



Chapitre 3: Structures Répétitives (Itératives)



Objectifs



A la fin de ce chapitre, l'étudiant doit :

- Appliquer les structures itératives : **for, while, do...while,**
- Connaître la différence entre les trois boucles
- Appliquer chaque structure itérative dans son contexte.



Plan

- Introduction
- La boucle « **For** »
- Les boucles **while** et **do...while**
- La règle à retenir

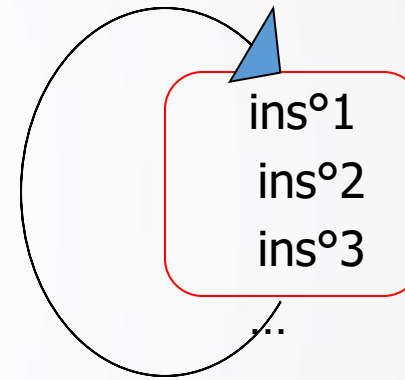
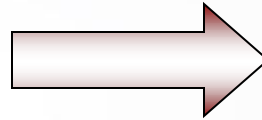
► Introduction



- Une ou plusieurs instructions qui se répètent:

Un cas
simple

```
ins°1  
ins°2  
ins°3  
ins°1  
ins°2  
ins°3  
ins°1  
ins°2  
ins°3
```



- Il existe 2 familles de boucles en langage C:
 1. **For** (nombre d'itérations connu à l'avance)
 2. **While** et **Do ... While** (nombre d'itérations non connu à l'avance).



La boucle « **for** »

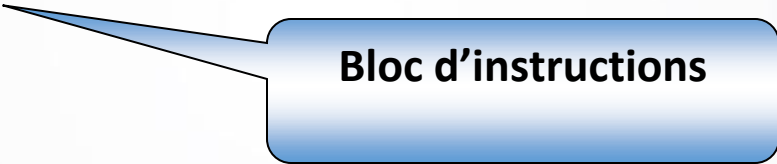


La boucle « for »



