

Algorithmique et Structures de Données 1

CH1: Introduction à l'algorithmique

ENSEIGNANT: FETHI MGUIS
Sections: LGLSI1/LIRIS1

A.U: 2023/2024

1 Introduction

Ecrire un programme, ce n'est pas seulement connaître le langage de dialogue entre l'ordinateur et vous. Un programme est un discours adressé à l'ordinateur, et comme tout discours, il présente deux aspects : un contenu et une forme, un sens et une grammaire. La tâche la plus importante dans l'informatisation d'un problème réel est la phase de transformation des objets réels en objets logiques. Cette transformation nécessite trois étapes :

1. La définition exacte du problème,
2. La formalisation du problème,
3. La programmation.

2 Démarche de la programmation

2.1 Définition du problème

Pour résoudre et automatiser un problème à l'aide d'un ordinateur, il est nécessaire de le définir et de préciser les données qu'on dispose et les objectifs qu'on souhaite atteindre.

D'une manière générale, la définition d'un problème consiste à prévoir des réponses à tous les cas envisageables.

2.2 La formalisation du problème

- C'est la phase algorithmique.
- Elle fournit la méthode et la démarche que l'ordinateur va suivre pour résoudre le problème posé.
- La phase la plus difficile de la résolution d'un problème n'est pas la traduction de l'algorithme dans n'importe quel langage, mais c'est la conception de l'algorithme lui-même. Cette étape est connue sous le nom d'analyse.

Définition d'un algorithme

Un algorithme est une suite finie d'instructions indiquant de façon précise l'ordre dans lequel doit être effectué un ensemble d'opérations pour obtenir la solution d'un problème.

2.3 La programmation

C'est le passage du modèle logique qui est un algorithme au modèle directement exploitable par l'ordinateur. Pour cela, il faut traduire l'algorithme en un programme écrit dans un langage de programmation choisi par l'utilisateur.

Exemple : JAVA, PASCAL, LANGAGE C, FORTRON, BASIC.

3 Les différentes étapes du processus de la programmation

Remarques :

- La programmation est donc, l'art de concevoir et d'exprimer des algorithmes.
- Pour programmer en un langage donné il faut bien le connaître et le comprendre.

