

ALGORITHMIQUE

1. Rappels

1.1 Notion d'algorithme

Un algorithme est une suite d'instructions détaillées qui, si elles sont correctement exécutées, conduit à un résultat donné.

Exemple :

La suite d'instructions suivante :

1. Choisir un nombre entier.
2. Le multiplier par lui-même.
3. Énoncer le résultat obtenu

est un algorithme permettant d'obtenir le carré d'un nombre entier ;

1.2 Pseudo-code

Les instructions doivent être formulées dans un langage compréhensible par l'exécutant. Dans le cas d'un humain, il s'agira du langage courant (langue maternelle) ; dans le cas d'une machine, il faudra recourir à un langage de programmation (Assembleur, Basic, C, Java, php ...).

En algorithmique, nous utiliserons un langage situé à mi-chemin entre le langage courant et un langage de programmation appelé pseudo-code

Il n'y a pas de norme concernant ce pseudo-code qui peut varier légèrement d'un enseignant à l'autre. Le but est surtout de mettre l'accent sur la logique de l'algorithme. L'avantage du pseudo-code est qu'il permet de rester proche d'un langage informatique sans qu'il soit nécessaire de connaître toutes les règles et spécificités d'un langage particulier

Exemple :

Dans notre pseudo-code, nous déclarerons les variables de la façon suivante :

- x : nombre
- y : texte
- a, b, c : entiers.

1.3 Les variables

En informatique, les variables sont des emplacements réservés dans la mémoire de l'ordinateur auxquels on attribue une étiquette.

Déclarer une variable c'est indiquer le nom et le type (nombre texte, tableau,...) d'une variable que l'on utilisera dans l'algorithme.

La déclaration des variables se fait au début de l'algorithme avant la première instruction

1.4 Les instructions d'entrée-sortie

Faire effectuer un calcul à une machine c'est bien... Mais il faut au moins être capable d'entrer des valeurs et il faut aussi que la machine puisse afficher un résultat !

Les instructions qui permettent de « dialoguer » avec une machine s'appellent les instructions « d'entrée/sortie » ou de « lecture/écriture »

1.4.1 Lecture

Dans notre pseudo-code nous utiliserons l'instruction lire (ou *entrer*, ou *saisir*, etc.) suivie du nom d'une variable pour pouvoir saisir une valeur (en anglais cette instruction se nomme généralement *input*).

Lorsqu'elle rencontre une telle instruction, la machine s'arrête et attend que l'utilisateur entre une nouvelle valeur. Une fois la valeur saisie, la machine affecte la valeur saisie à la variable dont le nom figure après lire. Ensuite elle passe à l'instruction suivante .

Exemple :

```
Variable x : entier
début algorithme
  lire x
  y prend la valeur 2*x
fin algorithme
```

Cet algorithme demande d'entrer un nombre entier, stocke la valeur de ce nombre dans la variable x, puis calcule le double du nombre entré et affecte ce double à la variable y.

Le résultat n'est pas affiché (d'où le paragraphe suivant...)

1.4.2 Écriture

Dans notre pseudo-code nous utiliserons l'instruction afficher suivie du nom d'une variable ou d'une constante (nombre, texte ...) pour afficher une valeur (on peut également utiliser écrire ou print en anglais).

Pour afficher un texte on utilise des guillemets (simples ou doubles) :...

```
afficher 'Ce texte sera affiché'
```

...Il est fréquent d'afficher un texte pour donner des consignes ou des informations à l'utilisateur.

Pour afficher le contenu d'une variable on fait suivre « *afficher* » du nom de la variable sans guillemet :...

```
afficher
```

Exemple :

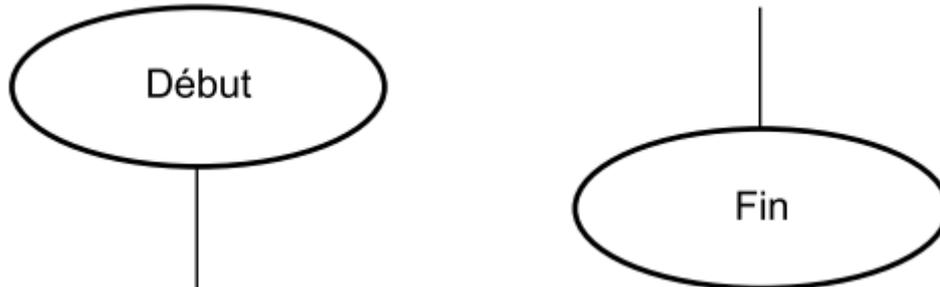
```
Variables
x, y, z : entiers
début
x prend la valeur 5
y prend la valeur x
z prend la valeur x+y+1
afficher z
fin de l'algorithme
```

2. Organigramme

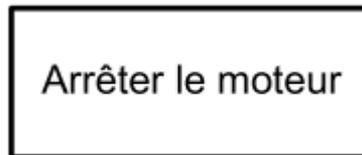
2.1 Définition

L'organigramme est constitué de figures qui ont une signification, complété par du texte. Ces figures suivent un ordre précis.

