

Université Abdelmalek Essaâdi
Ecole nationale des sciences appliquées d'Al Hoceima
(ENSAH)

Algorithmique

Préparé et présenté par
Mr. Ouazzani Chahdi

Année universitaire: 2018/2019

1

10-Dec-18

Structure d'un algorithme et instructions de base

2

10/12/2018

1-Notion de variable, constante et type

Dans un algorithme (ou programme), on va avoir en permanence besoin de **stocker provisoirement des valeurs**. Pour cela on dispose des variables et des constantes.

1.1-Les constantes

Une constante est une **donnée fixe** qui ne varie pas durant l'exécution d'un algorithme :

- Constante : 0.2, 85, 'a', "Ahmed", ...
- Constante symbolique : se caractérise par son **nom** et sa **valeur** (fixe) : $\pi = 3.14$, $g = 9.98$, ...

1.2-Les variables

Une variable est un objet dont le contenu peut être **modifié** par une action durant l'exécution de l'algorithme. Elle se caractérise par son **nom** et son **type**.

3

10/12/2018

Exemple :

On considère un algorithme qui permet de calculer la surface d'un disque en utilisant la relation suivante : $S = \pi \times r^2$, on a alors :

- Les variables sont : S qui représente la surface et r qui représente le rayon, et qui sont tous les deux de type réel.
- La constante est π qui prend la valeur 3.14.

1.3-Le type d'une variable

En algorithmique le **type** d'une variable caractérise la nature des valeurs qu'elle peut contenir.

Il existe trois catégories de type d'une variable :

- Les types **numériques**
- Les types **alphanumériques**
- Les types **logiques** ou **booléens**.

4

10/12/2018

1.3.1- Types numériques

Une variable de type numérique peut contenir que des valeurs numériques comme les nombres.

En algorithmique, on trouve les trois types numériques suivants (les plus utilisés) :

- **Byte** (Octet) : une variable de ce type prend ses valeurs dans l'ensemble $\{0, 1, 2, 3, \dots, 255\}$.
- **Entier** : une variable de ce type prend ses valeurs dans l'ensemble \mathbb{Z} .
- **Réel** : une variable de ce type prend ses valeurs dans l'ensemble \mathbb{R} .

Remarque : pour les nombres Réels, la virgule est représentée par un point (exemple : 3.14, 0.875, 17.5).

5

10/12/2018

1.3.2- Types alphanumériques

Une variable de type alphanumérique prend comme valeur soit un caractère ou une chaîne de caractères.

Pour cela on a deux types :

- **Caractère** : une variable de ce type prend ses valeurs de type symbole (&, €, #, @, +, ...), lettre (a, b, c, ...), ponctuation (?, :, «, », ...), etc.
- **Chaîne** : chaîne de caractère, par exemple une phrase, un numéro de téléphone, matricule, etc.

Remarque :

En algorithmique un caractère est délimité par un double quotes '', et une chaîne de caractères est notée entre guillemets " ".

6

10/12/2018

Exemples :

- Caractère : 'a', '0', '?', '(', '%'.
- Chaîne : "Ahmed", "0678980945", "37aR0sF", "100\$ - 20%".

1.3.3- Types Logiques (ou booléen)

Une variable de ce type ne peut prendre que deux valeurs possibles : VRAI ou FAUX.

7

10/12/2018

1.3.4- Types personnalisés

On peut définir un nouveau type par réduction des autres type à un intervalle de valeurs, appelé type **Intervalle**.

Syntaxe : `Type Nom_Type = Inf..Sup`

Où *Inf* et *Sup* désigne les valeurs extrêmes de l'intervalle considéré.

Exemples :

- **Type** Nombre_Positif = 0..10000
- **Type** Minuscule = 'a'..'z'
- **Type** Majuscule = 'A'..'Z'

8

10/12/2018

2-Structure d'un algorithme

Un algorithme (ou un programme) se compose de trois parties : l'**En-tête**, le **bloc de déclaration** et le **Corps**.