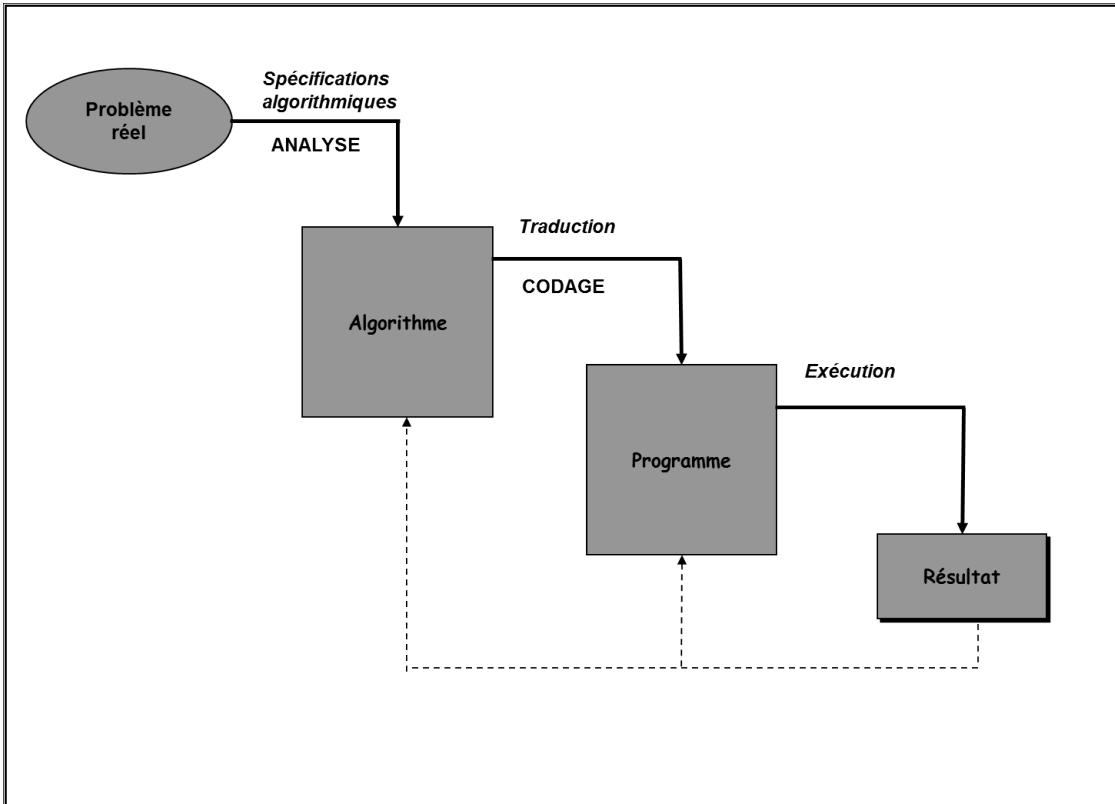


FIGURE 1 – Les différentes étapes du processus de la programmation.

4 Notion de variable

Une variable est un identificateur (objet), utilisé dans un algorithme, dont la valeur peut être modifiée à tout moment de l'algorithme.

Une variable est caractérisée par :

- **Un nom** : étiquette de la case mémoire, il est préférable qu'il soit significatif.
- **Un type** : déterminant la nature des données qu'elle peut accueillir (nombres, caractère, chaîne de caractère,...).
- **Un contenu** : ce qu'elle a comme valeur pouvant être modifiée au cours de l'exécution.

Remarques :

- Le nom d'une variable est formé d'une ou plusieurs lettres ; les chiffres sont également autorisés, à condition de ne pas les mettre au début du nom.
- A titre d'exemple : "P", "Quantité" et "g21" sont tous des identificateurs corrects. Alors que "1cycle", "calcul salaire" sont des identificateurs incorrects.

5 Notion de constante

Contrairement aux variables, les constantes sont des données dont la valeur reste fixe durant l'exécution du programme.

Exemple : pi = 3.14 N = 100

6 Notion de type

A chaque variable utilisée dans le programme, il faut associer un type qui permet de définir :

- L'ensemble des valeurs que peut prendre la variable
- L'ensemble des opérations qu'on peut appliquer sur la variable.

La syntaxe de la déclaration d'une variable est la suivante :

Variable1, Variable2, ... : Type

6.1 Type entier

- C'est l'ensemble des entiers relatifs tel que : 8, -10, +3 ;
- Les opérations définies sur les entiers sont : +, -, *, div, mod (17 div 5 = 3 ; 17 mod 5 = 2)

6.2 Type réel

- C'est l'ensemble IR
- Il existe deux formes de représentation des réels :
 1. La forme usuelle avec le point comme symbole décimal
Exemples : -4.5678 5 13.9 +12.44
 2. La notation scientifique selon le format aEb, où a est la mantisse, qui s'écrit sous la forme usuelle et b est l'exposant représentant un entier relatif
Exemples : 345 = 3.45E2 = 0.345 E3 = 3450 E-1 = ...
- Les opérations définies sur les réels sont : +, -, *, /

6.3 Type caractère

Une variable de type caractère aura pour valeur un et un seul caractère qui peut être pris parmi :

- les lettres (majuscules ou minuscules),
- les chiffres
- les caractères spéciaux inclus le caractère blanc ou espace.

Exemples : 'A', 'd', '2', '*', ''

- Les opérations définies sur les données de type caractère sont : =, <>, <, <=, >, >=

Remarque : La comparaison entre les caractères se fait selon leurs codes ASCII

Exemple : 'A' < 'B' ... < 'a' < 'b' < ... (Code ASCII de 'A' = 65 et code ASCII de 'a' = 97)

6.4 Type chaîne de caractère

Une chaîne est un ensemble de caractères.

