

Eléments d'algorithmique

Définition : On peut définir un algorithme comme une suite d'instructions précises, à appliquer dans un ordre précis, pour arriver en un nombre fini d'étapes à un certain résultat. L'algorithmique est la science des algorithmes (création, amélioration...)

1) Affecter une variable

Il s'agit de donner une valeur de type caractère, nombre ou booléen, liste ou tableau, ... (ou assigner) à un espace de la mémoire caractérisé par un nom.

Syntaxe : `NomDeVariable ← Expression calculée`

Exemples : `Toto ← 4`
 donne la valeur 4 à Toto
`Toto ← Titi`
 recopie la valeur de Titi dans Toto
`Toto ← Titi + 5`
 Toto prend la valeur de Titi, augmentée de 5
`Toto ← Toto - 1`
 diminue de 1 la valeur de Toto

Variables indicées (listes ou tableaux) : on peut, en cas de besoin, ranger des valeurs dans une variable « multiple », portant un nom unique mais dont chaque emplacement est caractérisé par un numéro appelé indice.

Exemple : `Tata[5] ← 4`
 range la valeur 4 dans l'emplacement d'indice 5 du tableau Tata.

Remarque : les indices des tableaux commencent à zéro dans la plupart des logiciels de programmation, c'est le cas de Python.

2) Echanger des données avec l'utilisateur

Lorsque le programmeur souhaite que la machine affiche une information à l'écran, il pourra employer l'instruction Ecrire.

Syntaxe : `Ecrire expression` ou `Ecrire(expression)` ou `Afficher ...`

Exemples : `Ecrire(45)`
 affiche à l'écran 45
`Ecrire("Bonjour")`
 affiche à l'écran le texte « Bonjour »
`Ecrire(Toto)`
 affiche à l'écran la valeur de la variable Toto

Inversement, lorsque l'on souhaite que l'algorithme (ou la machine) demande à l'utilisateur une valeur qui sera entrée au clavier, on emploiera l'instruction Lire.

Syntaxe : `Lire NomDeVariable` ou `Lire(NomDeVariable)` ou `Saisir ...`

Exemple : `Lire(Toto)` interrompt l'exécution, attend qu'une valeur soit frappée au clavier et range cette valeur dans la variable Toto

3) Effectuer des tests

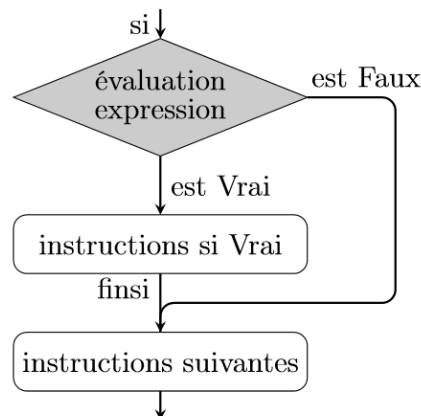
Les algorithmes permettent d'évaluer une condition (vraie ou fausse) et d'effectuer, selon le résultat, telle ou telle série d'actions.

Syntaxe :
Si condition Alors
 Série d'actions 1
SinonSi condition Alors
 Série d'actions 2
etc...
Sinon
 Série d'actions $n + 1$
FinSi

Seules les lignes Si... et FinSi sont obligatoires. Les instructions SinonSi et Sinon ne sont employées qu'en cas de besoin.

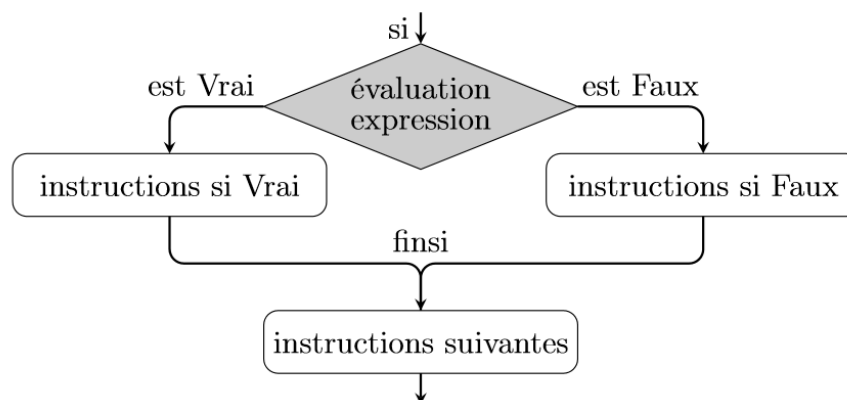
Exemple 1, une conditionnelle **sans** alternative :

```
Si Toto = 0 Alors  
    Ecrire("Toto vaut zéro")  
FinSi
```



