

Cours 3 : Boucles

Laurent Debize



TIIS1

Initiation à l'algorithmique

① Boucles

pour
tant que

Les boucles

Les boucles permettent de répéter un bloc d'instructions plusieurs fois de suite.

Il existe deux boucles principales :

- La boucle **pour** : répéter un bloc d'instructions un certain *nombre de fois connu à l'avance*
↔ idéal quand **on connaît** le nombre de répétitions
- La boucle **tant que** : répéter un bloc d'instructions *tant qu'une condition est vérifiée*
↔ idéal quand **on ne connaît pas** le nombre de répétitions

Vocabulaire

- Une répétition d'une boucle s'appelle une **itération**.

Boucle pour

Pour répéter un bloc d'instructions un certain **nombre de fois connu à l'avance**

Structure

pour i de 0 à 9 **faire**
| instructions à effectuer
fin

- La variable i est un **compteur**
- Elle prend initialement la valeur 0 et augmente **automatiquement** de 1 à chaque tour
- On sort de la boucle lorsque la valeur finale est dépassée.
- On peut (ou pas) utiliser la variable i dans la boucle, mais on ne doit **jamais** modifier sa valeur.

Boucle pour

Si on veut que i augmente d'une valeur différente de 1 à chaque tour, il faut modifier le pas :

pour i de 0 à 9 **par pas de 2 faire**
| instructions à effectuer
fin

Boucle pour

Exemple 1 : simple affichage d'un texte

Algorithme 1 : Le rébarbatif

début

pour i de 1 à 10 **faire**
 Afficher " Je radote !"

fin

fin

Boucle pour

Exemple 2 : utilisation du compteur pour l'affichage

Algorithme 2 : Le rébarbatif qui compte

début

pour i de 1 à 10 **faire**

 Afficher " Je radote, cela fait " i " fois que je le dis !"

fin

fin

Boucle pour

Exemple 3 : utilisation du compteur dans les calculs

Algorithme 3 : Factorielle de n ($n!$) avec pour

Variables :

produit, *n* : entiers

début

Afficher "Entrez un nombre entier :"

Lire *n*

produit $\leftarrow 1$

pour *i* de 1 à *n* faire

| *produit* \leftarrow *produit* \times *i*

fin

Afficher "Factorielle de " *n* " = " *produit*

fin

Boucle tant que

Pour répéter un bloc d'instructions **tant qu'une condition est vérifiée**

Structure

tant que *condition* **faire**

| instructions à effectuer

fin

- Le test de la condition est effectué **avant** d'entrer dans la boucle.
- Si **condition** est vraie, on rentre dans la boucle et on exécute les instructions
- Une fois toutes les instructions effectuées, on remonte **automatiquement** au **tant que** et on évalue à nouveau la condition
- Si **condition** est fausse, **on sort de la boucle**, c'est-à-dire qu'on se place après le mot **fin** sans effectuer les instructions à l'intérieur de la boucle.

Boucle tant que

Exemple

Algorithme 4 : Forcer la saisie d'un nombre positif

Variables :

nombre : flottant

début

| *nombre* \leftarrow -1

| **tant que** *nombre* < 0 **faire**

| | Afficher "Veuillez entrer un nombre positif :"

| | Lire *nombre*

| **fin**

fin

Boucle tant que

Remarque :

- N'importe quelle boucle **pour** peut s'écrire à l'aide d'une boucle **tant que**

Exemple avec la factorielle

Algorithme 5 : Factorielle de n ($n!$) avec tant que

Variables :

produit, n : entiers

début

Afficher " Entrez un nombre entier :"

Lire n

$produit \leftarrow 1$

$i \leftarrow 1$

tant que $i \leq n$ **faire**

$produit \leftarrow produit \times i$

$i \leftarrow i + 1$

fin

Afficher " Factorielle de " n " = " $produit$

fin

Boucle tant que

Plus généralement :

```
début
  ...
  |
  |   pour  $i$  de  $a$  à  $b$  par pas de  $c$  faire
  |   ...
  |   fin
  |
  |   ...
fin
```



```
début
  ...
  |
  |    $i \leftarrow a$ 
  |   tant que  $i \leq c$  faire
  |   |   ...
  |   |    $i \leftarrow i + c$ 
  |   fin
  |
  |   ...
fin
```

Exercices

Pour tous les exercices ci-après, vous écrirez dans un premier temps votre algorithme sur papier. Ce n'est qu'une fois que l'algorithme est écrit sur papier que vous chercherez à l'implémenter en C++.

Regardez le fichier de correspondance algo/C++ pour vous aider à traduire votre algorithme en C++.

Cette fois-ci, vous rendrez votre fichier source C++ par mail (le **main.cpp**).

Objet du mail : **TIIS1 boucles**

N'essayez de pas bâcler tous les exercices. Mieux vaut en faire un ou deux correctement que 4 bâclés. N'oubliez pas de :

- bien **indenter** votre code
- écrire des **commentaires** pour expliquer ce que vous faites
- choisir des **noms de variables explicites**

Exercices

Exercice 1 :

Écrire un algorithme qui permet d'afficher les nombres pairs entre 1 et 11. Que faut-il simplement faire pour afficher ensuite les nombres impairs ?

Exercice 2 :

Écrire l'algorithme qui permet de saisir une valeur minimum et une valeur maximum de type entier, pour obtenir ensuite l'affichage de la somme des valeurs comprises entre le minimum et le maximum.

Exercice 3 (recette de film) :

Un film a coûté 150 millions d'euros. La première semaine, le film réalise une recette de 31 millions d'euros. Chaque semaine qui passe, la recette est 20% inférieure à celle de la semaine précédente.

Donnez l'algorithme qui permet d'indiquer au bout de combien de semaines le film est bénéficiaire.

Exercice 4

Écrire l'algorithme qui permet de saisir un nombre entier et d'afficher sa table de multiplication. L'affichage devra prendre cette forme (par exemple pour 3) :

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

...

$$3 * 10 = 30$$

Forcer la saisie d'une valeur entre 1 et 9 et utiliser la boucle la plus adaptée pour l'affichage.