2.1- L'en-tête

Constitué du mot **Algorithme** suivi d'un **nom** identifiant l'algorithme.

Syntaxe : **Algorithme** Nom_Algorithme

2.2- Le bloc de déclaration

Ce bloc se divise en deux parties, la **déclaration des constantes** et la **déclaration des variables**.

9

10/12/2018

a. Déclaration des constantes

Constituée du mot **Constante** suivi du **nom** de la constante et sa **valeur**.

Syntaxe : **Constante** Nom_Constante = **Valeur**

Exemple : Constante Pi = 3.14

b. Déclaration des variables

Constituée du mot **Variable** suivi du **nom** de la variable et son **type**.

Syntaxe : **Variable** Nom_Variable : **Type**

Exemples :

- Variable x : Réel
- Variable c : Caractère
- Variable test: Logique

10

10/12/2018

Remarque:

On peut déclarer plusieurs constantes et plusieurs variables en utilisant les syntaxes suivantes :

Constante Nom_Constante1 = Valeur1
Nom_Constante2 = Valeur2
...

Variable Var1, Var2, ..., VarN : Type1
V1, V2, ..., Vn : Type2
...

1.3- Le corps de l'algorithme

Il contient les étapes de l'algorithme. Il commence par le mot **DEBUT** et se termine par le mot **FIN**.

11

10/12/2018

Alors, un algorithme possède la structure suivante :

Algorithme Nom_Algorithme } **En-tête**

Constante } **Bloc de**
Variable } **déclaration**

Début }
... } **Corps de**
Fin } **l'algorithme**

Remarque:

Dans le cas où l'algorithme n'utilise ni variables ni constantes, alors on peut omettre le bloc de déclaration.

12

10/12/2018

3-Les instructions simples

- ❖ Comme il a été mentionné dans le premier chapitre, généralement, un algorithme reçoit des données en entrée et fournit des résultats en sortie.
- ❖ Celui-ci est **destiné à être implanter dans un ordinateur en tant que programme**. Donc utilisera les unités d'entrée/sortie qui sont généralement le clavier et l'écran.
- ❖ Alors il doit disposer des instructions qui lui permettent de **lire des données** à partir du clavier, et **d'écrire des résultats** à l'écran.

13

10/12/2018



14

10/12/2018

3.1- Les instructions de lecture et écriture

3.1.1- L'instruction de lecture

Cette instruction permet de lire des données saisies au clavier et les enregistrer dans des variables appropriées.

Syntaxe : Lire (Variable)

Lire (Var1, Var2, ..., VarN)

Exemples:

- Lire (N) : cette instruction signifie que l'ordinateur va lire à partir du clavier une valeur, et la stocke dans la variable N.
- Lire (nom, prenom, age) : cette instruction signifie que l'ordinateur va lire deux chaînes de caractères et les enregistrer respectivement dans les variables *nom* et *prenom*, et va lire aussi une valeur numérique qui l'enregistre dans la variable *age*.

15

10/12/2018

3.1.2- L'instruction d'écriture

Cette instruction permet d'afficher des messages sur l'écran ainsi de communiquer avec l'utilisateur.

Syntaxe : Ecrire (obj1, obj2, ..., objN)

Où *obj1, obj2, ..., objN*, est une suite d'objets qui peuvent être des constantes, des variables, des expressions ou des chaînes de caractères.

Remarques :

Dans cette instruction:

- Les chaînes de caractères sont délimitées par des guillemets . Et les autres expressions s'écrivent comme telles.
- Les objets sont séparés par une virgule.