Exemple : "Faculté des sciences de Gabes", "Algorithmique".

6.5 Type logique ou booléen

- Ce type de variable est souvent utilisé pour représenter une condition et qui ne peut prendre que l'une de ces deux valeurs : Vrai ou Faux.
- Les opérations définies sur les variables de type logique sont : la négation (NON), l'intersection (ET) et l'union (OU)

7 Les expressions

Ce sont des combinaisons entre des variables et des constantes à l'aide d'opérateurs. Elles s'expriment un calcul (expressions arithmétiques) ou une relation (expressions logiques)

7.1 Les expressions arithmétiques

Priorité	Opérateurs
1	- signe négatif (opérateur unaire)
2	() parenthèses
3	^ puissance
4	* et / multiplication et division
5	+ et - addition et soustraction

TABLE 1 – Ordre de priorité des opérateurs arithmétiques.

Exemple : $y * 54.9 / (4 + pi)$

En cas de conflit entre deux opérateurs de même priorité, on commence par celui situé le plus à gauche.

7.2 Les expressions logiques

Ce sont des combinaisons entre des variables et des constantes à l'aide d'opérateurs relationnels (=, <>, <, <=, >, >=) et/ou des combinaisons entre des variables et des constantes logiques à l'aide d'opérateurs logiques (NON, ET, OU).

Ici encore, on utilise les parenthèses et l'ordre de priorité entre les différents opérateurs pour résoudre les problèmes de conflits.

Priorité	Opérateurs
1	NON
2	ET
3	OU

TABLE 2 – Opérateurs logiques.

Priorité	Opérateurs
1	>
2	\geq
3	<
4	\leq
5	=
6	\neq

TABLE 3 – Opérateurs relationnels.

8 Les instructions simples

8.1 L'instruction d'affectation

Cette action permet de ranger une nouvelle valeur dans une variable.

Syntaxe générale :

Nom_variable \leftarrow Expression

- Nom-variable : est le nom de la variable qui reçoit la donnée.
- Expression : représente la valeur à affecter qui doit être de même type que celui de la variable.
L'expression peut être :
 - une constante : Exp : A \leftarrow 5
 - Une variable : Exp : A \leftarrow B
 - Une expression arithmétique : Exp : A \leftarrow 2*B+1
 - Une expression logique : Exp : X \leftarrow vrai

Remarque :

- Une constante ne peut jamais figurer à gauche d'une affectation.
- Après l'affectation, l'ancien contenu est perdu pour être substitué par le nouveau contenu.
- Lorsqu'une variable est déclarée, une case mémoire lui est allouée.

Exemple :

Quelles sont les valeurs des variables A, B, et C après l'exécution des instructions suivantes :

```
A $\leftarrow$ 5
B $\leftarrow$ 3
C $\leftarrow$ A+B
A $\leftarrow$ 2
C $\leftarrow$ B-A
```

Instructions	A	B	C
A \leftarrow 5	5	-	-
B \leftarrow 3	5	3	-
C \leftarrow A+B	5	3	8
A \leftarrow 2	2	3	8
C \leftarrow B-A	2	3	1

8.2 L'instruction d'entrée (Lecture)

Cette action permet d'affecter, à une variable, une donnée introduite à partir d'un périphérique d'entrée (clavier). Lors de l'exécution de cette instruction, l'ordinateur attend que l'utilisateur lui fournit la valeur à introduire.

Syntaxe générale :

Lire(nom_variable1, nom_variable2, ...)

