

Algorithmique et programmation

Chapitre 1 :

Les éléments de base d'un algorithme

Introduction:

- **Dans la vie courante, un algorithme peut prendre la forme d'une recette de cuisine ou bien une résolution d'exercice. Une recette de cuisine par exemple, est un algorithme, à partir des ingrédients, elle explique comment parvenir au plat, de même, une résolution d'exercice explique comment à partir des données, on obtient d'une solution finale en un certain nombre d'étapes.**

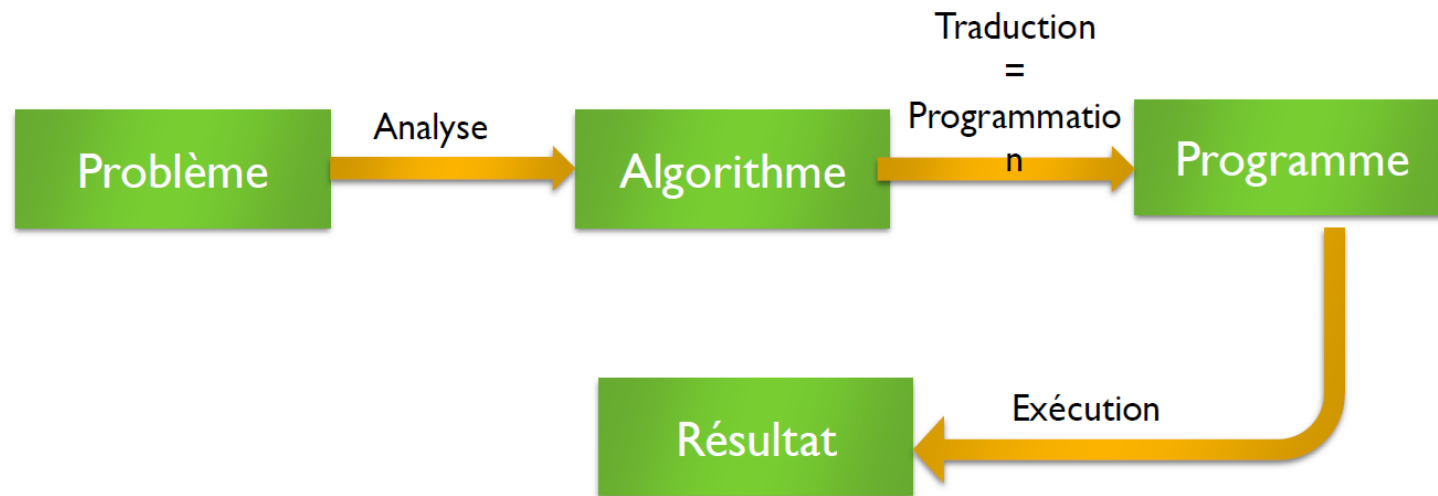
1. Qu'est-ce qu'un algorithme?

- **Un algorithme est une suite d'instructions élémentaires devant être exécutées dans un ordre déterminé pour obtenir un résultat ou résoudre un problème.**
- **Un Algorithmique est une suite des étapes à suivre pour réaliser un travail.**

Algorithme 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 une suite d'instructions, écrit dans un langage de programmation, exécutées par 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 comme ça :

2. Schéma de résolution d'un problème Informatique



- Parmi les langages de programmations, on peut citer: Pascal, C, C++, JAVA, Python.....

3) Structure générale d'un algorithme

Un algorithme est composé de trois principale:

- **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 (Variables, constantes, etc).
- **Le corps de l'algorithme:** cette partie contient les instructions de l'algorithme. Elle est délimitée par les mots **début** et **Fin**.

Entête	Algorithme Nom_Algorithme
partie déclarative	Variable Nom: type Constante Nom=valeur
corps de l'algorithme	Début Instruction 1 Instruction 2 Instruction n Fin

4)Les variables et les constantes

4-1) Notion de variable

Une variable est une zone mémoire dans laquelle on peut mémoriser de façon temporaire une valeur pour une exploitation ultérieure. Son contenu peut varier au cours de l'exécution de l'algorithme.

