

Chapitre5:
**LES STRUCTURES
RÉPÉTITIVES**

Syrine KHIARI - Wafa NEJI – Salma HAJJEM

ESPRIT



2012-2013

Ecole Supérieure Privée
d'Ingénierie et de Technologies



Problématique



- Dans les problèmes quotidiens, on ne traite pas uniquement des séquences d'actions, sous ou sans conditions, mais il peut être fréquent d'être obligé d'exécuter un traitement (une séquence d'actions), **plusieurs fois !**

- Exemple:

- *Saisir les N notes d'un étudiant et calculer sa moyenne.*
 - Besoin d'une structure qui effectue une série de séquences d'écriture/lecture!

- On doit alors utiliser une structure algorithmique qui sert à "répéter" un certain nombre de fois une instruction ou un bloc d'instructions



*Une telle structure est appelée **structure répétitive***

Contexte et objectifs



- Dans le cadre de ce cours, l'objectif visé est la maîtrise des structures répétitives en C
- On distingue deux types de structures répétitives:
 - *L'itération*: quand on sait le nombre de fois que l'on va répéter un traitement :
 1. *La structure répétitive FOR ()*
 - *La répétition conditionnelle*: quand la poursuite d'une répétition dépend d'une condition :
 2. *La structure répétitive WHILE*
 3. *La structure répétitive DO WHILE*

1. La structure répétitive *FOR*

- *Définition* :



- La structure répétitive *FOR* est une structure répétitive qui itère le même traitement pour une plage de valeurs entières comprises entre une **borne inférieure** et une **borne supérieure**.
- L'arrêt du traitement se réalise lorsqu'on dépasse l'une des bornes.

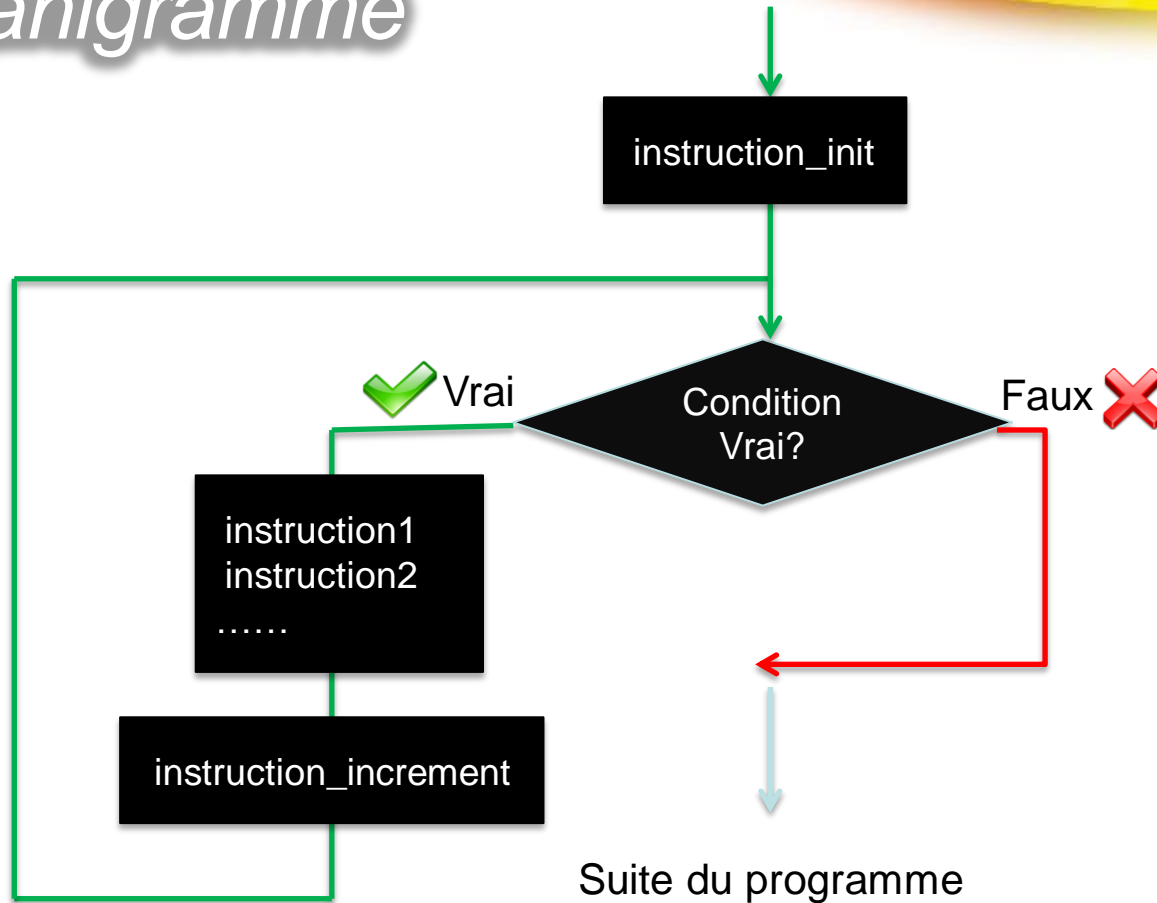
*La structure
répétitive FOR...??*



1. La structure répétitive *FOR*



- *Organigramme*



1. La structure répétitive *FOR*

• Syntaxe:



En Langage C

```
for (instruction_init ;  
condition ;  
instruction_increment)  
{  
    Instruction1;  
    Instruction2;  
    .....  
}
```



Le for prend **trois** arguments encadrés par deux parenthèse (..) et séparés par des points-virgules.