Exemple :

Lire (A) signifie "lire une valeur à partir du périphérique d'entrée, qui est le clavier par défaut, et la ranger dans la variable identifiée par A"

Nom	Rôle	Exemple
Tronc(x)	Supprime la partie décimale	Tronc (3.14)=3
Arrondi(x)	Renvoie l'entier le plus proche de l'arrondi de x	Arrondi(9.499)=9 Arrondi (9.5)=10
Abs(x)	Renvoie la valeur absolue de x	Abs(-8)=8
Carré(x)	Renvoie le carré de x	Carré(5)=25
Sin(x)	Renvoie le sinus de x	Sin(PI/2)=1
Cos(x)	Renvoie le cosinus de x	Cos (PI/2)=0
Exp(x)	Renvoie l'exponentielle de x	Exp(0)=1
Ln(x)	Donne le logarithme népérien de x si x positif sinon provoque une erreur	Ln (1)=0
Racinecarré(x) Ou RAC(x)	Renvoie la racine carrée de x si x positif sinon provoque une erreur	Racinecarré(9)=3
Arctan(x)	Renvoie la valeur en radian de arctan (x)	Arctan(1)=PI/4

TABLE 4 – Fonctions arithmétiques standards.

8.3 L'instruction de sortie (Ecriture)

L'ordre d'écriture permet la transmission de l'information de l'ordinateur à l'utilisateur par l'intermédiaire d'un périphérique de sortie.

Syntaxe générale :

Écrire (expression1, expression2, ...)

Exemple :

- Écrire ("Bonjour")
Résultat à l'écran : Bonjour
- Affichage du contenu d'une variable.
 $A \leftarrow 145$
Écrire (A)
Résultat à l'écran : 145

Attention :

- Écrire ("A"), affichera le caractère A.
- Affichage mixte : (Commentaires et variables)

On utilise la virgule comme séparateur. Écrire ("A = ", A). Affichera A = 145

Remarque : Le paramètre expression peut être donné sous la forme de constante, de variable ou d'expression.

Exemple :

Écrire ("Bonjour")
Écrire ("Le nom de la ville : ", Nom_ville)
Écrire (X+Y*2)

9 Présentation générale d'un algorithme

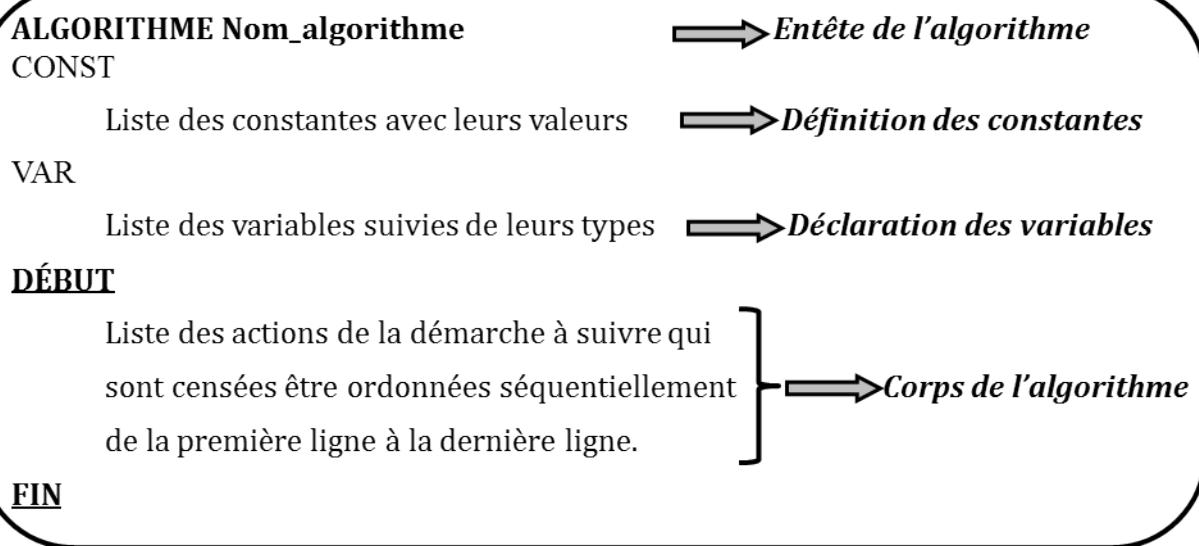


FIGURE 2 – Structure générale d'un algorithme.