



Université
Internationale
de Casablanca

UNIVERSITE INTERNATIONALE DE CASABLANCA

Cours Algorithmme: Boucles

INSTRUCTIONS ITÉRATIVES

BOUCLES

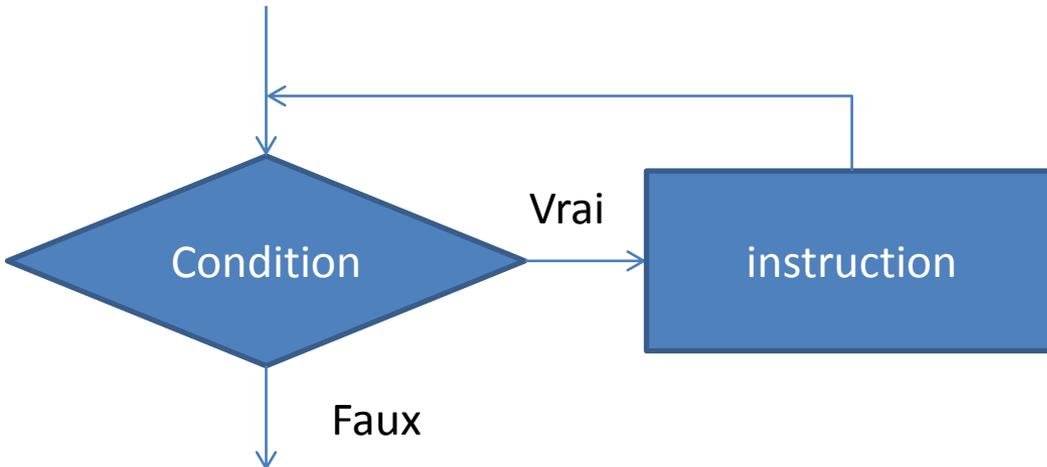
Boucles

- Les boucles servent à répéter l'exécution d'un groupe d'instructions un certain nombre de fois
- On distingue trois sortes de boucles en langages de programmation :
 - Les ***boucles tant que*** : on y répète des instructions tant qu'une certaine condition est réalisée
 - Les ***boucles répéter...jusqu'à*** : on y répète des instructions jusqu'à ce qu'une certaine condition soit réalisée
 - Les ***boucles pour*** ou ***avec compteur*** : on y répète des instructions en faisant évoluer un compteur (variable particulière) entre une valeur initiale et une valeur finale

BOUCLE TANT QUE

Boucles « TANT QUE »

Tant Que (condition)
Instructions;
Fin Tant Que



- la condition (dite condition de contrôle de la boucle) est évaluée avant chaque itération
 - si la condition est vraie, on exécute instructions (corps de la boucle), puis, on retourne tester la condition. Si elle est encore vraie, on répète l'exécution, ...
 - si la condition est fausse, on sort de la boucle et on exécute l'instruction qui est après Fin TantQue

Remarques

- Le nombre d'itérations dans une boucle Tant Que n'est pas connu au moment d'entrée dans la boucle. Il dépend de l'évolution de la valeur de condition.
- Une des instructions du corps de la boucle doit absolument changer la valeur de condition de vrai à faux (après un certain nombre d'itérations), sinon le programme tourne indéfiniment.

Attention aux boucles infinies

Exemple de boucle infinie

- Si on considère le bloc d'instruction suivant:

$i \leftarrow 2;$

Tant Que ($i > 0$)

$i \leftarrow i+1;$ (*attention aux erreurs de frappe : + au lieu de -*)

Fin TantQue

BOUCLE POUR

Boucles « POUR »

Pour *compteur* allant de *initiale* à *finale* par pas de *valeurdupas* **Faire**

instructions;

FinPour

- **Compteur** est une variable de type *entier*. Elle doit être déclarée.
- **Valeurdupas** est un entier qui peut être positif ou négatif. Valeurdupas peut ne pas être mentionné, car par défaut sa valeur est égal à 1.
Dans ce cas, le nombre d'itérations est égal à finale - initiale + 1 .
- **Initiale et finale** peuvent être des valeurs, des variables définies avant le début de la boucle ou des expressions de même type que compteur .

Boucles « POUR »

- **Remarque :**
 - Le nombre d'itérations dans une boucle Pour est connu avant le début de la boucle.

Déroulement de la boucle POUR

- La valeur initiale est affectée à la variable compteur,
- On compare la valeur du compteur et la valeur de finale :
 - Si la valeur du compteur est $>$ à la valeur finale dans le cas d'un pas positif (ou si compteur est $<$ à finale pour un pas négatif), on sort de la boucle et on continue avec l'instruction qui suit Fin Pour.
 - Si compteur est \leq à finale dans le cas d'un pas positif (ou si compteur est \geq à finale pour un pas négatif), instructions seront exécutées.
 - Ensuite, la valeur de compteur est incrémentée de la valeur du pas si pas est positif (ou décrémenté si pas est négatif).
 - On recommence l'étape 2 : La comparaison entre compteur et finale est de nouveau effectuée, et ainsi de suite ...

Exemple

- Calcul de x à la puissance n où x est un réel non nul et n un entier positif ou nul

Remarque

- Il faut éviter de modifier la valeur du compteur (et de finale) à l'intérieur de la boucle. En effet, une telle action :
 - perturbe le nombre d'itérations prévu par la boucle Pour
 - rend difficile la lecture de l'algorithme
 - présente le risque d'aboutir à une boucle infinie

Exemple :

Pour i allant de 1 à 5 faire

$i \leftarrow i - 1$

 écrire(" i = ", i)

Fin Pour

Pour vs TantQue

- Si le nombre d'itérations est connu à l'avance:
 - choisir la boucle *pour*
- Si la boucle doit s'arrêter quand survient un évènement:
 - choisir la boucle *tant que*

Pour vs Tant Que

- La boucle Pour est un cas particulier de Tant Que (cas où le nombre d'itérations est connu et fixé) .
- Tout ce qu'on peut écrire avec Pour peut être remplacé avec Tant Que (la réciproque est fausse)

Pour vs TantQue

Pour compteur allant de initiale à finale par pas valeurdupas

instructions ;

Fin Pour

compteur \leftarrow initiale;

Tant Que (compteur \leq finale)

instructions ;

compteur \leftarrow compteur+valeurdupas;

Fin TantQue

BOUCLE RÉPÉTER JUSQU'À

Boucles « REPETER ... JUSQU'À »

Répéter

instructions

Jusqu'à (condition)

- Condition est évaluée après chaque itération:
 - les instructions entre *Répéter* et *jusqu'à* sont exécutées au moins une fois et leur exécution est répétée jusqu'à ce que condition soit vraie (tant qu'elle est fausse).

Récapitulatif

- Si on peut déterminer le nombre d'itérations avant l'exécution de la boucle, il est plus naturel d'utiliser la boucle *Pour*.
- S'il n'est pas possible de connaître le nombre d'itérations avant l'exécution de la boucle, on fera appel à l'une des boucles *Tant Que* ou *Répéter ... Jusqu'à*.
- Pour le choix entre *Tant Que* et *Jusqu'à* :
 - Si on doit tester la condition de contrôle avant de commencer les instructions de la boucle, on utilisera *Tant Que*
 - Si la valeur de la condition de contrôle dépend d'une première exécution des instructions de la boucle, on utilisera *Répéter ... Jusqu'à*.

À suivre....