

IFT159

Analyse et programmation

Chapitre 5 — Structures itératives

Gabriel Girard

Département d'informatique



21 septembre 2008



Analyse et programmation

1/38

Chapitre 5 — Structures itératives

- 1 Nécessité de l'itération
- 2 Concept de boucle
- 3 Boucle : « Tant que ... répéter »
 - Énoncés abrégés
 - Exemples
 - Boucle de comptage et boucle conditionnelle
- 4 Boucle : « Répéter un certain nombre de fois »
- 5 Boucle : « Répéter ... tant que »
- 6 Boucle : « Récursivité »
- 7 Boucle emboîtées
- 8 Exercices



Analyse et programmation

2/38

Nécessité de l'itération

- Résoudre un problème pour un cas c'est le résoudre pour une quasi-infinité de cas.
- Il suffit de répéter la solution autant de fois que nécessaire.
- Deux possibilités :
 - 1 Recopier le code un certain nombre de fois.
 - 2 Exécuter le même code un certain nombre de fois.



Analyse et programmation

4/38

Boucle

- Exécuter le code plusieurs fois s'appelle « faire une boucle ».
- Le corps de la boucle contient les énoncés à répéter.
- **DANGER ******* boucle sans fin.



Analyse et programmation

6/38

Boucle : conception

- Pour tous les cas.
- Pour chaque cas.
- Exemple :
 - Pour tous les étudiants.
 - Pour chaque employé.

Types d'énoncés de contrôle

- Tant qu'une condition est remplie répéter.
- Répéter tant qu'une condition est remplie.
- Répéter jusqu'à ce qu'une condition soit vraie.
- Répéter un certain nombre de fois.
- Récursivité.

Boucle : Tant qu'une condition est remplie répéter

- Syntaxe

```
while (expression logique)           // condition
    énoncé (simple ou composé)      // corps
```

Boucle : Tant qu'une condition est remplie répéter

- Syntaxe.. suggérée

```
while (expression logique) // condition
{
    énoncé                      // corps
}
```

Boucle : Tant qu'une condition est remplie répéter

- La boucle se répète tant que la condition est vraie.
- Après l'exécution de l'énoncé, le contrôle est retourné à la condition.
- La variable de contrôle doit être modifiée dans le corps de la boucle.

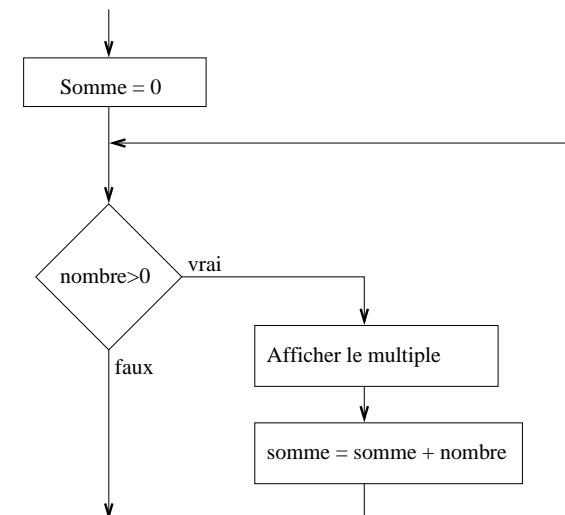
Boucle : Tant qu'une condition est remplie répéter

- Exemple : *Écrire un programme qui imprime tous les multiples d'un nombre donné, et ce jusqu'à une valeur finale donnée.*

Boucle : Tant qu'une condition est remplie répéter

- Éléments importants à considérer :
 - Initialisation de la variable.
 - La condition de répétition.
 - La modification de la variable de contrôle.

Boucle : Tant qu'une condition est remplie répéter



Énoncés abrégés

- `somme = somme + nombre; → somme += nombre;`
- `somme = somme * valeur; → somme *= valeur;`
- `somme = somme + 1; → somme ++;`
- `somme = somme - 1; → somme --;`
- `++somme; ≠ somme++;`
- `--somme; ≠ somme--;`

Exemples

- *Cumuler des sommes partielles.*
- *Afficher les puissances d'un nombre.*

Boucle de comptage

- Les boucles de comptage s'exécutent un nombre fixe de fois.
- Exemple :
Écrire un programme qui affiche le n premiers multiples d'un nombre.