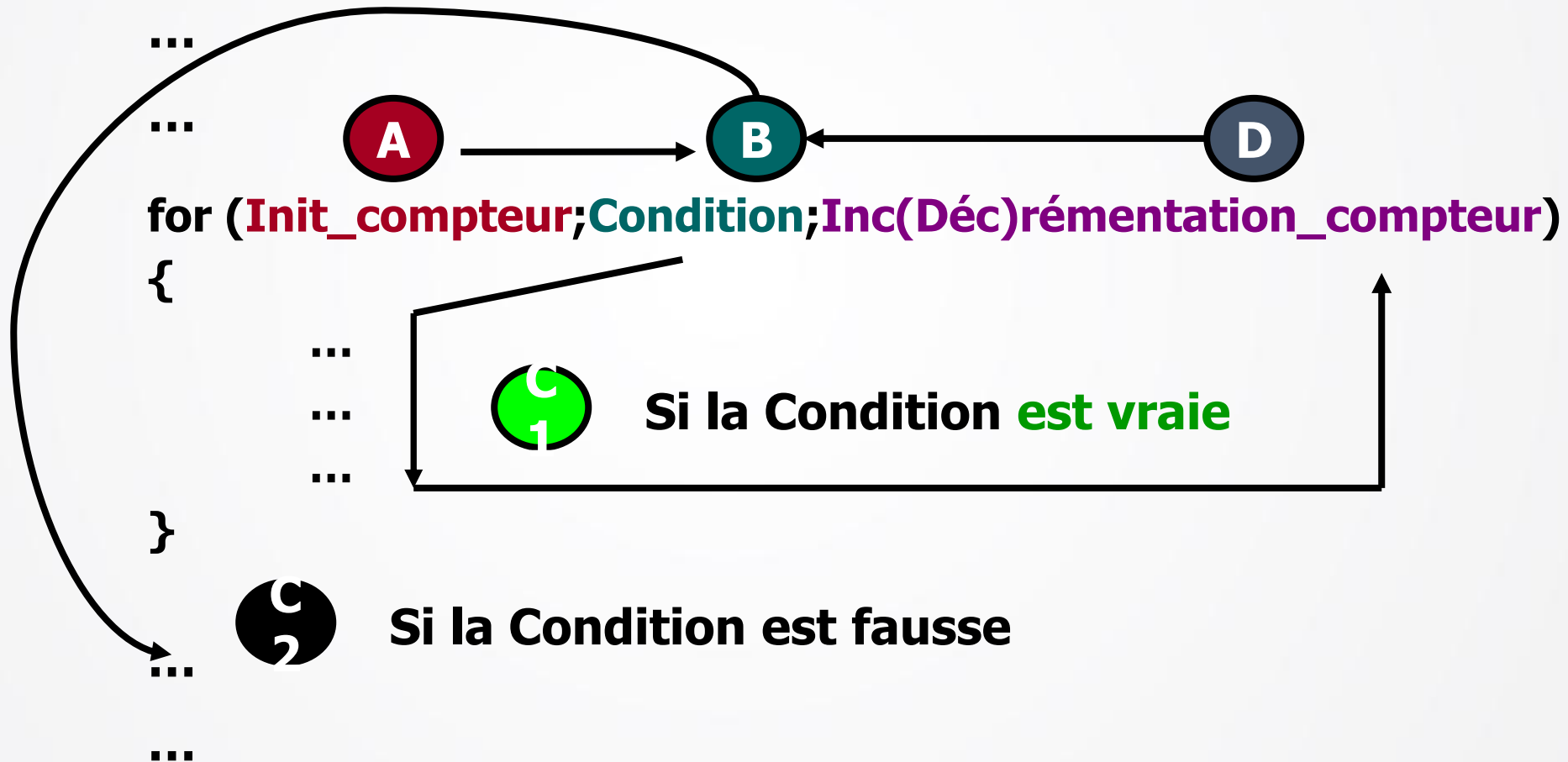
- Syntaxe de la boucle **For**:

```
for (Init_compteur;Condition;Inc(Déc)rémentation_compteur)  
{  
    ...  
    ...  
    ...  
}
```



- Cette boucle permet d'exécuter un bloc d'instructions un certain nombre de fois (ce nombre est connu à l'avance).
- Une variable d'itération (appelée **Compteur**) servira à compter le nombre de répétitions effectuées
- Si le bloc contient une seule instruction, les accolades ne sont plus nécessaires.

► Schéma de la boucle "For"



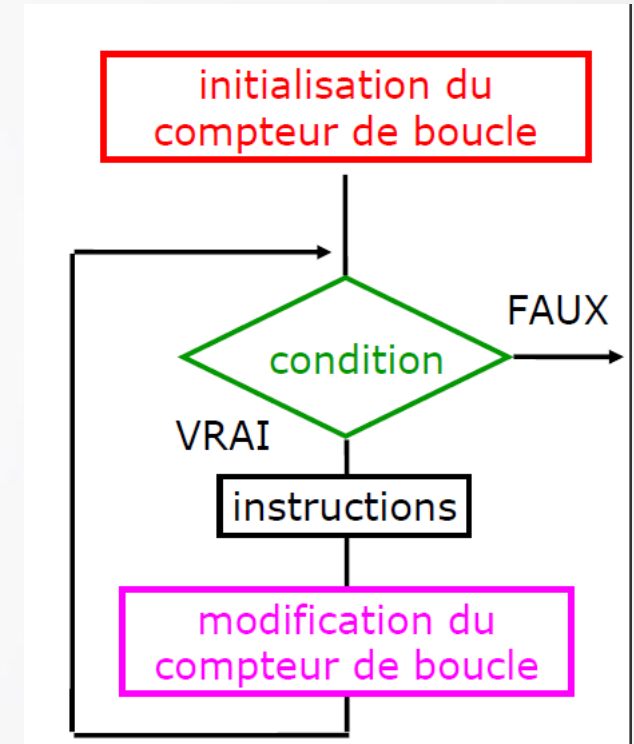


Exemple

// Programme utilisant une boucle for

```
#include <stdio.h>
void main ()
{
    int a, k=10;

    for (a=1; a <= k; a++)
        printf ("Bonjour")
} ;
```





Exerçons-nous?