16

10/12/2018

Exemples :

- **Ecrire**("Bonjour tout le monde"): l'ordinateur affiche à l'écran le message *Bonjour tout le monde*.
- **Ecrire**("Votre âge est : ", age, "ans"): l'ordinateur affiche à l'écran le message *Votre âge est 18 ans* (avec *age* une variable numérique dont la valeur est 18).
- **Ecrire**("La somme est : ", a+b): l'ordinateur affiche à l'écran le message *La somme est 7* (avec *a* et *b* deux variables numériques dont les valeurs sont respectivement 3 et 4).

17

10/12/2018

Exercice :

Ecrire un algorithme qui demande à l'utilisateur son âge et ensuite l'affiche à l'écran en utilisant un message approprié.

Solution :

```

Algorithme Demande_Age
Variable age : Entier
Début
  Ecrire("Donner votre age : ")
  Lire(age)
  Ecrire("votre âge est ", age, "ans")
Fin

```

Simulation

18

10/12/2018

3.2- L'instruction d'affectation

L'affectation est l'instruction qui permet d'attribuer à une **variable**, une **valeur**, le **contenu d'une autre variable** ou une **expression**, en utilisant l'opérateur d'affectation \leftarrow .

Syntaxe : Variable \leftarrow Valeur
 Variable \leftarrow AutreVariable
 Variable \leftarrow Expression

Remarque : La valeur ou l'expression affectée doivent être du même type que celui de la variable concernée par l'affectation..

Exemples :

- $A \leftarrow 3$: La variable *A* reçoit la valeur 3
- $X \leftarrow A$: La variable *X* reçoit le *contenu de la variable A*
- $L \leftarrow 'w'$: La variable *L* reçoit le caractère *w*
- $Nom \leftarrow "Ahmed"$: La variable *Nom* reçoit la chaîne de caractères *Ahmed*
- $Y \leftarrow A + X$: La variable *Y* reçoit le *résultat de l'expression A+X*.

19

10/12/2018

3.3- Les expressions et les opérateurs

- ❖ Une expression est composée d'**opérandes**, d'**opérateurs** et de **parenthèses**, et **équivalente à une seule valeur**.
- ❖ Un **opérande** est une quantité sur laquelle une opération est exécutée. Il peut être une valeur, une constante, une fonction, etc.
- ❖ Un **opérateur** est un signe qui relie deux opérandes, pour produire un résultat.
- ❖ Les types d'opérateurs et des opérandes dépendent de la nature de l'expression entrant en jeux.
- ❖ En algorithmique, on utilise trois types d'expressions :
 - Les expressions **arithmétiques**.
 - Les expressions **logiques**.
 - Les expressions **alphanumériques**.

20

10/12/2018

3.3.1- Les expressions arithmétiques

Une expression arithmétique est équivalente à une valeur de type numérique, et on a :

- Les opérandes sont des valeurs numériques, des variables de type numérique ou des constantes numériques.
- Et les opérateurs sont les opérateurs arithmétiques et unaires :

+ : addition	^ : puissance
- : soustraction	Mod : modulo
* : multiplication	Div : division entière
/ : division réelle	Unaire : + et -
- L'opérateur **Mod** donne le reste de la division entière et l'opérateur **Div** donne le quotient de la division entière. Ils ne sont définis que pour les types entiers.
- Les opérateurs unaires servent à désigner le signe d'un opérande.

Exemples:

- $-x/5 + (y + 7)$
- $a*x^2 + b*x + c$
- $(27 \text{ Mod } 5) + z^x$

24

10/12/2018

3.3.2- Les expressions logiques

Une expression logique est équivalente à une valeur de type Logique, et on a :

- Les opérandes sont des valeurs, des variables ou des constantes.
- Pour les opérateurs, il existe deux types :
 - Les opérateurs de **comparaison**.
 - Les opérateurs **logiques**.

Les opérateurs de comparaison sont utilisés dans les **expressions logiques simples**, et les opérateurs logiques sont utilisés dans les **expressions logiques composées**.



Qu'est ce qu'une **expression logique simple** et qu'est ce qu'une **expression logique composée** ?

