



CHAPITRE 1 NOTIONS D'ALGORITHMES

1 QU'EST CE QU'UN ALGORITHME ?

Le terme algorithme est employé en informatique pour décrire une méthode de résolution de problème programmable sur machine.

Un algorithme est une suite finie et ordonnée d'opérations (actions) élémentaires finies (en temps et moyens). Elle est régie par un ensemble de règles ou d'instructions de contrôle (séquencement, sélection et itération) permettant d'aboutir à un résultat déterminé d'un problème donné.

Exemple 1:

Creuser un trou, reboucher un trou et placer un arbre dans un trou sont des opérations élémentaires (des actions simples) que toute personne (machine dans le cas de l'informatique) est censée savoir exécuter. Néanmoins, si un jardinier (programmeur) veut faire planter un arbre par une personne qui ne sait pas le faire, il doit lui fournir "un descriptif" (un algorithme) qui lui indique les opérations à faire ainsi que leur ordre d'exécution (séquencement).

Algorithme de plantation d'un arbre

- 1- Creuser un trou.
- 2- Placer l'arbre dans le trou.
- 3- Reboucher le trou.

Exemple 2:

Pour planter et arroser un ensemble d'arbres, on peut procéder de la manière suivante: planter l'ensemble d'arbres et les arroser tous à la fin.

Algorithme de plantation et d'arrosage de plusieurs arbres

- 1- Creuser un trou.
- 2- Placer un arbre dans le trou.
- 3- Reboucher le trou.
- 4- S'il existe encore des arbres exécuter les actions 1, 2, 3 et 4.
Sinon exécuter les actions suivantes.
- 5- Arroser les arbres.

L'algorithme correspondant indique un ensemble d'opérations (1, 2, 3 et 4) à répéter un certain nombre de fois (règle d'itération) et l'opération ou les opérations (5 ou 1, 2, 3 et 4) à exécuter selon qu'il reste ou non des arbres à planter (règle de sélection).

Remarque:

Pour résoudre le problème précédent, on peut procéder autrement : planter et arroser arbre par arbre. On conclut alors qu'à un problème donné pourraient correspondre plusieurs algorithmes.

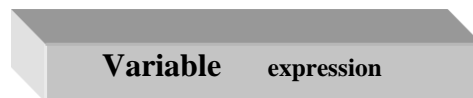


2 OPERATIONS DE BASE

Ce paragraphe décrit la liste des opérations de base pouvant composer un algorithme. Elles sont décrites en pseudocode (pseudolangage). Il s'agit d'un langage informel proche du langage naturel et indépendant de tout langage de programmation.

Les données manipulées dans un algorithme sont appelées des variables.

2.1 L'AFFECTION



L'affectation, notée par le symbole \leftarrow , est l'opération qui évalue une expression (constante ou une expression arithmétique ou logique) et attribue la valeur obtenue à une variable.

Exemples d'affectation

$a \leftarrow 10$	a reçoit la constante 10
$a \leftarrow (a*b)+c$	a reçoit le résultat de $(a*b)+c$
$d \leftarrow 'm'$	d reçoit la lettre m

2.2 LA LECTURE



Cette opération permet d'attribuer à une variable une valeur introduite au moyen d'un organe d'entrée (généralement le clavier).

Exemples de lecture

Lire a	On demande à l'utilisateur d'introduire une valeur pour a
Lire (a,b,c)	On demande à l'utilisateur d'introduire 3 valeurs pour a, b et c respectivement

Remarque :

L'organe d'entrée est assimilé à un ruban composé d'une suite de cases chacune peut contenir un caractère ou un chiffre.

C'est la valeur qui se trouve à la tête du ruban qui sera attribuée à la variable à lire. Une fois, la lecture est terminée, elle est supprimée du ruban.

2.3 L'ECRITURE



Elle communique une valeur donnée ou un résultat d'une expression à l'organe de sortie.