// Programme utilisant une boucle for

```
#include <stdio.h>
```

```
void main ()
```

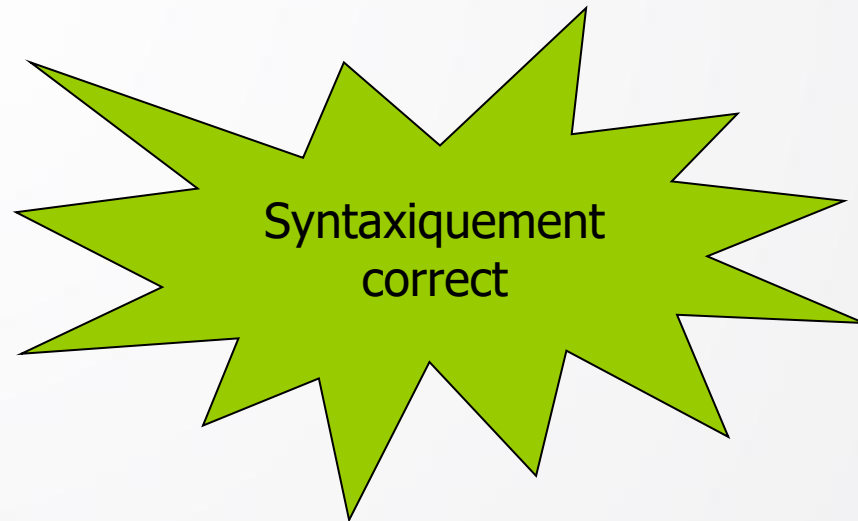
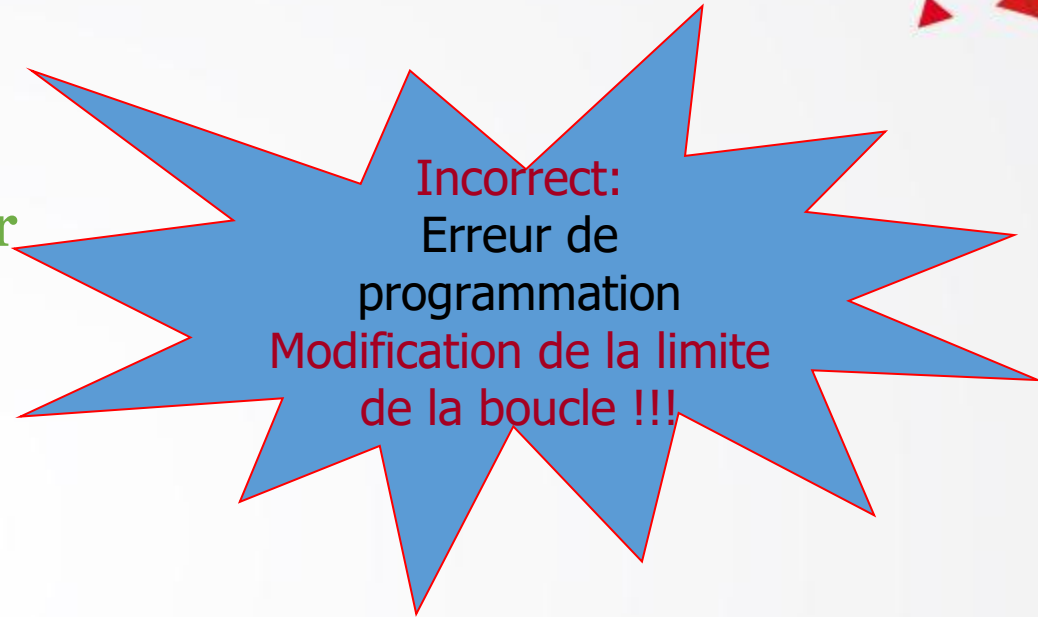
```
{
```

```
    int a, k=100;
```

```
    for (a=1; a <= k+100; a++)
```

```
        k = k+10;
```

```
}
```