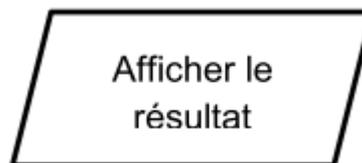
- Chaque programme commence par un début et se termine par une fin. Il y a donc des figures pour symboliser le début et la fin.



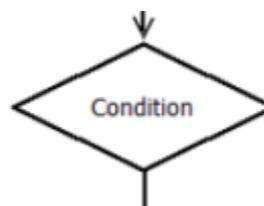
- Des flèches indiquent le sens du programme.
- Une instruction est symbolisé par un rectangle avec l'instruction inscrit à l'intérieure.



- Les entrées/sorties, variables sont symbolisés par un parallélogramme.



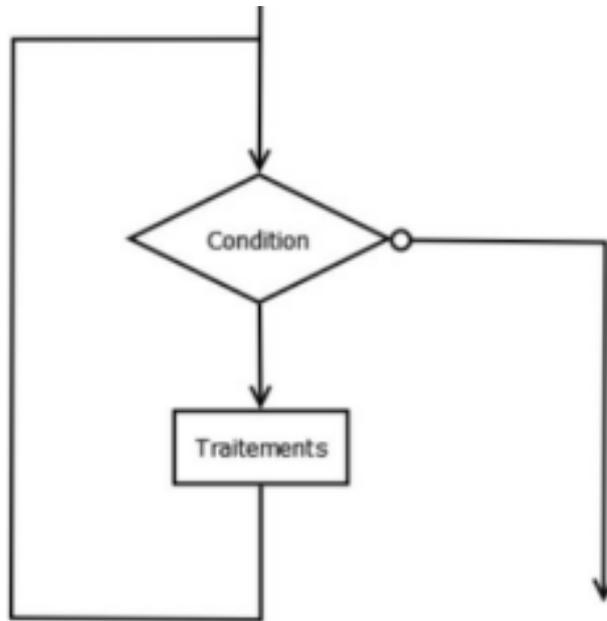
- Pour symboliser une condition simple, on utilise un losange.



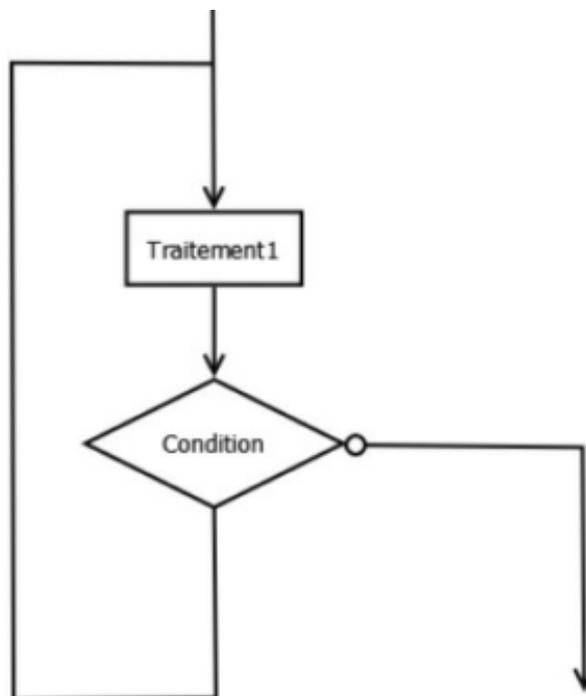
2.2 Les boucles

Il en existe deux sortes :

« La boucle tant que faire » est symbolisé dans les organigrammes de la manière suivante :



« La boucle répéter jusqu'à » permet de répéter une tâche tant qu'une condition n'est pas favorable. Elle est représentée dans un organigramme de la manière suivante :



3. Organigramme de la résolution de $a x^2 + b x + c = 0$

