

Chapitre Deuxième : Les Concepts

- I. Notion d'algorithme
- II. Définitions
- III. Structure générale d'un algorithme
 - III.1. La partie déclaration
 - III.2. La partie traitement
- IV. Les concepts
 - IV.0. Le symbole d'affectation
 - IV.1. Les opérateurs
 - IV.2. Les données
 - IV.2.1. Les constantes
 - IV.2.2. Les variables
 - IV.2.3. Les expressions
 - IV.2.4. Les variables composées
 - IV.2.4.1. Les tableaux
 - IV.2.4.2. Déclaration d'un tableau

Chapitre Deuxième



I. Notion d'algorithme

Soit à résoudre l'équation de second degré suivante :

$$3X^2 - 2X + 1 = 0$$

I.1. Formulation (avant l'écriture de l'Algorithme)

- 1^{ère} formulation :

- 1- Détermination du **Discriminant**
- 2- **Tester** le discriminant
- 3- Calculer les **Racines**

- 2^{ème} formulation :

- 1- $\Delta = B^2 - 4 \cdot A \cdot C$

- 2- Si $\Delta > 0$ Alors $X_1 = \frac{-B + \sqrt{\Delta}}{2A}$; $X_2 = \frac{-B - \sqrt{\Delta}}{2A}$

- 3- Si $\Delta = 0$ Alors $X_1 = \frac{-B}{2A}$; $X_2 = \frac{-B}{2A}$

- 4- Si $\Delta < 0$ Alors **Pas de Racines Réelles**

- 3^{ème} formulation

1. Lire A,B,C ;
2. $\Delta = B^2 - 4AC$
3. Si $\Delta > 0$ Alors $X_1 = \dots$;
Sinon....
4.

Algorithme EQUATION_SECOND_DEGRES ;

VAR

A,B,C : ENTIERS ;
DELTA : REEL ;
X1,X2 : REEL ;

DEBUT

LIRE (A,B,C) ;
DELTA ← $B^2 - 4 * A * C$;

SI DELTA >= 0 **ALORS**

Début

$X_1 \leftarrow \frac{-B + \text{SQR}(\text{DELTA})}{2 * A}$;
 $X_2 \leftarrow \frac{-B - \text{SQR}(\text{DELTA})}{2 * A}$;
Ecrire (X1,X2) ;

Fin

Sinon

ECRIRE ('PAS DE RACINE REELLE') ;

FSI ;

FIN.

II. Définitions

- Un algorithme est une **suite d'opérations logique** formulée à l'aide de structures spécifiques dans un **langage naturel** afin de faciliter sa traduction dans un langage de programmation.
- C'est une étape **préliminaire** dans l'écriture d'un programme.
- Un algorithme est la description, **étape par étape**, d'un processus de résolution d'un problème. Il spécifie les opérations nécessaires à effectuer pour l'obtention de la solution d'un problème. Il indique l'**ordre** dans lequel les opérations doivent être exécutées.
- Un algorithme est **une suite cohérente d'actions** à dérouler dans un **ordre déterminé** et permettant d'obtenir **automatiquement** la solution d'un problème.

III. Structure générale d'un algorithme

Un algorithme débute par le mot-clé **ALGORITHME** et se termine par le mot-clé **FIN**.

Un algorithme est composé essentiellement de deux parties :

- Une partie **Déclaration** ;
- Une partie **Traitement** ;

séparées par le mot-clé **DEBUT** qui indique le début de la partie traitement.



III.1. La partie déclaration

La partie déclaration comprend les déclarations de tous les éléments de données utilisés dans la partie traitement de l'algorithme. La déclaration consiste à nommer les différentes variables, spécifier leur type (entier, réel, chaîne, logique,...), leurs dimensions,...

Tous objets (variable, constante, étiquette,...) se trouvant dans la partie traitement doit avoir été déclarée dans la partie déclaration.

La partie déclaration débute par le mot-clé **VAR**. et se termine par le mot-clé **DEBUT** qui indique le début de la partie traitement.

III.2. La partie traitement

Appelé aussi partie procédure, la partie traitement comprends toutes les instructions et toutes les opérations à effectuer sur les données.

On y trouve des instructions de lecture, des instructions d'affectation, des instructions d'écriture, des expressions algébriques, des expressions conditionnelles, des structures de contrôle,...etc.

La partie traitement débute par le mot-clé **DEBUT** et se termine par le mot-clé **FIN** qui indique la fin de l'algorithme.

IV. Les concepts

IV.o. Le symbole d'affectation

Le symbole (\leftarrow) qui désigne l'affectation ;

Ce qui est à **Droite** du symbole (\leftarrow) est affecté à la variable qui se trouve à **Gauche**

IV.1. Les opérateurs

Les opérateurs permettent de faire des opérations sur les termes d'une expression. Ils permettent d'exprimer des expressions.