La boucle : *while*

► La boucle **while**



- Tant qu'une condition est vraie, répéter un bloc d'instructions.

```
while (condition)
```

```
{
```

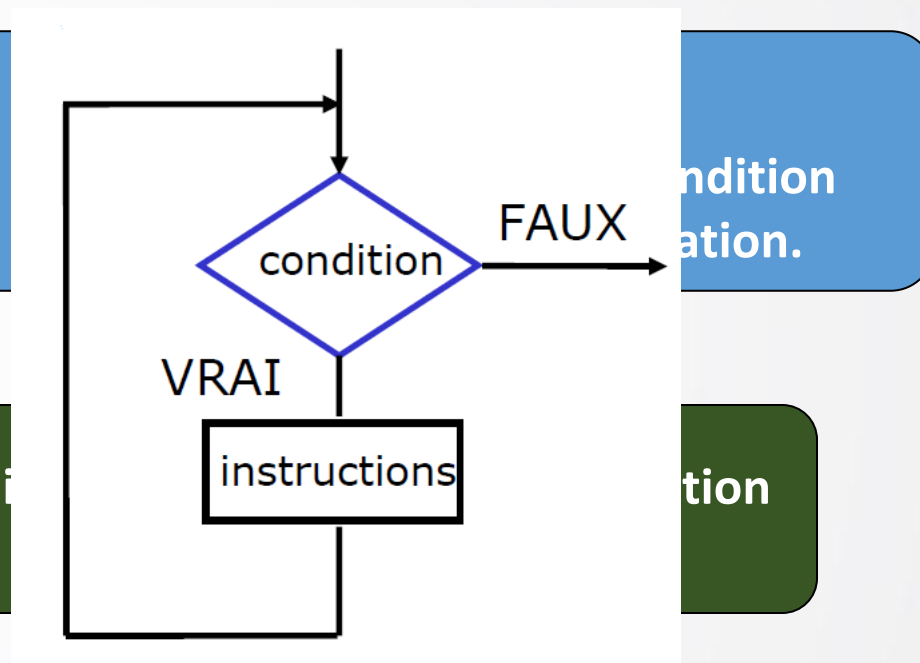
```
...
```

```
  bloc d'instructions
```

```
...
```

```
}
```

bloc d'i



Si la boucle ne comporte qu'une instruction, les accolades ne sont plus nécessaires.



Problème ?



Comment calculer la moyenne de plusieurs étudiants sachant qu'on ne connaît pas leur nombre?



La solution



```
#include <stdio.h>
void main()
{
    float Note1, Note2, Moyenne;
    char reponse = 'O';

    while (reponse == 'O') || (reponse == 'o')
    {
        printf ("Donner la note 1: ");
        scanf ("%f", &Note1);
        printf ("Donner la note 2: ");
        scanf ("%f", &Note2);
        Moyenne = (Note1 * 0.8) + (Note2 * 0.2);
        printf ("Moyenne de l'étudiant: %f", Moyenne);

        printf ("Voulez-vous calculer la moyenne d'un autre étudiant (O/N)?");
        scanf ("%c", &reponse);
    }
}
```



Que fait ce programme ?



```
#include <stdio.h>
void main ()
{
    int n = 2;
    while ( (n%2) == 0) && (n != 0) )
    {
        printf ("Donner un entier : ");
        scanf ("%d", &n);
    }
}
```


On sort de la boucle quand l'utilisateur saisit un entier impair ou nul.



