

Chapitre 01 : Les éléments de base d'un algorithme

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

Exemple : Préparer la pâte à tarte

➤ **Les ingrédients :**

250 g de farine ;
50 g de beurre ;
1 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 a une bonne consistance, alors la laisser reposer une demi heure

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. Définition d'un algorithme

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

Nous pouvons adopter la définition suivante : un algorithme est la description de la méthode de résolution d'un problème quelconque en utilisant des instructions élémentaires. Ces instructions deviennent compréhensibles par l'ordinateur lors de la traduction de l'algorithme en un programme.

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 utilisé.

Un programme est un enchaînement d'instructions, écrit dans un langage de programmation, exécutées 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) :

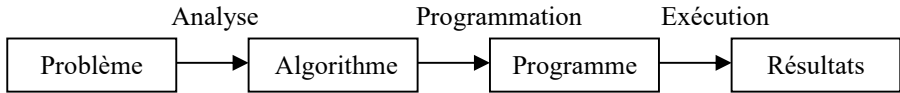


Figure 1 : Cycle de développement d'un programme

Exemple :

Parmi les langages de programmations, on peut citer : Pascal, C, C++, Visual Basic, Java, C#, J#, etc.

II. Structure générale d'un algorithme

Un algorithme est composé de trois parties principales (figure 2) :

- **l'en-tête** : cette partie sert à donner un nom à l'algorithme. Elle est précédée par le mot *Algorithme* ;
- **la partie déclarative** : dans cette partie, on déclare les différents objets que l'algorithme utilise (constantes, variables, etc.) ;
- **le corps de l'algorithme** : cette partie contient les instructions de l'algorithme. Elle est délimitée par les mots *Début* et *Fin*.

Entête	{	<i>Algorithme</i> Non
Partie déclarative	{	<i>Constante</i> Ident <i>Variable</i> Ident
	{	Début Instruction 1

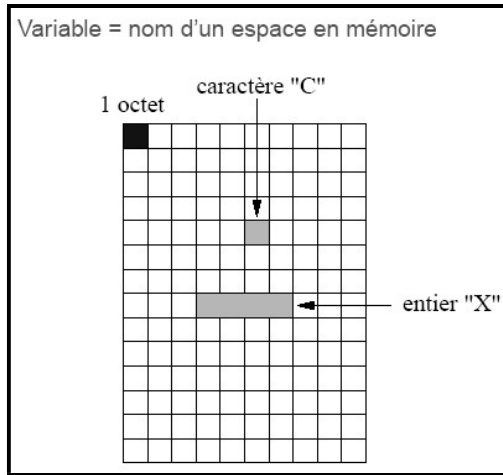
Figure 2 : Structure d'un algorithme

III. Les variables et les constantes

III.1. Notion de variable

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

Ainsi, une variable (figure suivante) est rangée dans un emplacement mémoire nommé, de taille fixe (ou non) prenant au cours du déroulement de l'algorithme, un nombre indéfini de valeurs différentes.



III.2. Déclaration des variables

La partie déclaration 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

Exemples :

Variable surface : réel

Variable a : entier

Variable a, b, c, d : entiers

Variable Nom_Prenom : chaîne

Variable absent : logique

➤ Identificateur

Un identificateur est le nom donné à une variable, une fonction, etc. Ce nom doit obligatoirement **commencer** par **une lettre** suivie d'une suite de lettres et de chiffres et il **ne doit pas contenir 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 logique (booléen) peut prendre les valeurs *Vrai* ou *Faux*.

Les différents types utilisés en algorithmique :

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

Exemple :

Variables n : entier
r : réel
a, b : logiques
nom_etudiant : chaîne

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

Type	Opérations possibles	Symbole ou mot correspondant
------	----------------------	------------------------------

Entier	Addition Soustraction Multiplication Division Division entière Modulo (le reste de la division entière) x exposant y Comparaisons	+ - * / (DIV en VB et % en C) (MOD en VB) ^ : en vb et pow(x,y) en C <, =, >, <=, >=, ≠
En algorithmique nous symbolisons la division entière par DIV et le reste de la division entière par MOD		
Réel	Addition Soustraction Multiplication Division Exposant Comparaisons	+ - * / ^ <, =, >, <=, >=, ≠
Caractère	Comparaisons	<, =, >, <=, >=, ≠
Chaîne	Concaténation Comparaison	(+, & : en VB) <, =, >, <=, >=, ≠
Booléen	Logiques	ET, OU, NON et OU _{ex}

Exemple :

$$5 / 2 = 2.5$$

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

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

$$5 \wedge 2 = 25$$

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

L'expression $5 > 2$ est Vraie.