Exemple 2, une conditionnelle **avec** alternative

```
Si Toto > Tata Alors  
    Ecrire("Toto est le plus grand")  
Sinon  
    Ecrire("Tata est le plus grand")  
FinSi
```



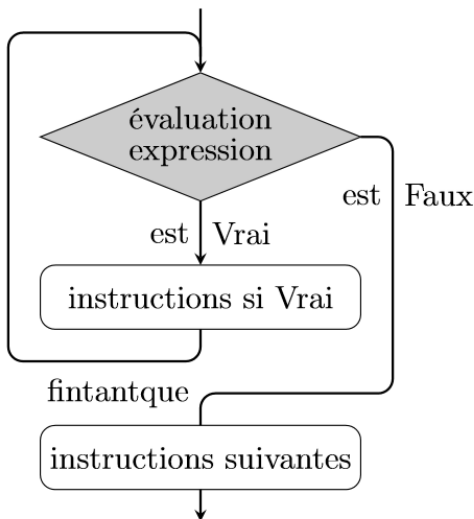
Exemple :

Ecrire un algorithme qui, après avoir demandé les réels a et b à l'utilisateur, résout et affiche les éventuelles solutions de l'équation de la variable $x : ax + b = 0$

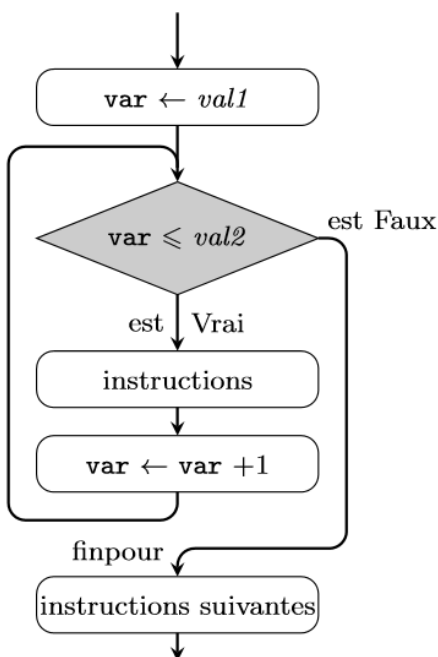
4) Effectuer des boucles

Il s'agit de spécifier à l'ordinateur de recommencer une série d'actions, soit jusqu'à ce qu'un booléen devienne **FAUX** (syntaxe A) soit jusqu'à ce qu'un compteur ait atteint une valeur spécifique (syntaxe B).

Syntaxe A : TantQue booléen
 Série d'actions
 FinTantQue



Syntaxe B : Pour NomdeVariable ← ValeurInitiale à ValeurFinale
 Série d'actions
 FinPour



Exemple :

Utiliser les deux formes de boucles pour, après avoir demandé à l'utilisateur un entier n , calcule puis affiche la somme des entiers allant de 1 à n .

Comparaison des boucles pour et tant que

Implicite, l'instruction *pour* :

- initialise un compteur
- incrémente le compteur à chaque pas
- vérifie que le compteur ne dépasse pas la borne supérieure

Explicite, l'instruction *tant que* doit

- initialiser un compteur (*amorçage*)
- incrémenter le compteur à chaque pas (*relance*)
- vérifier que le compteur ne dépasse pas la borne supérieure (*test de boucle*)

5) Les problèmes fondamentaux en algorithmique

La complexité

- En combien de temps un algorithme va-t-il atteindre le résultat escompté ?
- De quel espace a-t-il besoin ?

La calculabilité

- Existe-t-il des tâches pour lesquelles il n'existe aucun algorithme ?
- Etant donnée une tâche, peut-on dire s'il existe un algorithme qui la résolve ?

La correction

- Peut-on être sûr qu'un algorithme réponde au problème pour lequel il a été conçu ?