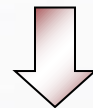
La boucle : `do .. while`

► La boucle **do..while**

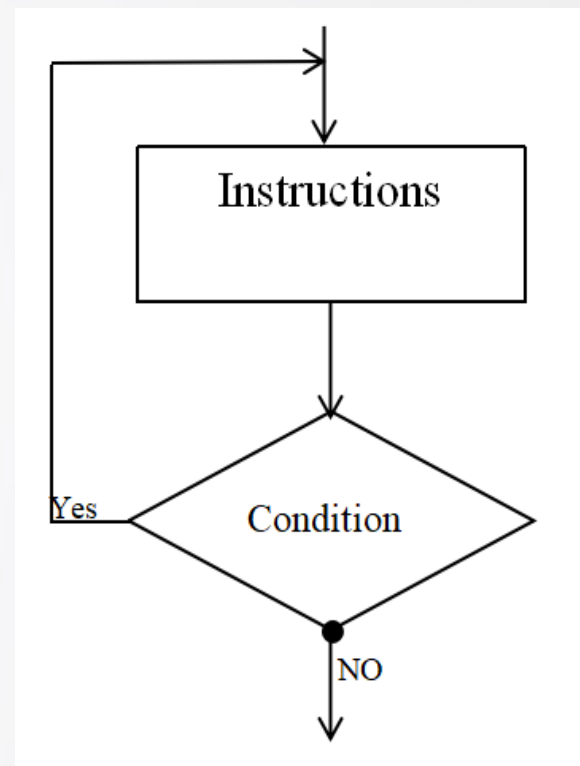
```
do {  
    bloc d'instructions  
}  
while (condition) ;
```



- La structure **do..while** ressemble à la structure **while**, sauf que...
- La structure **while** évalue la condition **avant** d'exécuter le bloc d'instructions, alors que la structure **do..while** évalue la condition **après** avoir exécuté le bloc d'instructions (à la fin de l'itération).



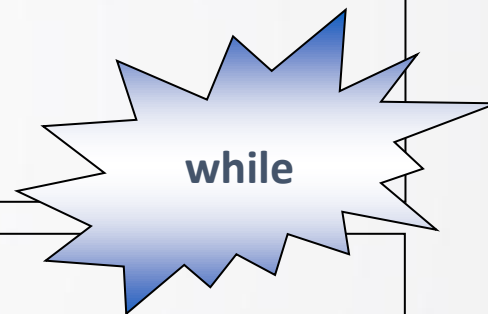
Avec la boucle *do..while* le bloc d'instructions est exécuté au moins une fois.



▶ while ou do..while ? (1)



```
int i;  
printf("Combien il vous reste de séances de cours aujourd'hui?" );  
scanf("%d", &i);  
do  
{    printf("Il reste %d séances ", i );  
    i--;  
}  
while (i > 0);
```



```
int i;  
printf("Combien il vous reste de séances de cours aujourd'hui?" );  
scanf("%d", &i);  
while (i > 0)  
{    printf("Il reste %d séances ", i );  
    i--;  
}
```



while ou do..while (2)



```
float note;  
do  
{  
    printf("Entrer votre note: " );  
    scanf("%f", &note);  
}  
while (note<0) || (note>20);
```

```
printf("Entrer votre note: " );  
scanf("%f", &note);  
while (note<0) || (note>20)  
{  
    printf("Entrer une note: " );  
    scanf("%f", &note);  
}
```





Application

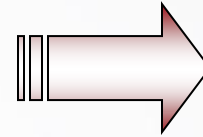
- Apporter les modifications nécessaires pour Ex 2 en ajoutant un contrôle de saisie sur les nombre $n1$ et $n2$, ces nombres doivent être positifs et $n2 > n1$.
- **Correction de l'exercice 5 et 6**



La règle à retenir



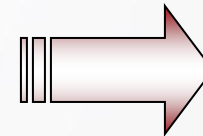
1. Si **on connaît** à l'avance le nombre de fois que le traitement sera répété



Boucle for

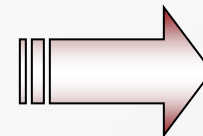
2. Si on **ne connaît pas** à l'avance le nombre de fois que le traitement sera répété:

- a) Si le traitement doit être exécuté **au moins une fois**



Boucle do...while

- b) Si le traitement peut ne pas être exécuté du tout **(0 fois)**



Boucle while