

# IFT159

## Analyse et programmation

### Chapitre 5 — Structures itératives

Gabriel Girard

Département d'informatique



23 septembre 2015

# 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices



# Nécessité de l'itération

- Résoudre un problème pour un cas c'est le résoudre pour une quasi-infinité de cas.

# 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.

# 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 :

# 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.

# 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.

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Boucle

- Exécuter le code plusieurs fois s'appelle « faire une boucle ».

# Boucle

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

# 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.

# Boucle : conception

- Pour tous les cas.

# Boucle : conception

- Pour tous les cas.
- Pour chaque cas.

# Boucle : conception

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

# Boucle : conception

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

# 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.

# Types d'énoncés de contrôle

- Tant qu'une condition est remplie répéter.
- Répéter tant qu'une condition est remplie.

## 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.

# 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.

# 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.



# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# 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.

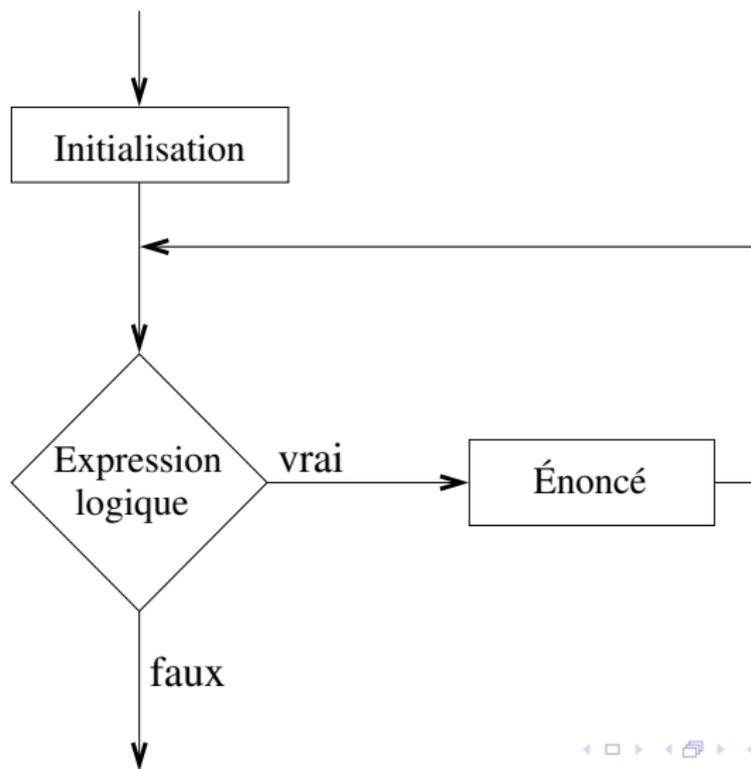
## Boucle : Tant qu'une condition est remplie répéter

- La boucle se répète tant que la condition est vraie.
- Après l'exécution de l'énoncé, le contrôle est retourné à la condition.

## 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



## 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 :

# Boucle : Tant qu'une condition est remplie répéter

- Éléments importants à considérer :
  - Initialisation de la variable.

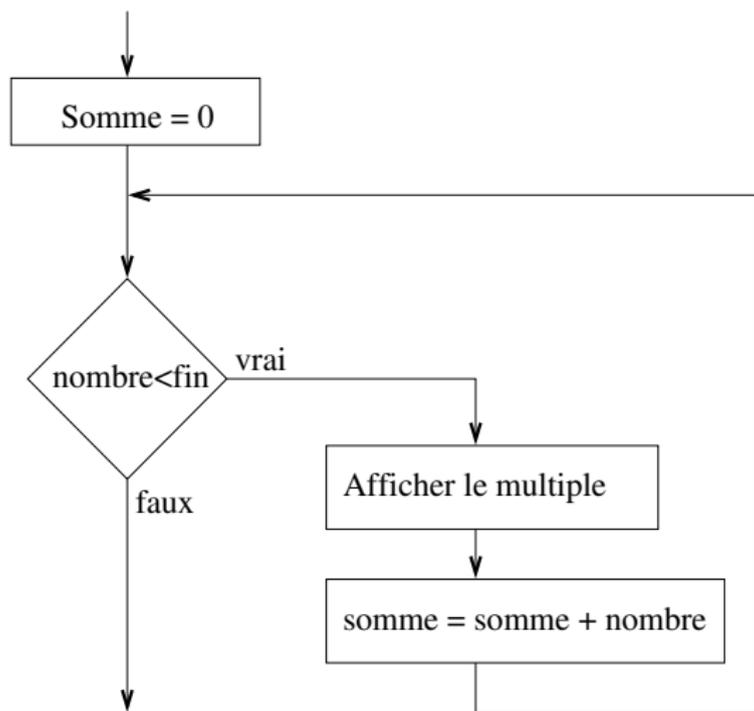
# Boucle : Tant qu'une condition est remplie répéter

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

# 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;`



# Énoncés abrégés

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



# Énoncés abrégés

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

# Énoncés abrégés

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



# Énoncés abrégés

- `somme = somme + nombre;`  $\longrightarrow$  `somme += nombre;`
- `somme = somme * valeur;`  $\longrightarrow$  `somme *= valeur;`
- `somme = somme + 1;`  $\longrightarrow$  `somme ++;`
- `somme = somme - 1;`  $\longrightarrow$  `somme --;`
- `++somme;`  $\neq$  `somme++;`