22

10/12/2018

❖ Les expressions logiques simples

Une **expression logique simple** est une **comparaison** entre deux opérandes, deux expressions ou entre opérande et expression en utilisant les opérateurs de comparaison suivants :

= : égal à	<= : inférieur ou égal
< : inférieur	>= : supérieur ou égal
> : supérieur	<> : différent

Exemples:

- $A < 3$
- $(A + B) <> C/2$
- $B >= (A * 7)$
- $'c' <= 'z'$

Remarques:

- Les objets d'une comparaison doivent être du même type.
- La comparaison entre les objets de type alphanumérique s'effectue en fonction de l'ordre alphabétique des caractères, cet ordre est établi selon le code ASCII.

23

10/12/2018

❖ Les expressions logiques composées

Une **expression logique composée** est la **composée** d'**expressions logiques simples** en utilisant les opérateurs logique **ET**, **OU** et **NON** et éventuellement des parenthèses :

- $Exp1$ **ET** $Exp2$: le résultat est vrai si $Exp1$ et $Exp2$ sont vraies.
- $Exp1$ **OU** $Exp2$: le résultat est vrai si l'une des $Exp1$ ou $Exp2$ est vraie.
- **NON** Exp : le résultat est vrai si l'expression Exp est fausse.

Exemples :

- $(A < 3)$ **ET** $(B > 7)$
- $(C >= (X - 5))$ **OU** $(\text{car} <> 'u')$
- $((D > C)$ **ET** $(X = 18.5))$ **OU** $(\text{NON}(A < 3))$
- $(z <= A$ **OU** $\text{char} = 't')$ **ET** $(\text{NON}(\text{car} <> 'u'))$

24

10/12/2018

3.3.3- Les expressions alphanumériques

Une expression alphanumérique est équivalente à une valeur de type Chaîne, et on a :

- Les opérandes sont des valeurs, des variables ou des constantes. Ils doivent être de type Caractère ou Chaîne.
- On utilise un seul opérateur, c'est l'opérateur de **concaténation &**, il permet de concaténer (ou fusionner) deux caractères, deux chaînes de caractère, ou caractère et chaîne de caractères.

Exemples :

- $ch1 \leftarrow "07ABL"$
- $ch2 \leftarrow "XXY"$
- $ch3 \leftarrow ch1 \ \& \ ch2$: la variable ch3 reçoit la chaîne de caractères "07ABLXXY"

26

10/12/2018

3.3.4- La priorité des opérateurs

❖ La priorité des opérateurs détermine l'ordre d'évaluation de celles-ci. Le tableau suivant montre cette priorité dans l'ordre décroissant :

Opérateur	priorité	Associativité
Unaires : + - NON	1	←
^	2	→
* / Div Mod	3	→
+ - &	4	→
< <= > >=	5	→
<> =	6	→
ET	7	→
OU	8	→
←	9	←

❖ Pour enlever toute ambiguïté on utilise les parenthèses.

26

10/12/2018

Exemple :

On considère l'algorithme suivant :

Algorithme Affectation

Constante $n = 1$

Variable X, Y : Réel

T1, T2 : Logique

Début

$X \leftarrow -8/2 * 3 + n$

$Y \leftarrow -8/2 * (3 + n)$

$T1 \leftarrow n < 7 \text{ OU } X = 2 \text{ ET NON } (Y \geq 10)$

$T2 \leftarrow (n < 7 \text{ OU } X = 2) \text{ ET NON } (Y \geq 10)$

Fin

Question : Évaluez ces expressions et déterminez les valeurs des variables X, Y, T1, T2.

