

# Introduction à la programmation

## Cours no. 3

Jean-Sébastien Coron

Université du Luxembourg

## Rappel: les boucles conditionnelles

- ▶ 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.

# Boucle for

- ▶ Une boucle for permet de répéter une instruction plusieurs fois, à l'aide d'une variable de contrôle.
  - ▶ `for(init;test;itération) opération;`
  - ▶ `init`: initialiser la variable de contrôle.
  - ▶ `test`: test de la variable de contrôle
  - ▶ `itération`: opération sur la variable de contrôle
  - ▶ `opération`: opération de la boucle
- ▶ Exemple: afficher les entiers de 1 à 10.
  - ▶ 

```
for (i=1;i<=10;i=i+1)
    printf("%d\n",i);
```

# Boucle for

- ▶ Syntaxe :
  - ▶ `for(init;test;itération) opération;`
- ▶ Séquence d'opération :
  - ▶ `init`
    - `test:si test=faux, saut à fin du for`
    - `opération`
    - `itération`
    - `retour à test`
    - `fin du for`
- ▶ Exemple: somme des entiers de 1 à 10 :
  - ▶ `s=0;`  
`for (i=1;i<=10;i=i+1) s=s+i;`

# Variable de contrôle

- ▶ On peut aussi décrémenter la variable :
  - ▶ Boucle affichant les nombres de 10 à 0 :

```
for (i=10;i>=0;i=i-1)
    printf("%d\n",i);
```
- ▶ Explication :
  - ▶ `i=10`: valeur initiale de la variable `i`
  - ▶ `i>=0`: la boucle continue tant que `i>=0`
  - ▶ `i=i-1`: après chaque exécution de l'opération de la boucle, la variable `i` est décrémentée.

# Boucle For

- ▶ Exemple :

- ▶ `for (i=0;i<6;i=i+1)`  
    `printf("%d\n",i);`

- ▶ La boucle affiche 0,1,2,3,4,5.

- ▶ Elle commence à  $i=0$ , vérifie que  $i<6$ , et affiche 0 à l'écran.

- ▶ La boucle continue avec  $i=1,2,3,4,5$ .

- ▶ Lorsque  $i=6$ , la condition  $i<6$  n'est plus réalisée, et on sort de la boucle.

- ▶ La valeur 6 n'est pas affichée.

- ▶ La boucle s'est exécutée en tout 6 fois, pour  $i=0,1,2,3,4,5$

## Exemple

- ▶ On veut calculer  $a \cdot b$  en calculant

$$a \cdot b = b + \dots + b$$

- ▶ Avec une boucle :

```
int mult(int a,int b)
{
    int i,c=0;
    for(i=0;i<a;i++)
    {
        c=c+b;
    }
    return c;
}
```

# Attention

- ▶ Toujours vérifier que la boucle se termine :
  - ▶ Incorrect : `for (i=0;i<6;i=i-1)`
- ▶ Si nécessaire, utiliser `printf` dans la boucle pour vérifier la valeur du compteur.
- ▶ Bien compter le nombre d'exécutions de la boucle :
  - ▶ `for(i=0;i<10;i++)` exécute 10 fois la boucle (i de 0 à 9)
  - ▶ `for(i=0;i<=10;i++)` exécute 11 fois la boucle (i de 0 à 10)
  - ▶ `for(i=1;i<10;i++)` exécute 9 fois la boucle (i de 1 à 9)



## Utilisation de for

- ▶ On utilise généralement une boucle `for` lorsqu'on souhaite exécuter une opération un nombre de fois connu à l'avance.
- ▶ Exemple: l'instruction va s'exécuter 10 fois.

```
int n=10;
int i;
for(i=0;i<n;i++)
{
    <instruction(i)>
}
```

## for = while

- ▶ Les instructions for et while sont en faites équivalentes.

```
for (init;test;itération)
    opération
```

```
init
while (test)
{
    opération
    itération
}
```