On distingue principalement 3 types d'opérateurs

- Les opérateurs **Algébriques**
- Les opérateurs **Relationnels**
- Les opérateurs **Logiques**.

IV.1.1. Les opérateurs Algébriques

Ce sont des opérateurs qui servent à exprimer des **expressions algébriques**.

On trouve :

- le symbole (-) qui désigne la **Soustraction** ;
- le symbole (+) qui désigne l'**Addition** ;
- le symbole (*) qui désigne la **Multiplication** ;
- le symbole (/) qui désigne la **Division** ;
- le symbole (^) ou (**) qui désigne l'**Exponentiation**.

IV.1.2. Les opérateurs Relationnels

Ce sont des opérateurs qui servent à exprimer des **expressions conditionnelles**. On trouve :

- Le symbole (>) qui désigne **supérieur à**
- Le symbole (<) qui désigne **inférieur à**
- Le symbole (>=) qui désigne **supérieur ou égal à**
- Le symbole (<=) qui désigne **inférieur ou égal à**
- Le symbole (=) qui désigne **égal à**
- Le symbole (<>) qui désigne **différent de**

IV.1.3. Les opérateurs Logiques

Ce sont des opérateurs qui permettent d'exprimer des **expressions logiques**. On trouve :

- l'opérateur **AND** qui désigne l'**intersection**(ET)
- l'opérateur **OR** qui désigne l'**union** (OU)
- l'opérateur **NOT** qui désigne la **négation**(NON)

IV.2. Les données

Dans un programme, on est appelé à manipuler des objets sur lesquels vont porter l'ensemble des opérations. Ces objets sont des données sur lesquelles porteront l'ensemble des traitements du programme. Les données peuvent être des valeurs fixes (**constantes**) ou des éléments qui sont appelés à prendre plusieurs valeurs durant l'exécution du programme (**variables**).

Les données doivent être **déclarées** dans tout algorithme **avant** leur utilisation.

Les variables sont des **structures de données** utilisées pour la manipulation des différentes données. Il existe plusieurs structures de données :

- les variables simples
- les tableaux ;
- les enregistrements ;
- les fichiers ;

- les piles
- les listes ;
- les ensembles ;
- etc ...

IV.2.1. Les constantes

Ce sont des **données fixes** (valeurs) qui ne changent pas durant l'exécution du programme. On distingue 3 types de constantes :

- les constantes **numériques**
- les constantes **chaînes de caractères**
- les constantes **logiques**

IV.2.1.1. Les constantes numériques

Les constantes numériques sont composées des nombres naturels. On trouve

- les constantes numériques **entières**
- les constantes numériques **réelles**

IV.2.1.1.1. Les constantes numériques entières

Une constante numérique est dite entière si elle ne comporte pas de point (**virgule**) décimal.

Exemples :

12 ; 258 ; 1854 ; -14 ; 0 ; 32 ; 100 ; -250

IV.2.1.1.2. Les constantes numériques réelles

Une constante numérique est dite réelle si elle comprend le point décimal.

Exemples :

12.5 ; 26.75 ; -12.5 ; .25 ; 18.0 ; -14. ; 3.143541 ; 0.

IV.2.1.2. Les constantes chaîne de caractères

Une constante chaîne de caractères est une chaîne de caractères comprise entre les caractères " " .

Exemples :

"ANNABA" ; "ALPHA20" ; "12XY" ; "1999" ; "VAR-2"

IV.2.1.3. Les constantes logiques

Les constantes logiques comprennent les valeurs TRUE (VRAI) et FALSE (FAUX).

IV.2.2. Les variables

Les variables sont des éléments de données qu'on peut **modifier** dans un algorithme. Ce sont des zones logiques (stockées en mémoire) qui peuvent prendre **plusieurs valeurs** durant l'exécution du programme.

C'est une structure qui peut recevoir des valeurs.

Une variable est désignée par un **nom identificateur**. Le nom d'une variable est une suite de caractères alphanumériques dont le **premier** doit être obligatoirement une lettre.

On distingue 2 classes de variables :

- les variables **simples**
- les variables **composées**

IV.2.2.1. Les variables simples

Une variable simple est une structure qui ne peut recevoir qu'**une seule** donnée.

Exemple :

```
A; DELTA; X1; VAL; I; K; MM; B727; RACINE
```

On distingue principalement 3 types de variables :

- les variables **numériques** ;
- les variables **chaîne de caractères** ;
- les variables **logiques**.

IV.2.2.1.1. Les variables numériques

Une variable numérique est une variable qui ne peut recevoir que des constantes numériques.

On distingue deux types de variables numériques :

- les variables numériques **entières** ;
- les variables numériques **réelles**.

IV.2.2.1.1.1. Les variables numériques entières

Une variable numérique de type entier **ne reçoit** que des constantes numériques entières.