27

10/12/2018

Solution :

$$X \leftarrow -8/2 * 3 + n$$

$$\begin{array}{c} \underline{-8} \\ \underline{-4} \\ \underline{-12} \\ -11 \end{array}$$

$$Y \leftarrow -8/2 * (3 + n)$$

$$\begin{array}{c} \underline{-8} \quad \underline{3 + n} \\ \underline{-4} \quad \underline{* \quad 4} \\ -16 \end{array}$$

$$T1 \leftarrow \underbrace{n < 7}_{\text{Vrai}} \text{ OU } \underbrace{X=2}_{\text{Faux}} \text{ ET NON } \underbrace{(Y \geq 10)}_{\text{Faux}}$$

$$\underbrace{\text{Vrai}} \quad \underbrace{\text{Faux}} \quad \underbrace{\text{Vrai}}$$

$$\underbrace{\text{Faux}}$$

$$\underbrace{\text{Vrai}}$$

3.3.5- Les fonctions mathématiques

Le tableau suivant présente les fonctions utilisées en algorithmique :

Fonction	Type d'argument	Description
Quotient(N1, N2)	Entier	Quotient de la division euclidienne.
Reste(N1, N2)	Entier	Reste de la division euclidienne.
Abs(x)	Réel	Valeur absolue.
Ent(x)	Réel	Partie entière.
Exp(x)	Réel	Exponentielle.
Racine(x)	Réel	Racine carrée.
Ln(x), Log(x)	Réel	Logarithme.
Sin(x), Cos(x), Tan(x)	Réel	Sinus, Cosinus et tangente.
Arrondi(x)	Réel	Arrondi à l'entier le plus proche.
Aléatoire(x)	Réel	Entier pseudo-aléatoire dans [0, x[.

2

Exercice :

Ecrire un algorithme qui permet de calculer la surface d'un disque en utilisant la relation suivante : $S = \pi \times r^2$.

Solution :

```

Algorithme Surface_Disque
Constante Pi = 3.14
Variable S, r : Réel
Début
  Ecrire ("Donnez le rayon : ")
  Lire (r)
  S ← Pi*r^2
  Ecrire ("La surface est : ", S)
Fin

```

30

10/12/2018

4-Quelques règles d'écriture

4.1- Les mots-clés

Les **mots-clés** ou **mots réservés** sont des mots réservés au pseudo-code de l'algorithme, c'est-à-dire constituent le vocabulaire du langage algorithmique.

En voici quelques un :

algorithme, alors, booléen, caractère, cas, chaîne, constante, début, écrire, entier, et, faire, faux, fermer, fichier, fin, findefichier, finitérer, finpour, finrépéter, finselon, finsi, fintantque, fois, fonction, itérer, jusqu'à, lire, non, ou, ouvrir, pour, prendre, procédure, retourner, répéter, réel, selon, si, sinon, tantque, type, variable, vrai, etc.

31

10/12/2018

4.2- Les identificateurs

Les identificateurs servent à désigner les noms des différents objets manipulés par un algorithme: variables, constantes, nom de l'algorithme, etc.

Il sont formés de lettres, des chiffres et le blanc souligné (`_`), le premier caractère ne devant pas être un chiffre.

Il ne doivent pas correspondre à un mot-clé et il ne sont pas sensible à la casse.

Exemples :

- Algo_Test, _total_, _675, valeur_9 : identificateurs corrects.
- 65_T, x-y, a.b. : identificateurs incorrects.
- X et x correspondent au même identificateur.

32

10/12/2018

4.3- Les séparateurs

Les **séparateurs** constituent la ponctuation du langage algorithmique. Voici les séparateurs avec leur rôle :

Symbol	Rôle
: deux points	Sépare un identifiant de son type dans une liste
() parenthèses	Encadrent des expressions
[] crochets	Symboles du composant tableau
. point	Accès à un composant agrégat
, virgule	Sépare les éléments d'une liste
; point-virgule	Sépare les éléments de déclaration
' ' quotes	Encadrent les caractères
"" guillemets	Encadrent les chaînes de caractères
espace	Sépare deux mots en l'absence d'autre séparateurs.