Exemples d'écriture

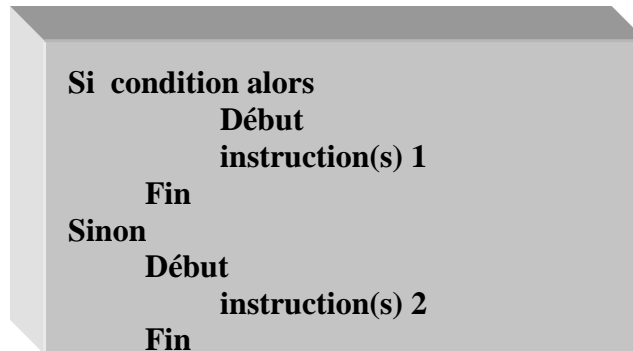
Ecrire 'bonjour'	Affiche le message bonjour (constante)
Ecrire 12	Affiche la valeur 12
Ecrire a,b,c	Affiche les valeurs de a, b et c
Ecrire a+b	Affiche la valeur de a+b



2.4 INSTRUCTIONS DE CONTROLE

2.4.1 INSTRUCTIONS SELECTIVES

a) Instruction Si



Elle indique le traitement à faire selon qu'une condition (expression logique) donnée est satisfaite ou non.

Il est possible d'omettre début et fin si le bloc d'instructions à exécuter est constitué d'une seule instruction.

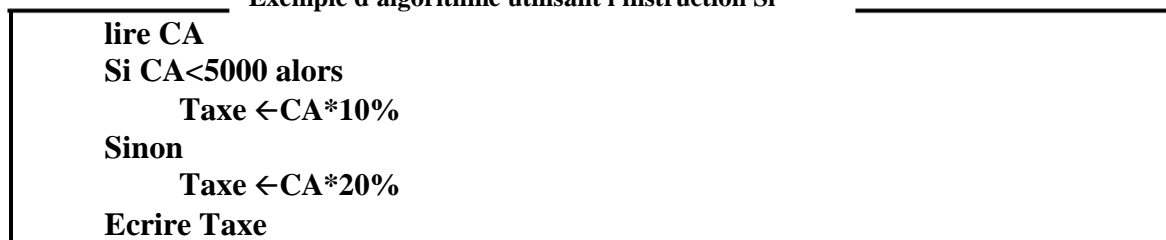
Exemple :

Calculer la taxe sur le chiffre d'affaire (CA) sachant qu'elle est de :

10% si le CA < 5000DH

20% si le CA ≥ 5000DH

Exemple d'algorithme utilisant l'instruction Si



Remarques:

1- Une instruction de contrôle peut se réduire à :

```

Si condition alors
    Début
    instruction(s)
    Fin
    
```

2- On peut avoir plusieurs si imbriqués comme:

```

Si condition1 alors
    Si condition2 alors
        Si condition3 alors
            <instruction(s) 1>
        Sinon
            <instruction(s) 2>
    Sinon
        <instruction(s) 3>
Sinon
    <instruction(s) 4>
    
```

Exemple:



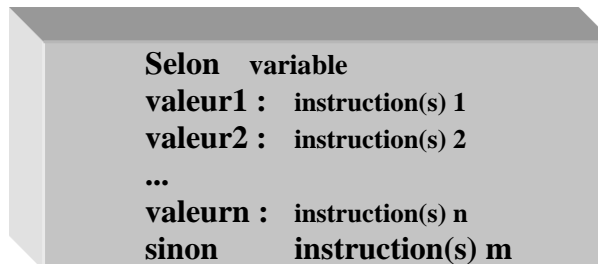
Vérifier si un agent a droit de bénéficier du crédit de logement ou non selon ses années d'ancienneté et sa note d'appréciation.