– ***instruction_init*** : Une instruction d'initialisation.. En général on initialise un compteur avec cette instruction. Exemple $i=0$.



– ***condition*** : Une condition pour continuer la boucle. En général on indique qu'un compteur doit être inférieur à une certaine valeur comme par exemple $i < 100$.



– ***instruction_increment***: Une instruction qui est exécutée à chaque fin de boucle, juste avant de faire le test sur la condition pour continuer. En général, on incrémente le compteur car ce n'est pas automatique.

1. La structure répétitive *FOR*

- *Exemple*

- On veut écrire un programme qui affiche les entiers (de 1 à 10) et leur carré



```
#include <stdio.h>
void main()
{ int i ;
  /* Une boucle qui affiche des entiers et leur carré */
  for (i =1;i <=10; i++)
  {
    printf ("i=%d -> i²=%d\n",i,i*i);
  }
}
```

1. La structure répétitive *FOR*



- *Exercice d'application*

- Ecrire un programme en C qui demande un nombre de départ, et qui affiche ensuite les dix nombres suivants.
 - Exemple: si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.




2. La structure répétitive *WHILE*

• *Définition :*



- La structure répétitive "*WHILE*" ou "*TantQue... Faire*" permet de répéter un traitement tant qu'une expression conditionnelle est vraie.
- Si d'emblée, la condition n'est pas vraie, le traitement ne sera pas exécuté.

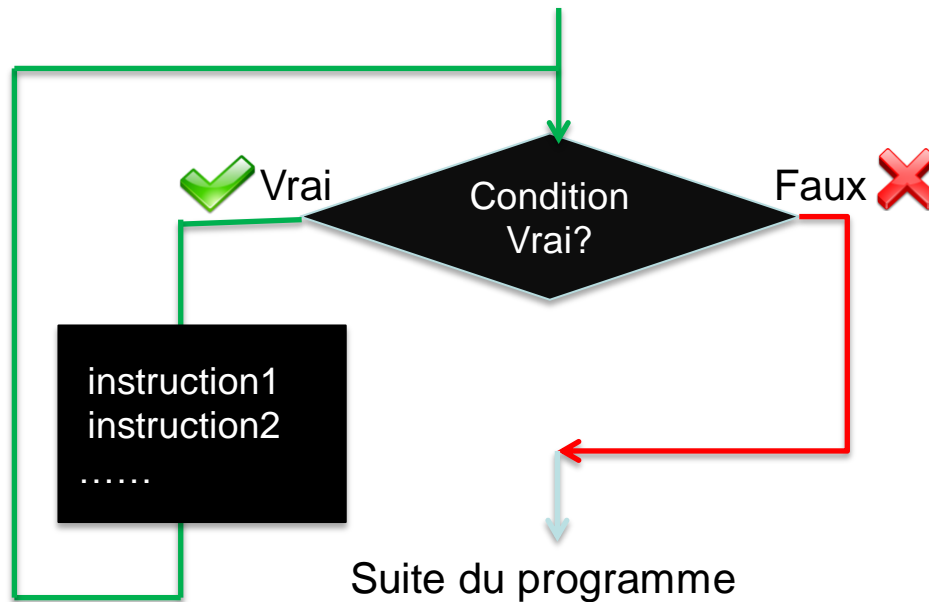
 On voit donc que la structure répétitive "*WHILE*" a un point commun avec la structure conditionnelle: si la condition n'est pas vraie, le traitement n'est pas exécuté.

*La structure
répétitive WHILE...?*



2. La structure répétitive *WHILE*

- *Organigramme*



! Dans cette structure la condition est testée au début

2. La structure répétitive *WHILE*

- *Syntaxe:*



En Langage C

```
While (condition)  
{  
  Instruction1 ;  
  Instruction2 ;  
  .....  
}
```



-Toutes les conditions sont entre deux parenthèses (*.....*)



Tant que la *condition* est vraie alors l'instruction ou le bloc d'instructions qui suit le "**While**" est exécuté.



