

**Département de Mathématiques**

**MASTER I**

**CONTRÔLE OPTIMAL**

**Cours**

**Algorithmique**

**et**

**Programmation C++**

Pr Tahar BOUHADADA

**(2019-2020)**

## **Chapitre Premier : Introduction à la Programmation**

- I. Préambule : L'algorithmique ?
- II. Exemples de résolution logique
- III. Généralités
  - III.1. Pourquoi enseigner l'Algorithmique ?
  - III.2. Objectif du cours
  - III.3. L'Algorithmique
- IV. Algorithmique et programmation
- V. Définitions
  - V.1. Instruction
  - V.2. Programme
  - V.3. Exemple
  - V.4. Algorithme

## Chapitre Premier

# Introduction à la Programmation

### I. Préambule : L'algorithmique ?

Il est fort possible que vous ayez déjà fait et exécuter des algorithmes !!!!

- Une recette de cuisine
- Monter un meuble

### II. Exemples de résolution logique

#### II.1. Exemples

Considérons les deux énoncés suivants :

#### E1: Calcul de la moyenne de trois nombres avec une calculatrice

1. **Appuyer** sur ON ;
2. **Taper** le premier nombre ;
3. **Appuyer** sur + ;
4. **Taper** le deuxième nombre ;
5. **Appuyer** sur + ;
6. **Taper** le troisième nombre ;
7. **Appuyer** sur / ;
8. **Taper** 3 ;
9. **Appuyer** sur =. (Le résultat est affiché)

#### E2 : Confection d'une omelette de six œufs

1. Casser six œufs dans une coupe
2. Battre les blancs et les jaunes avec une fourchette
3. Mettre de l'huile à chauffer dans une poêle sur la cuisinière
4. Lorsque la poêle est chaude, y verser le contenu de la coupe
5. Enlever la poêle de la cuisinière lorsque l'omelette est cuite

**Les énoncés E1 et E2 décrivent chacun un Algorithme.**

### III. Généralités

#### III.1. Pourquoi enseigner l'Algorithmique ?

- Objectif**: obtenir de la «machine» qu'elle effectue un travail à **notre place**.
- Problème**: **expliquer** à la «machine» **comment** elle doit s'y prendre

- Mais... comment le lui dire ?
- Comment le lui apprendre ?
- Comment s'assurer qu'elle fait ce travail aussi bien que nous ?
- Mieux que nous ?

### III.2. Objectif du cours

- Résoudre des problèmes «comme» une machine
- Savoir **explicitier** son raisonnement
- Savoir **formaliser** son raisonnement
- Concevoir (et écrire) des **algorithmes**:

C'est-à-dire : **séquence d'instructions qui décrit comment résoudre un problème particulier**

### III.3. L'Algorithmique

L'algorithmique est la phase **préliminaire** dans le processus d'élaboration et d'écriture d'un programme.

Toute écriture de programme doit être précédée d'une phase **d'écriture de l'algorithme**.



## IV. Algorithmique et programmation

- La programmation est le processus qui consiste à **traduire** un algorithme dans un langage de programmation afin de pouvoir le soumettre à une machine.
- La programmation est **la technique** d'écriture des programmes.

## V. Définitions

### V.1. Instruction

Une instruction est **l'opération élémentaire** que peut effectuer une machine.

### V.2. Programme

Un programme est un ensemble d'instructions.

- C'est une **suite d'instructions** fournie dans un ordre logique à une machine afin d'obtenir un résultat.
- Un programme comprend des instructions de **lecture**, des instructions d'**écriture**, des **expressions algébriques, logiques, relationnelles**, des **instructions d'affectations**, des **instructions de branchement**, etc....
- Un programme comprend des données en **entrées** qui vont subir des **traitements** afin de donner des **résultats** en sortie.



