

# Chapitre 5 : Les éléments de base d'un algorithme

Omar El Kharki & Jamila Mechbough

Site Web d'Apprentissage des NTIC : <http://www.elkharki.africa-web.org/>

## I. Introduction

### I.1. Notion d'algorithme

Dans la vie courante, un algorithme peut prendre la forme :

- d'une recette de cuisine;
- d'un itinéraire routier
- d'un mode d'emploi, etc.

Une recette de cuisine, par exemple, est un algorithme : à partir des ingrédients, elle explique comment parvenir au plat. De même, un itinéraire routier explique comment, à partir d'une position initiale, rejoindre une position finale en un certain nombre d'étapes. Etc.

#### **Exemple : Préparer la pâte à tarte**

##### ➤ **Les ingrédients :**

250 g de farine ;

50 g de beurre ;

Un verre de lait.

## ➤ Les actions élémentaires à réaliser

Début

Incorporer le beurre dans la farine

Pétrir le mélange jusqu'à ce qu'il soit homogène

Ajouter du lait

Mélanger

Si la pâte est trop sèche, alors ajouter du lait

Si la pâte à une bonne consistance, alors la laisser reposer

½ heures

Faire cuire la pâte

Fin

Un algorithme sert à transmettre un savoir faire. Il décrit les étapes à suivre pour réaliser un travail. Tout comme le savoir-faire du cuisinier se transmet sous la forme d'une recette, celui d'un informaticien se transmet sous la forme d'un algorithme.

## I.2. Définitions

Le mot « algorithme » provient de la forme latine (Algorismus) du nom du mathématicien arabe AL KHWARIZMI. Il tenta une première définition : « Un algorithme est une séquence d'opérations visant à la résolution d'un problème en un temps fini. »

Un algorithme est une suite d'action élémentaire qu'il faut effectuer pour résoudre un problème.

## I.3. Algorithmique et programmation

Tout problème à programmer doit être résolu, d'abord sous forme d'algorithme, puis converti en programme dans le langage de votre choix. En effet, un algorithme est indépendant du langage de programmation à utiliser.

Un programme est un enchaînement d'instruction, écrit dans un langage de programmation, exécutée par un ordinateur, permettant de traiter un problème et de renvoyer des résultats. Il représente la traduction d'un algorithme à l'aide d'un langage de programmation.

Le cycle de développement d'un programme (ou d'une application) informatique peut se résumer ainsi (figure1):

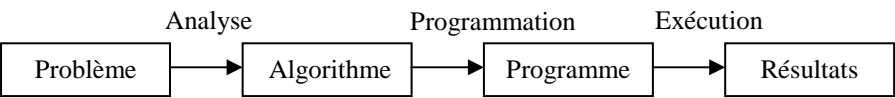


Figure 1 : Cycle de développement d'un programme

## II. Structure général d’un algorithme

Entête	{	<b>Algorithme</b>	NomAlgorithme
Déclaration	{	<b>Variable</b>	Identificateur : type
		<b>Constante</b>	Identificateur = valeur
Corps de l’algorithme	{	Début	
		Instruction 1	
		Instruction 2	
		.....	
		Instruction n	
		Fin	

### II.1. Notion de variable

Les données ainsi que les résultats des calculs intermédiaires ou finaux, sont rangés dans des *cases-mémoires* appelées **variables**.

Une variable est un emplacement mémoire nommé, de taille fixée prenant au cours du déroulement de l'algorithme un nombre indéfini de valeurs différentes.

### II.2. Déclaration des variables

La partie déclarations consiste à énumérer toutes les variables dont on aura besoin au cours de l'algorithme.

Chaque déclaration doit comporter le nom de la variable (identificateur) et son type.

## Syntaxe :

Variable identificateur : type

## Exemple :

Variable surface : réel

### ➤ Identificateur

Un identificateur est le nom donné à une variable.

Ce nom doit obligatoirement **commencer** par **une lettre** suivie d'une suite de lettre et de chiffre et il **ne doit pas contenir de caractère accentuer ni d'espace**.

### ➤ Types de données

Le type d'une variable est l'ensemble des valeurs qu'elle peut prendre. Par exemple, une variable de type booléen peut prendre les valeurs **Vrai** ou **Faux**.

Les différents types utilisés :

- **Type Entier** : pour manipuler les nombres entiers positifs ou négatifs. Par exemple : 5, -15, etc.
- **Type Réel** : pour manipuler les nombres à virgule. Par exemple : 3.14, -15.5, etc.
- **Type Caractère** : pour manipuler des caractères alphabétiques et numériques. Par exemple : 'a', 'A', 'z', ' ', '1', '2' etc.
- **Type Chaîne** : pour manipuler des chaînes de caractères permettant de représenter des mots ou des phrases. Par exemple : "bonjour, Monsieur"
- **Type Booléen** : pour les expressions logiques. Il n'y a que deux valeurs booléennes : vrai et faux.