S'il y a plusieurs instructions après le "**While**", on les encadre par des accolades : { }

2. La structure répétitive *WHILE*

- *Exemple*

- On veut afficher les entiers allant de 0 à 10



```
#include <stdio.h>
void main()
{
int i;
i=0;
```

```
/* Boucle while <Tant que... faire> */
```

```
while(i<=10)
```

```
{
```

```
printf("i= %d \n",i);
```

```
i++;
```

```
}
```

```
}
```



2. La structure répétitive *WHILE*



- *Exercice d'application*

- Ecrire un programme en C qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre.
 - Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$



3. La structure répétitive *DO WHILE*

• *Définition :*



- La structure répétitive "*DO WHILE*" ou "*Répéter...Jusqu'à* " permet de répéter un traitement jusqu'à ce qu'une condition soit vraie.
- Le traitement est exécuté, puis la condition est vérifiée.
 - On voit donc que la structure répétitive "*DO WHILE*" a aussi un point commun avec la structure conditionnelle: si la condition à la fin n'est pas vraie, le traitement n'est pas exécuté.

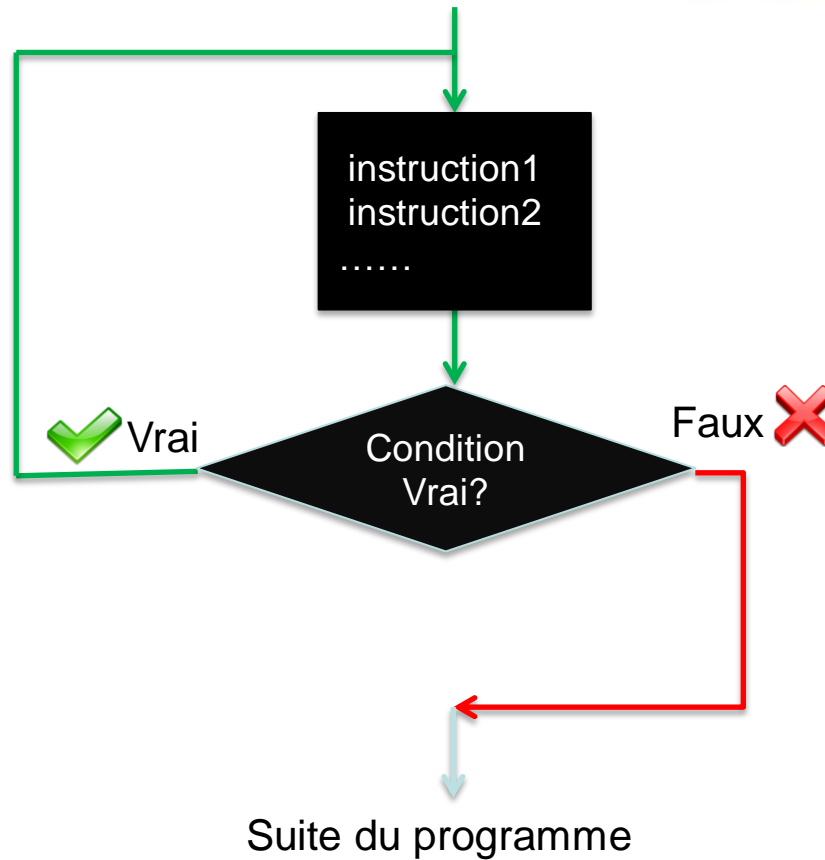
*La structure
répétitive DO
WHILE...?*



3. La structure répétitive *DO WHILE*



- *Organigramme*



! Dans cette structure la condition est testée à la fin

3. La structure répétitive *DO WHILE*

- *Syntaxe:*



En Langage C

Do

{

Instruction1;

Instruction2;

.....

}

While (*condition*);



Lorsque "**Do**" est rencontré l'instruction ou le bloc d'instructions qui suit le est exécutée puis la *condition* est testée, si elle est vérifiée la boucle est répétée sinon l'exécution continue à la première instruction après le "**While**".



S'il y a plusieurs instructions après le "**Do**", on les encadre par des accolades : **{ }**



- Toutes les conditions sont entre deux parenthèses (*.....*)
- L'instruction "**While** (*condition*)" est suivie d'un point-virgule ;. (contrairement à la version de la structure répétitive *WHILE*)

3. La structure répétitive *DO WHILE*

- *Exemple*

- On veut afficher les entiers allant de 0 à 10



```
#include <stdio.h>
void main()
{
int i;
i=0;

/* Boucle do while < Répéter...Jusqu'à > */
    do
    {
        printf("i= %d \n",i);
        i++;
    } while(i<=10) ;
}
```

3. La structure répétitive *DO WHILE*



- *Exercice d'application*

- Ecrire un programme en C qui demande à l'utilisateur un nombre compris entre 1 et 10 jusqu'à ce que la réponse convienne.



Différence entre *FOR*, *WHILE* et *DO WHILE* ?



Structures itérative	Fonctionnalités
<i>FOR</i> :	Le nombre d'itérations est connu à l'avance.
<i>WHILE</i> :	Condition vérifiée avant le traitement : les instructions peuvent ne pas être exécutées Condition de continuation : le traitement est répété si la condition est <i>vraie</i> .
<i>DO WHILE</i> :	Condition vérifiée après le traitement : les instructions sont forcément exécutées au moins une fois. Condition d'arrêt : le traitement est répété si la condition est <i>fausse</i> .



Merci Pour
votre
attention!