# Énoncés abrégés

- `somme = somme + nombre;`  $\longrightarrow$  `somme += nombre;`
- `somme = somme * valeur;`  $\longrightarrow$  `somme *= valeur;`
- `somme = somme + 1;`  $\longrightarrow$  `somme ++;`
- `somme = somme - 1;`  $\longrightarrow$  `somme --;`
- `++somme;`  $\neq$  `somme++;`
- `--somme;`  $\neq$  `somme--;`



# Exemples

- *Cumuler des sommes partielles.*

# 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.

# 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.

# Boucle de comptage

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

# 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.

# Boucle conditionnelle

- Si énumération par comptage impossible → (1) sentinelle.
- Exemple :  
*Calculer la moyenne d'un certain nombre de notes.*

# 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.

# 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.*

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Boucle : Répéter un certain nombre de fois (for)

- La boucle `for` sert lorsque le nombre d'itérations est connu.

## 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.

## 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 :



# Boucle de comptage

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

# Boucle de comptage

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

# 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**

```
for (int ligne=0 ; ligne <= n ; ligne++)  
{  
    cout << endl ;  
}
```

```
int ligne ;  
ligne = 0 ;  
while (ligne <= n)  
{  
    cout << endl ;  
    ligne++  
}
```

# Boucle : for

- La variable de contrôle peut être utilisé dans le corps de la boucle.

# 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 :

# Boucle : for

## ■ Exemples :

- 1 *Faire la somme des nombres impairs inférieurs à une certaine limite.*

# 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.*



# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

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

- Similaire à la boucle `while`.

# 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.



## 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
```



# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Boucle : Récursivité

- Une fonction qui s'appelle elle-même plante une boucle.

# Boucle : Récursivité

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

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Boucles emboîtées

- Une boucle peut contenir une autre boucle.

# Boucles emboîtées

- Une boucle peut contenir une autre boucle.
- Exemple : *On fait subir à un certain nombre de voitures un nombre fixe (4) du même essai routier ayant une distance prédéterminé (61.7 km). 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.*

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

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

## Exemple d'erreurs

- Mariner 1 : Décollage raté à cause d'une mauvaise syntaxe dans la boucle for (". " au lieu de ", ") : `DO 100 I=1.10` (plutôt que "1,10")



## Exemple d'erreurs

- Mariner 1 : Décollage raté à cause d'une mauvaise syntaxe dans la boucle for (". " au lieu de ", ") : `DO 100 I=1.10` (plutôt que "1,10")
- Les cartes d'Apple Maps qui donnaient des directions vers nulle part. Dans la carte d'Apple, des lacs entiers, des gares et des attractions touristiques étaient absents ou mal étiquetés. Ainsi la gare principale d'Auckland en Nouvelle Zélande se trouvait en plein milieu de l'océan.

## Exemple d'erreurs

- Mariner 1 : Décollage raté à cause d'une mauvaise syntaxe dans la boucle for (". " au lieu de ", ") : `DO 100 I=1.10` (plutôt que "1,10")
- Les cartes d'Apple Maps qui donnaient des directions vers nulle part. Dans la carte d'Apple, des lacs entiers, des gares et des attractions touristiques étaient absents ou mal étiquetés. Ainsi la gare principale d'Auckland en Nouvelle Zélande se trouvait en plein milieu de l'océan.
- World of Warcraft : l'incident du sang corrompu

## Paroles à retenir

- Gerald Weinberg

If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.

## Paroles à retenir

- Gerald Weinberg  
If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.
- Martin Golding  
Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.

# Paroles à retenir

- Gerald Weinberg  
If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.
- Martin Golding  
Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.
- Edward V Berard  
Walking on water and developing software from a specification are easy if both are frozen.

## Paroles à retenir

- Gerald Weinberg  
If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.
- Martin Golding  
Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.
- Edward V Berard  
Walking on water and developing software from a specification are easy if both are frozen.
- Anonymous  
In order to understand recursion, one must first understand recursion.

# Structures sélectives

- 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 Exemples d'erreurs et de paroles à retenir
- 9 Exercices

# Exercices

- 1 Écrire une boucle contrôlée par indicateur qui lit 2 entiers à chaque fois, et ce tant que l'un n'est pas divisible par l'autre.
- 2 Écrire un programme qui lit une suite de nombres et affiche les deux nombres ayant les valeurs minimale et maximale.
- 3 Écrire un programme qui, étant donné un entier impair, réalise le triangle suivant (si l'entier vaut 7 (pour 7 lignes)) :

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

#### 4 Que réalise le programme suivant :

```
void main()
{
    int ligne, milieu ;
    cout << "Donner un entier: " ;
    cin >> ligne ;
    if ( (ligne % 2) != 1) ligne++ ;
    milieu = ligne / 2 + 1 ;
    for (int i = 1 ; i <= ligne ; i++)
    {
        for (int j = 1 ;j <= !(i / milieu) * (i % milieu) +
                (i / milieu) * milieu + -
                (i / milieu) * (i % milieu) ; j++)
            cout << j << " " ;
        cout << endl ;
    }
}
```