## Exemple :

Variables n : entier

r : réel

a, b: booléens

nom\_etudiant : chaîne

A un type donné, correspond un ensemble d'opérations définies pour ce type :

Type	Opérations possibles	Symbole ou mot clé correspondant
<b>Entier</b>	Addition Soustraction Multiplication Division Division Entière Modulo Exposant (puissance) Comparaisons	+ - * / DIV MOD ^ <, =, >, <=, >=, <>
<b>Réel</b>	Addition Soustraction Multiplication Division Exposant Comparaisons	+ - * / ^ <, =, >, <=, >=, <>
<b>Caractère</b>	Comparaisons	<, =, >, <=, >=, <>
<b>Chaîne</b>	Concaténation	+, &
<b>Booléen</b>	Comparaison Logiques	<, =, >, <=, >=, <> ET, OU, NON

### Exemple :

$$5 / 2 = 2.5$$

$$5 \text{ Div } 2 = 2$$

$$5 \text{ Mod } 2 = 1$$

$$5 ^ 2 = 25$$

"Bonjour" & " " & "Monsieur" donne "Bonjour Monsieur"

Les opérations définies pour le type booléen sont :

- Le **ET** logique (and)
- Le **OU** logique (Or)
- Le **NON** logique (not)
- Le **OU** exclusif (Xor)

Nous résumons dans une table de vérité les résultats obtenus suivant les valeurs de deux opérands :

P	Q	Non P	Non Q	P et Q	P ou Q	P xor Q
0	0	1	1	0	0	0

0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

### Lois de De Morgan

Non (P et Q)  $\Leftrightarrow$  (non P) ou (non Q)

Non (P ou Q)  $\Leftrightarrow$  (non P) et (non Q)

## II.3. Les constantes

Comme une variable, une constante est un emplacement mémoire réservé auquel on accède par le nom qui lui a été attribué, mais dont la valeur stocké ne sera jamais modifiée au cours du programme.

**Syntaxe :**

**Constante nom\_de\_la\_constant = valeur**

**Exemple :**

Constante pi = 3.14

## II.4. Les instructions

Une instruction est une action élémentaire commandant à la machine un calcul, ou une communication avec un de ses périphériques d'entrées ou de sorties. Les instructions de base sont :

### ➤ L'instruction d'affectation :

L'affectation permet d'affecter une valeur à une variable.

Symbolisée en algorithmique par " $\leftarrow$ ", elle précise le sens de l'affectation.

**Syntaxe :**

**Variable  $\leftarrow$  Expression**

Expression peut être soit :

- Identificateur
- Constante
- Expression arithmétique
- Expression logique

**Sémantique :**

Une affectation peut être définie en deux étapes :

- Evaluation de l'expression qui se trouve dans la partie droite de l'affectation
- Placement de cette valeur dans la variable.

**Exemple :**

0	Algorithme	Calcul
1	Variables	A, B, C, D : entier
2	Debut	
3		$A \leftarrow 10$
4		$B \leftarrow 30$
5		$C \leftarrow A+B$
6		$D \leftarrow C*A$
7	Fin	

**Note :** Les lignes sont numérotées pour faciliter l'explication.

Nous pouvons expliquer ce qui se passe par le tableau suivant :

Variable	N° de ligne					
	1	2	3	4	5	6
<b>A</b>	?	?	10	10	10	10
<b>B</b>	?	?	?	30	30	30
<b>C</b>	?	?	?	?	40	40
<b>D</b>	?	?	?	?	?	400

**Remarque :**

Les variables numériques ne sont pas forcément initialisées à zéro. Leur valeur peut être n'importe quoi. C'est pour cette raison que nous avons mis un point d'interrogation avant qu'une première valeur ne soit affectée à la variable.

**Exemple :**

0	Algorithme	Logique
1	Variables	A, B, C : Booléen
2	Debut	
3	A	← Vrai
4	B	← Faux
5	C	← A et B
6	Fin	

Variable	N° de ligne				
	1	2	3	4	5
<b>A</b>	?	?	Vrai	Vrai	Vrai
<b>B</b>	?	?	?	Faux	Faux
<b>C</b>	?	?	?	?	Faux

### ➤ L'instruction d'entrée :

L'instruction d'entrée ou de lecture permet à l'utilisateur de saisir des données au clavier pour qu'elles soient utilisées par le programme.

Cette instruction assigne (affecte) une valeur entrée au clavier dans une variable.

#### Syntaxe :

**Lire (identificateur)**

#### Exemple :

Lire (A)

Cette instruction permet à l'utilisateur de saisir une valeur au clavier qui sera affectée à la variable A.

#### Remarque :

Lorsque le programme rencontre cette instruction, **l'exécution s'interrompt** et attend que l'utilisateur tape une valeur. Cette valeur est rangée en mémoire dans la variable désignée.

### ➤ L'instruction de sortie:

L'instruction de sortie (d'écriture) permet d'afficher des informations à l'utilisateur à travers l'écran.

#### Syntaxe :



## Ecrire (expression)

Expression peut être une valeur, un résultat, un message, le contenu d'une variable...

### Exemple :

Ecrire (A)

Cette instruction permet d'afficher à l'écran la valeur de la variable A.

### Exercice :

Ecrire un algorithme qui permet de saisir le prix HT (PHT) d'un article et de calculer son prix total TTC (PTTC). TVA=20%.

### Solution :

0	Algorithme    Calcul_PTTC
1	Variables   PHT, PTTC : réel
2	Constante TVA = 0,2
3	Début
4	Ecrire ("Entrez le prix hors taxes :")
5	Lire (PHT)
6	$PTTC \leftarrow PHT + (PHT * TVA)$
7	Ecrire ("Le prix TTC est ", PTTC)
8	Fin

Explication de l'algorithme :

N° de ligne	Explication
0	Déclare un algorithme dont le nom est Calcul_PTTC
1	Déclare les différentes variables utilisées par l'algorithme
2	Déclare la constante TVA
3	Marque le début des traitements effectués par l'algorithme.
4	Affiche à l'écran le message : Entrez le prix hors taxes :
5	Permet à l'utilisateur de saisir une valeur au clavier qui sera affectée à la variable PHT.
6	Calcul le prix TTC et affecte le résultat à la variable PTTC
7	Affiche le résultat à l'écran
8	Marque la fin de l'algorithme