Exemple d'algorithme utilisant des si imbriqués

```

lire (ancienneté, note)
Si ancienneté < 5 alors
    Si ancienneté=4 et note≥16 alors
        écrire ('L'agent a droit de bénéficier du crédit')
    Sinon
        Si ancienneté=3 et note≥18 alors
            écrire ('L'agent a droit de bénéficier du crédit')
        Sinon
            écrire ('L'agent n'a pas droit de bénéficier du crédit')
Sinon
    Si note≥13 alors
        écrire ('L'agent a droit de bénéficier du crédit')
    Sinon
        écrire ('L'agent n'a pas droit de bénéficier du crédit')
    
```

b) Instruction Selon



Elle indique le traitement à faire selon la valeur d'une variable.

Exemple:

Vérifier et Afficher si un caractère saisi est une voyelle ou consonne.

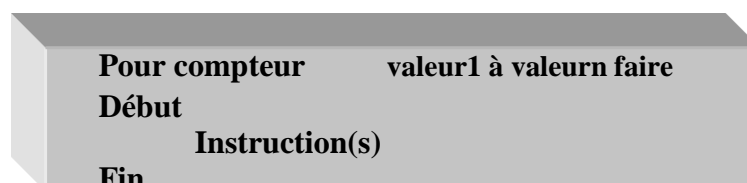
Exemple d'algorithme utilisant l'instruction Selon

```

lire c
Selon c
    'a' : écrire ('le caractère est une voyelle')
    'e' : écrire ('le caractère est une voyelle')
    'i' : écrire ('le caractère est une voyelle')
    'o' : écrire ('le caractère est une voyelle')
    'u' : écrire ('le caractère est une voyelle')
    'y' : écrire ('le caractère est une voyelle')
sinon écrire ('le caractère est une consonne')
    
```

2.4.2 INSTRUCTIONS ITERATIVES

a) Instruction Pour





Elle permet de répéter un traitement un nombre de fois précis et connu en utilisant un compteur (variable à incrémenter d'une itération à l'autre).

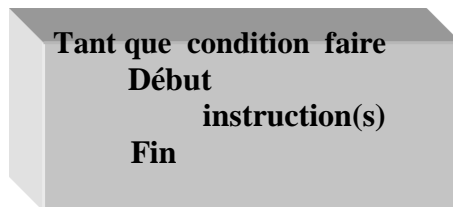
Exemple :

Afficher la somme des entiers compris entre 0 et une valeur n saisie au clavier ($n \geq 0$).

```

Exemple d'algorithme utilisant l'instruction Pour
lire n
s ← 0
Pour i 1 à n faire
    Début
        s ← s+i
    Fin
écrire s
    
```

b) Instruction Tant que



Elle permet de répéter un traitement tant qu'une condition est satisfaite.

Exemple :

Calculer la somme s des entiers compris entre 0 et un nombre n saisie au clavier (on suppose que $n \geq 0$).

```

Exemple d'algorithme utilisant l'instruction Tant que
lire n
s ← 0
Tant que n > 0 faire
    Début
        s ← s+n
        n ← n-1
    Fin
Ecrire s
    
```

Remarque :

Dans le cas d'utilisation de l'instruction tant que, si la condition est fausse au départ, le bloc d'instructions ne sera pas du tout exécuté (Dans l'exemple ci-dessus, c'est le cas où $n \leq 0$ au départ).

c) Instruction Faire Tant que Faire



Elle permet de répéter un traitement tant qu'une condition est satisfaite.



Exemple :

Calculer la somme s des entiers compris entre 0 et un nombre n saisi au clavier.

Exemple d'algorithme utilisant l'instruction Faire Tant que

```
lire n
s ← 0
i ← 0
Faire
    Début
        s ← s+i
        i ← i+1
    Fin
Tant que i ≤ n
Ecrire s
```

Remarque :

Dans le cas d'utilisation de l'instruction Faire Tant que, le bloc d'instructions est exécuté au moins une fois quelle que soit la condition.

Reprenons l'exemple précédent sous une autre forme (On initialise cette fois-ci i à 1) :

Exemple d'algorithme utilisant l'instruction Faire Tant que

```
lire n
s ← 0
i ← 1
Faire
    Début
        s ← s+i
        i ← i+1
    Fin
Tant que i ≤ n
Ecrire s
```