▶ Résultat d'un test.
 - ▶ Le résultat du test est un entier nul si le test est faux, non nul si le test est vrai.

Opérations sur les tests

- ▶ Négation:
 - ▶ `!(test).`
- ▶ Conjonction (ET)
 - ▶ `((test1) && (test2))`
- ▶ Disjonction (OU)
 - ▶ `((test1) || (test2))`

Example

```
int maximum(int a,int b)
{
    int m;
    if (a>b) m=a;
    else m=b;
    return m;
}
```

Exemple

- ▶ On souhaite vérifier que deux quantités sont positives, et afficher un message si ce n'est pas le cas.

```
float x,y;  
...  
...  
if((x<0) || (y<0))  
{  
    printf("Erreur: la variable x<0 ou y<0");  
}
```

Exemple

- ▶ On souhaite déterminer si une variable x appartient à un certain intervalle $[a, b]$.

```
int intervalle(float x,float a,float b)
{
    if ((x>=a) && (x<=b)) return 0;
    return 1;
}
```


Boucles conditionnelles

- ▶ Il est possible de répéter plusieurs fois le même bloc d'instruction, en fonction du résultat d'un test.
- ▶ Instruction `while (test) instruction`
 - ▶ L'instruction s'exécute tant que le résultat du test est vrai.
 - ▶ L'instruction n'est pas exécutée si le résultat du test est faux la première fois.
- ▶ Instruction `do (instruction) while (test)`
 - ▶ L'instruction est exécutée une fois, puis répétée si le résultat du test est vrai.
 - ▶ L'instruction est exécutée au moins une fois.

Instruction while

- ▶ Exemple: affiche les entiers de 1 à 10.

```
#include <stdio.h>
int main()
{
    int i=1;
    while(i<=10)
    {
        printf("%d\n",i);
        i=i+1;
    }
}
```

Instruction do while

- ▶ Exemple: on demande à l'utilisateur d'entrer un entier qui doit être positif, sinon on repose la question.

```
#include <stdio.h>
int main()
{
    int a;
    do
    {
        printf("Entrez un nombre positif.\n");
        scanf("%d",&a);
    } while (a<0);
}
```