

# Programmation Langage C

## Les boucles

Youssef ALJ

9 mars 2020

## 1 Les boucles

- Boucle for
- Boucle do while
- Boucle while
- Break et Continue

La boucle pour est constituée :

- d'initialisation exécuté **avant toutes les itérations**.
- de condition de boucle exécuté **avant chaque itération**.
- d'instruction de fin de boucle (souvent une incrémentation ou une décrémentation) exécuté **après chaque itération**.

```
for (initialisation ; condition ; incrementation)
{
    instructions_a_repeter ;
}
```

# Exemple 1

```
#include <stdio.h>
int main()
{
    int i ;
    for (i = 0 ; i < 10 ; i = i + 1)
    {
        printf ("iteration %d \n", i) ;
    }
    printf ("valeur de i apres la boucle : %d \n", i) ;
    return 0 ;
}
```

On veut afficher les nombres pairs de 0 jusqu'à 10.

On veut afficher les nombres pairs de 0 jusqu'à 10.

```
#include <stdio.h>

int main()
{
    for (int i = 0; i <= 10; i++)
    {
        if (i%2 == 0)
        {
            printf("i=%d\n", i);
        }
    }
    return 0;
}
```

On peut initialiser la variable  $i$  à l'intérieur ou à l'extérieur de la boucle

```
for (int i = 0; i <= 10; i++)
{
    // instructions qu'on veut répéter
}
printf("i=%d\n", i);
```

ou

```
int i;
for (i = 0; i <= 10; i++)
{
    // instructions qu'on veut répéter
}
printf("i=%d\n", i);
```

Quelle est la différence entre les deux ?

On peut voir deux styles pour incrémenter dans la boucle.

```
for (int a =0; a<3; ++a)
{
    printf("a=%d\n", a);
}
```

ou

```
for (int a =0; a<3; a++)
{
    printf("a=%d\n", a);
}
```

Quelle différence y a-t-il ?



Qu'affiche le programme suivant ?

```
#include <stdio.h>

int main()
{
    int i=0,j=0;

    int k,l;
    k = ++i;
    l = j++;
    printf("i=%d\n",i);
    printf("j=%d\n",j);
    printf("k=%d\n",k);
    printf("l=%d\n",l);
    return 0;
}
```

- Sert à répéter des instructions dans lesquelles la condition est vérifiée à la fin.
- On est donc sûr d'exécuter au moins une fois le bloc d'instructions à répéter.

```
do {bloc d instructions a repeter;  
} while (condition de rebouclage) ;
```

## Exemple de boucle "faire tant que"

```
#include <stdio.h>
int main () {
    int i = 0 ;
    do {
        printf ( "iteration %d \n", i ) ;
        i = i + 1 ;
    } while ( i < 10 ) ;
    printf ( "valeur de i apres la boucle : %d \n", i ) ;
    return 0 ;
}
```

- Dans la boucle while, le bloc peut ne jamais être exécuté.
- La condition est vérifiée avant le bloc.

```
while (condition de boucle) {  
    bloc d'instructions à répéter;  
}
```

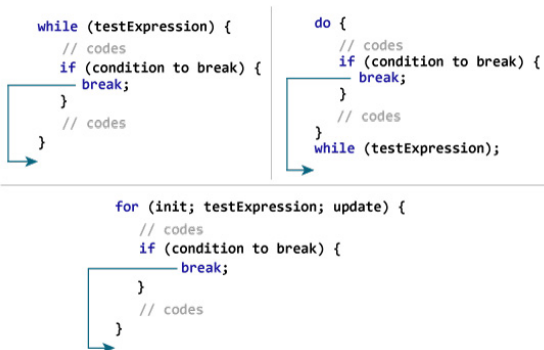
# Boucle "tant que" exemple

```
#include <stdio.h>

int main () {
    int i = 0 ;
    while ( i < 10) {
        printf ("iteration %d \n", i) ;
        i = i + 1 ;
    }
    printf ("valeur de i apres la boucle : %d \n", i) ;
    return 0 ;
}
```

Attention : il ne faut pas oublier d'incrémenter *i*.

- L'instruction break permet de sortir d'une boucle immédiatement si la condition est réalisée.
- Surtout utilisable avec if et switch.



Qu'affiche le programme suivant?

```
# include <stdio.h>
int main()
{
    int i;
    double nombre, somme = 0.0;
    for(i=1; i <= 10; ++i)
    {
        printf("Entrer un nombre n%d: ", i);
        scanf("%lf", &nombre);
        // si nombre negatif saisi alors fin de la boucle
        if(nombre < 0.0)
        {
            break;
        }

        somme += nombre; // somme = somme + nombre;
    }
    printf("Somme = %.2lf", somme);
    return 0;
}
```

L'instruction continue permet de sauter l'itération courante de la boucle et continue avec la prochaine itération.





Qu'affiche le programme suivant?

```
# include <stdio.h>
int main()
{
    int i;
    double nombre, somme = 0.0;
    for(i=1; i <= 10; ++i)
    {
        printf("Entrer un nombre n%d: ", i);
        scanf("%lf",&nombre);

        if(nombre < 0.0)
        {
            continue;
        }

        somme += nombre; // somme = somme + nombre;
    }
    printf("Somme = %.2lf",somme);
    return 0;
}
```