IV.2.2.1.1.2. Les variables numériques réelles

Une variable numérique de type réel reçoit des constantes numériques réelles.

IV.2.2.1.2. Les variables chaîne de caractères

Une variable de type chaîne de caractères **ne reçoit** que des constantes chaînes de caractères.

IV.2.2.1.3. Les variables logiques

Une variable logique **ne peut recevoir** que des constantes logiques.

IV. 2.3. Les expressions

Une expression est un ensemble de variables et de constantes relié par des opérateurs

On distingue 3 types d'expression :

- les expressions **algébriques** ;
- les expressions **relationnelles** ;
- les expressions **logiques**.

IV.2.3.1. Les expression algébriques

Une expression algébrique est un ensemble de variables et de constantes numériques reliés par des opérateurs **algébriques**.

Exemple :

```
DELTA ← B^2 - 4*A*C
```

IV.2.3.1.1. Evaluation d'une expression algébrique

L'évaluation d'une expression algébrique s'effectue en tenant compte de la **priorité** des opérateurs en allant **de la gauche vers la droite**.

La priorité des opérateurs est définie comme suit :

- 1- les parenthèses () ;
- 2- l'exponentiation (^) ;

3- la multiplication et la division (*, /)

4- l'addition et la soustraction (+, -)

Exemple :

Soit l'expression algébrique suivante :

$$X = \frac{(A+B^2)}{L+M} + \frac{2(C-K)^3}{A-100}$$

Elle sera traduite comme suit:

$$X \leftarrow (A+B^2)/(L+M) + 2 * (C-K)^3/(A-100)$$

IV.2.3.2. Les expressions relationnelles

Les expressions relationnelles permettent d'exprimer des expressions **conditionnelles** et des expressions de **comparaison**. Dans une expression relationnelle on trouve des variables, des constantes reliées par des opérateurs **relationnelles**.

Exemple :

Si DELTA >= 0 Alors

Si X1 > X2 Alors

Si MOY >= 10 Alors

Si GAMMA = VRAI Alors...

IV.2.3.3. Les expressions logiques

Ce sont des expressions utilisant des variables logiques liées par des opérateurs logiques.

Exemple :

TEST ← A AND B

IV.2.4. Les variables composées

Une variable composée est une structure qui peut recevoir **plusieurs** données à la fois.

C'est un ensemble de données désignées par le **même nom identificateur**.

On trouve :

- les tableaux ;
- les enregistrements
- les fichiers
- les listes
- les ensembles
- les piles
- etc...

IV.2.4.1. Les tableaux

Un tableau est une variable **composée** concernant un ensemble de données ayant **les mêmes caractéristiques (type)** et désigné par le **même nom-identificateur**.

Exemple 1 :

Les mois peuvent être regroupés dans un tableau de 12 éléments appelé **MOIS**

MOIS

JANVIER	FEVRIER	NOVEMBRE	DECEMBRE
---------	---------	-------	----------	----------

Un tableau est représenté (schématisé) par **un ensemble de cases** contenant les données. C'est une représentation schématique qui est différente de la représentation interne en machine ou sur le support.

Exemple 2 :

Les wilayas peuvent être rangées dans un tableau désigné par **WILAYA** de 48 cases où chaque case recevra le nom d'une wilaya.

WILAYA

Adrar	Chlef	Annaba	Guelma	Relizane
01	02			23	24			48

IV.2.4.1.1. L'accès à un élément dans un tableau

L'accès à un élément quelconque d'un tableau est réalisé en utilisant **un indice**. Un indice est une **variable numérique entière** dont la valeur indique **le rang et la position** d'un élément dans le tableau.

Exemple :

Soit un tableau **VAL** de 05 valeurs entières :

17	12	8	6	15
----	----	---	---	----

VAL(2) désigne l'élément de rang 2 du tableau VAL et il contient la valeur 12

VAL(4) désigne l'élément de rang 4 du tableau VAL et il contient la valeur 6

On distingue généralement 2 types de tableaux :

- tableaux **unidimensionnels** (à **une** dimension) ;
- tableaux **bidimensionnels** (à **2** dimensions).

IV.2.4.1.2. Les tableaux unidimensionnels

Ce sont des tableaux linéaires qui utilisent **un seul indice**. Un élément de ce tableau sera référencé par la valeur de son indice.

Exemples :

- Les vecteurs
- T(5) est un tableau de 5 cases.

7	2	8	6	5
---	---	---	---	---

IV.2.4.1.3. Les tableaux bidimensionnels

Ce sont des tableaux qui utilisent **deux indices**, un indice pour désigner **les lignes** et un indice qui désigne **les colonnes**. Un élément de ce tableau sera référencé par la valeur de l'indice des lignes et la valeur de l'indice des colonnes.