4-2) *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 variable (identificateur) et son type.

Syntaxe: **Variable** **identificateur:** **type**

- **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 les chiffres et il ne doit pas contenir d'espace.

Types de donnée :

- **Type entier:** sert à manipuler les nombres entiers positifs ou négatifs. Par exemple: 5, 20, -12
- **Type réel:** sert à manipuler les nombres à virgule. Par exemple: 5, 2.1,-1.2...
- **Type caractère** sert à manipuler des caractères alphabétiques et numériques. Par exemple: "a","B " , "6"...
- **Type chaîne:** sert à manipuler des chaînes de caractères permettant de représenter des mots ou des phrases comme: "bonjour", "Cours_5"...
- **Type booléen** utilise les expressions logiques. Il n'y a que deux valeurs booléennes: **Vrai** et **faux**

Exemple:

Variables a, b: entiers
c: réel
nom: chaîne_caractères
absent: booléen

Les opérations sur des variables:

Type	Opérations	Symboles	Exemples
Entier	Addition	+	$2 + 5 = 7$
	Soustraction	-	$8 - 5 = 3$
	Multiplication	*	$2 * 4 = 8$
	Division	/	$10 / 4 = 2,5$
	Division entier	Div	$10 \text{ Div } 4 = 2$
	Modulo (le reste de la division entier)	Mod	$10 \text{ Mod } 3 = 1$
	Comparaison	$\leq \geq > < = \neq$	$7 > 2$ vrai
Réel	Addition	+	$2 + 5,4 = 7,4$
	Soustraction	-	$8,5 - 5 = 3,5$
	Multiplication	*	$2 * 4 = 8$
	Division	/	$10,6 / 4 = 2,65$
	Comparaison	$\leq \geq > < = \neq$	$7 < 2$ faux
Caractère	Comparaison	$\leq \geq > < = \neq$	'B' > 'K' faux
Chaine	Concaténation	& +	"ok " & " " & "by"
	Comparaison	$\leq \geq > < = \neq$	'moh' > 'an' vrai
Booléen	Logiques	et non ou	non(7>1) faux

4-3) Les constantes

Comme une variable, il existe une constante correspond à 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_Constante = valeur

Exemple:

Constante Pi = 3.14

5) 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 bases ont:

5-1) L'instruction d'affectation:

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

Syntaxe: Variable \leftarrow Expression

Exemple: Var \leftarrow 8 ;

Var1 \leftarrow Var2 ;

A \leftarrow B +2

Algorithme Test1

Variables a,b,c: booléens

Debut

a ← vrai

b ← faux

c ← b

fin

c contient la valeur faux

Algorithme Test2

Variables a,b,c: entiers

Debut

a ← 10

b ← 30

c ← a+b

b ← a+2

a ← a*b

fin

a contient 120

b contient 12

c contient 40

5-2) 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)** ou **Lire (A, B, C)**
- L'instruction **Lire(A)** permet à l'utilisateur de saisir une valeur à clavier. Cette valeur sera affectée à la variable A.

5-3) L'instruction de sortie:

- Avant de lire une variable, il est conseillé d'écrire un message à l'écran, afin de prévenir l'utilisateur de ce qu'il doit taper.

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

- **Syntaxe: Ecrire** (expression)
- **Exemple: Ecrire** ("Donner votre âge: ")
- **Ecrire (A)** cette instruction permet d'afficher à l'écran la valeur de variable A.

5-3) *Les commentaires:*

- Lorsqu'un algorithme devint 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 d'algorithme (programme) sans que le compilateur ne les prenne en compte. On va voir deux types de commentaires

// Commentaire sur une ligne

/*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 comme dans cet exemple:
- Variable i: entier
- // Variable j: réel