



Algorithmique

Algorithmes élémentaires opérants par boucle simple

Anicet E. T. Ebou, ediman.ebou@inphb.ci



Ce travail est soumis à une licence internationale Creative Commons Attribution 4.0.

Mise en situation

Version simplifiée de la facturation des produits dans un supermarché :

- Le prix unitaire est fourni
- Le nombre d'articles identiques est saisi puis on passe au produit suivant et ainsi de suite...

Quand tous les produits sont entrés, le total est calculé et affiché

Éléments algorithmiques

Répéter et avancer: On considère un article...puis on passe au produit suivant.

Éléments algorithmiques

Condition d'entrée et condition d'arrêt

On commence : il y a au moins un produit à enregistrer

On répète l'action :

- Tant qu'il y a des produits à enregistrer
- Tant que le nombre de produit à enregistrer n'est pas atteint

On arrête la répétition : Quand tous les produits sont entrés, le total est calculé et affiché

Éléments algorithmiques

Notion d'itération:

Quelles sont les actions à répéter ?

- Le prix unitaire est fourni
- Le nombre d'articles est saisi
- Le prix total pour un article est calculé

Combien de fois le bloc d'action sera-t-il exécuté ?

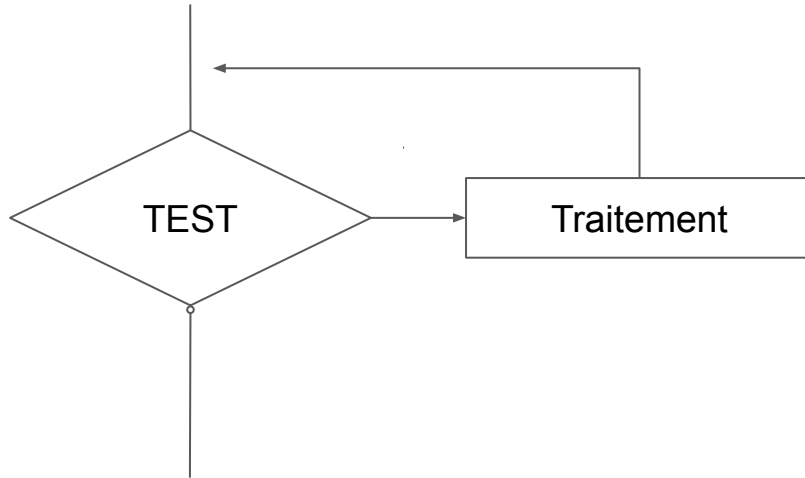
01

Boucle <<TANT QUE>>

Boucle <<TANT QUE>>

Cette structure algorithmique permet de réaliser un certain nombre d'actions tant qu'une certaine condition est remplie (vraie).

Boucle <<TANT QUE>>



Algogigramme

```
TantQue condition faire  
  
    Action1_pour_vrai  
  
    Action2_pour_vrai  
  
    ...  
  
FinTantQue
```

Pseudo-code

Boucle <<TANT QUE>>: remarques

Seule la condition TEST décide de l'exécution (ou de la ré-exécution) du bloc Traitement. FinTantQue n'indique pas de fin directe après l'exécution du bloc, sauf si la condition devient fausse directement bien sûr. Il en sera de même pour toutes les structures de type boucle.

```
Algorithme facturation
Var
    prix_unit, total, nb_art: Entier
    reponse: Caractère
Debut
    reponse ← 'o'
    total ← 0
    TantQue (reponse='o') faire
        Ecrire('prix unitaire : ')
        Lire(prix_unit)
        Ecrire("nbre d'articles: ")
        Lire(nb_art)
        total ← total + prix_unit * nb_art
        Ecrire('Ajouter un autre article ? o/n')
        Lire(reponse)
    FinTantQue
    Ecrire('Net à payer :', total)
Fin
```



Travaux Pratiques

Représentez l'algorithme précédent (slide 10) à l'aide d'un algorithme.

02

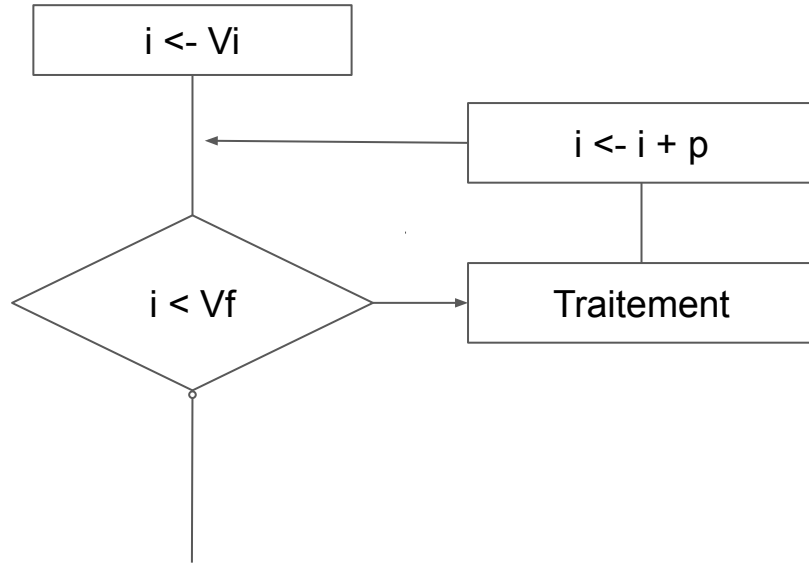
Boucle <<POUR>>

Boucle <<POUR>>

Cette structure algorithmique permet de réaliser un certain nombre d'actions en s'appuyant sur un compteur géré automatiquement.

La boucle POUR est surtout adaptée à la gestion des situations où le nombre de répétitions est connu ou facile à établir.

Boucle <<POUR>>



Allogigramme

Vi : Valeur initiale
Vf: Valeur finale
p: pas

Pour compteur \leftarrow Vi **à** Vf **faire**

Action1_pour_vrai

Action2_pour_vrai

...

FinPour

Pseudo-code

Boucle <<POUR>>: remarques

- Le lettre i représente la variable d'itération de la boucle. On l'utilise sans doute parce que c'est la première lettre des mots index et indice, mais aussi parce qu'elle est souvent utilisée comme variable muette quand on étudie les suites en mathématiques (avec j et k).
- La variable d'itération va prendre des valeurs différentes dans chacun des tours de la boucle.

Boucle <<POUR>>: remarques

- La lettre p désigne le pas de la boucle. C'est la valeur des sauts que fait la variable d'itération. Si on ne l'indique pas, il faut 1 par défaut (c'est pratique car on veut le plus souvent travailler avec des entiers consécutifs).
- La boucle POUR peut toujours être mise en place par une boucle tant que.

Boucle <<POUR>>: remarques

- Les boucles de type pour ne s'arrêteront au FinPour que si le bon nombre d'itérations a été effectué. Le mot « FinPour » ne signale que la fin du bloc.

Boucle <<POUR>>

On vous demande d'écrire un algorithme permettant de calculer la moyenne arithmétique de N valeurs fournies par l'utilisateur.

Algorithme moyenne

Var

 x, moy: Réel

 i,n: Entier

Debut

 Ecrire('Entrez le nombre de valeurs :')

 Lire(n)

 moy \leftarrow 0

 Pour i \leftarrow 1 à n faire

 Ecrire('Entrez la valeur', i)

 Lire(x)

 moy \leftarrow moy+(x/n)

 FinPour

 Ecrire('La moyenne est : ', moy)

Fin



Travaux Pratiques

Représentez l'algorithme précédent (slide 19) à l'aide d'un algogigramme.