Boucle de comptage

- Principes de la boucle :
 - Initialisation de la variable de contrôle.
 - Test de la variable de contrôle.
 - Modification de la variable de contrôle.

Boucle conditionnelle

- Si énumération par comptage impossible → (1) sentinelle.
- Exemple :
Calculer la moyenne d'un certain nombre de notes.
- Si énumération par comptage impossible → (2) indicateur.
- Exemple :
On recherche le premier caractère qui soit un chiffre.

Boucle : Répéter un certain nombre de fois (for)

- La boucle for sert lorsque le nombre d'itérations est connu.
- C'est une réécriture d'une boucle de comptage.
- On regroupe « initialisation, modification, test » de la variable de contrôle dans l'entête.

Boucle : for

- Syntaxe
`for (définition/init.; test; modification)`
 énoncé (simple ou composé) `// corps`

Boucle de comptage

- Principes de la boucle :
 - Initialisation de la variable de contrôle.
 - Test de la variable de contrôle.
 - Modification de la variable de contrôle.

Boucle : **for**■ Comparaison **for** – **while**

<pre>for (int ligne=0; ligne <= n; ligne++) { cout << endl; }</pre>	<pre>int ligne; ligne = 0; while (ligne <= n) { cout << endl; ligne++; }</pre>
--	---

Boucle : **for**

- La variable de contrôle peut être utilisé dans le corps de la boucle.
- Exemple : *Trouver les carrés et les racines carrés de n premiers entiers.*

Boucle : **for**

■ Exemples :

- 1 *Faire la somme des nombres impairs inférieurs à une certaine limite.*
- 2 *Faire une conversion de degrés Fahrenheit à degrés Celcius pour des températures variant de +50 à -50 par intervalles de 5 degrés.*

Boucle : Répéter tant qu'une condition est remplie
(do ..while)

- Similaire à la boucle **while**.
- On l'utilise quand la boucle doit s'exécuter au moins une fois.
- Exemple : *Recherche du premier chiffre dans une suite de caractères.*

Boucle : Répéter tant qu'une condition est remplie (do ..while)

■ Syntaxe

```
do
    énoncé (simple ou composé) // corps
while (expression logique);    // condition
```

Boucle : Répéter tant qu'une condition est remplie (do ..while)

■ Syntaxe ... suggérée...

```
do
{
    énoncé                                // corps
{
while (expression logique); // condition
```

Boucle : Récursivité

- Une fonction qui s'appelle elle-même plante une boucle.
- Exemple : *Trouver le factoriel de n.*

Boucles emboîtées

- Une boucle peut contenir une autre boucle.
- Exemple : *On fait subir à un certain nombre de voitures un nombre fixe du même essai routier ayant une distance prédéterminé. À chaque essai on note la consommation d'essence en litres/100kms. On veut savoir, à la fin de tous les essais routiers, la consommation d'essence de chacune des voitures testées.*

Boucles emboîtées

- Les boucles emboîtées peuvent être dépendantes une de l'autre.

- Exemple :

Tracer un triangle isocèle d'une hauteur donnée.

```

      *
     ***
    *****
   ********
  **********
 
```

Exercices

- 1 Demander à l'utilisateur de rentrer des caractères. S'arrêter et indiquer la position du premier caractères qui est un chiffre.
- 2 Calculer tous les multiples d'un nombre n inférieurs à une valeur f : l'utilisateur choisi n et f .
- 3 Calculer m multiples d'un nombre n : l'utilisateur choisi n et m .
- 4 Calculer la moyenne de n notes : les notes et n sont décidés par l'utilisateur.
- 5 Calculer la moyenne d'une série de notes. On s'arrête lorsque l'utilisateur rentre une note négative.
- 6 Dessiner un triangle isocèle (cf. acétate précédente) de la hauteur voulue.