L'expression $7 < 4$ est fausse.

Les opérations définies pour le type booléen sont :

- Le **ET** logique (and)
- Le **OU** logique (Or)
- Le **NON** logique (not)
- Le **OU_{ex}** (Ou exclusif appelée en VB Xor)

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

P	Q	Non P	Non Q	P et Q	P ou Q	P ou _{ex} 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)

III.3. Les constantes

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

Syntaxe :

Constante NOM_DE_LA_CONSTANTE = valeur

Exemple :

Constante PI = 3.14

IV. Les instructions de base

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

IV.1. L'instruction d'affectation

L'affectation permet d'affecter une valeur à une variable. Elle est symbolisée en algorithmique par " \leftarrow ".

Le signe " \leftarrow " précise le sens de l'affectation.

Syntaxe :

Variable ← Expression

Expression peut être soit :

- identificateur ;
- constante ;
- expression arithmétique ;
- expression logique.

Sémantique :

Une affectation peut être définie en deux étapes :

- évaluation 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	Début
3	A ← 10
4	B ← 30
5	C ← A+B
6	D ← 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
A	?	?	10	10	10	10
B	?	?	?	30	30	30
C	?	?	?	?	40	40
D	?	?	?	?	?	400

Remarque :

Les variables numériques ne sont pas forcément initialisées à zéro.

Leur valeur peut être n'importe quoi. C'est la raison de la présence du point d'interrogation avant qu'une première valeur ne soit affectée à la variable.

Exemple :

```

0   | Algorithmes   Logique
1   | Variables   A, B, C : Booléen
2   | Début
3   |     A ← Vrai
4   |     B ← Faux
5   |     C ← A et B
6   | Fin
    
```

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

IV.2. L'instruction d'entrée

L'instruction d'entrée ou de lecture donne la main à l'utilisateur pour saisir une donnée au clavier. La valeur saisie sera affectée à une variable.

Syntaxe :

Lire (identificateur)

Exemples :

Lire(A)
Lire(A, B, C)

L'instruction **Lire(A)** permet à l'utilisateur de saisir une valeur au clavier. Cette valeur 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.

IV.3. L'instruction de sortie

Avant de lire une variable, il est conseillé d'écrire des libellés à l'écran, afin de prévenir l'utilisateur de ce qu'il doit frapper (sinon, l'utilisateur passe son temps à se demander ce que l'ordinateur attend de lui).

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

Syntaxe :

Ecrire (expression)

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

Exemple 1 :

Ecrire (A)

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

Exemple 2 :

$A \leftarrow 2$

Ecrire ("La valeur de A est = ", A)

La dernière instruction affiche à l'écran :

La valeur de A est = 2

Exercice : Calcul PTTC

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 | Algorithmme   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 ← 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

V. Les commentaires

Lorsqu’un algorithme devient long, il est conseillé d’ajouter des lignes de commentaires dans l’algorithme, c’est-à-dire des lignes qui ont pour but de donner des indications sur les instructions effectuées et d’expliquer le fonctionnement du programme (algorithme) sans que le compilateur ne les prenne en compte.

Dans la suite de ce livre, nous utiliserons trois types de commentaires :

```
// Commentaire sur une ligne
/* Commentaire */
' Commentaire
/* Commentaire
sur
plusieurs
lignes */
```

Remarque :

Parfois on utilise les commentaires pour annuler l'action de quelques instructions dans un algorithme ou un programme au lieu de les effacer. Voir l'exemple suivant :

```
Variable i : entier
// Variable i : réel
i ← 4
/* i ← 4
i ← 4 * I */
```

Les instructions précédentes sont équivalentes à :

```
Variable i : entier
i ← 4
```

Note :

Les annexes 01 et 02 de ce livre traduisent les instructions algorithmiques vues dans ce chapitre et les chapitres qui suivent en langage de programmation Visual Basic (VB) et en langage C.