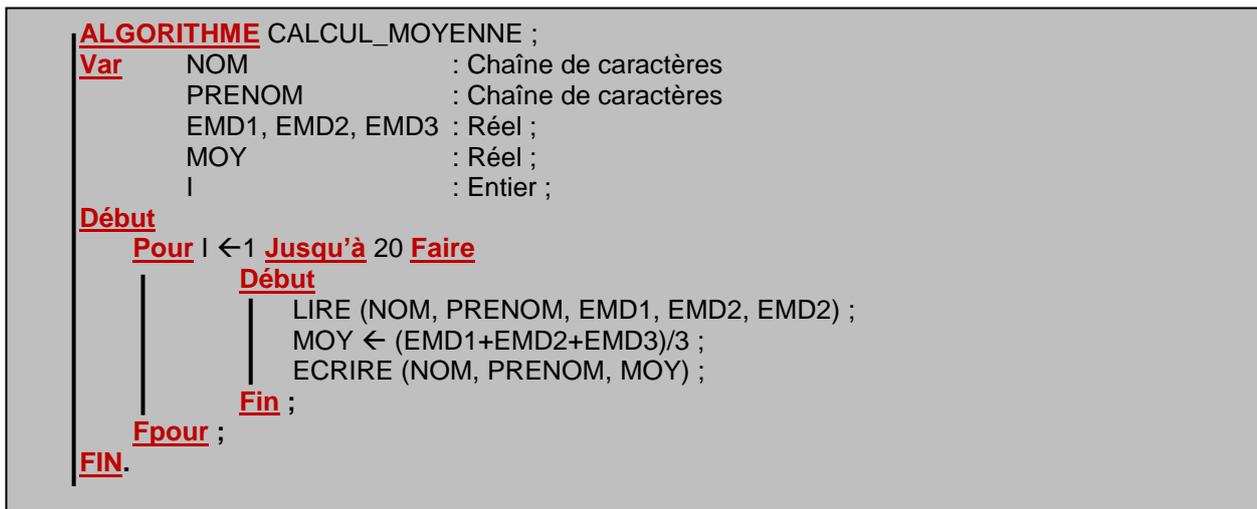
### V.3. Exemple

Soit le problème suivant :

Ecrire un **programme** qui permet de calculer la moyenne de 20 étudiants. On connaît sur chaque étudiant :

- Le nom
- Le prénom
- Les 3 notes d'EMD.

L'algorithme est comme suit :



La traduction de l'algorithme dans un langage de programmation est comme suit :

1. **En utilisant le langage PASCAL**, on aura :

```

PROGRAM CALCUL_MOYENNE ; (* Programme de calcul des moyennes des étudiants*)
Var NOM           : String[1..20] ;
        PRENOM      : String [1..20] ;
        EMD1,EMD2,EMD3 : Real ;
        MOY         : Real ;
        I           : Integer ;
BEGIN
  FOR i :=1 TO 20 DO
    BEGIN
      READLN(NOM, PRENOM, EMD1, EMD2, EMD3) ;
      MOY := (EMD1+EMD2+EMD3)/3 ;
      WRITELN(NOM,PRENOM,MOY)
    END ;
END.
  
```

2. **En utilisant le langage BASIC**, on aura :

```

10 REM PROGRAMME CALCUL DE MOYENNE D'ETUDIANT
20 FOR I = 1 TO 20
30 INPUT NOM, PRENOM, EMD1, EMD2, EMD3
40 LET MOY = (EMD1+EMD2+EMD3)/3
50 PRINT NOM, PRENOM, MOY
60 NEXT I
70 END
  
```

## V.4. Algorithme

### V.4 .1. Définition 1

Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.

### V.4 .2. Définition 2

C'est un "ensemble de règles opératoires propres à un calcul."

- Un peu plus précisément :

Une **séquence de pas de calcul** qui prend un **ensemble de valeurs** comme entrée (**input**) et produit un **ensemble de valeurs** comme sortie (**output**).



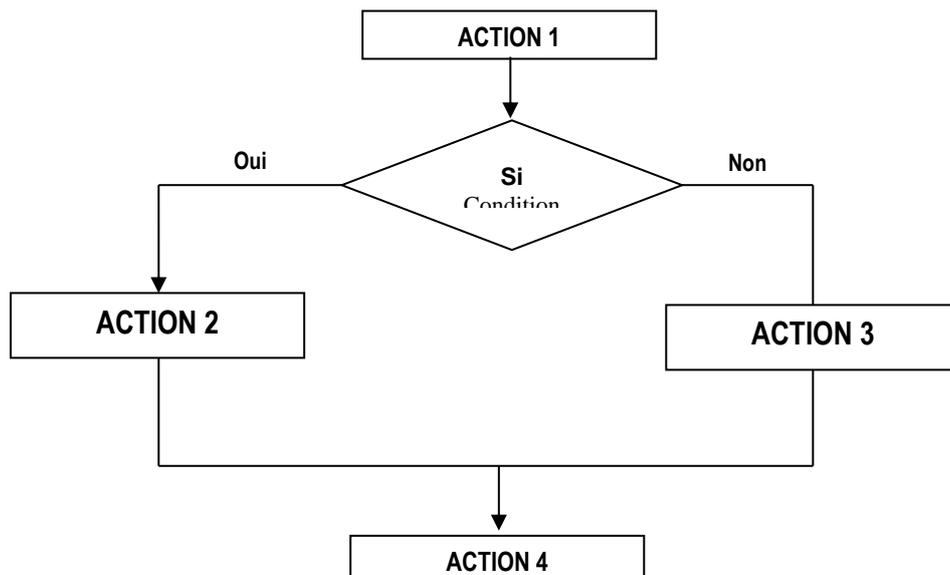
- Un algorithme résout toujours un problème de calcul. L'énoncé du problème spécifie la relation input / output souhaitée.

### V.4 .3. Définition 3

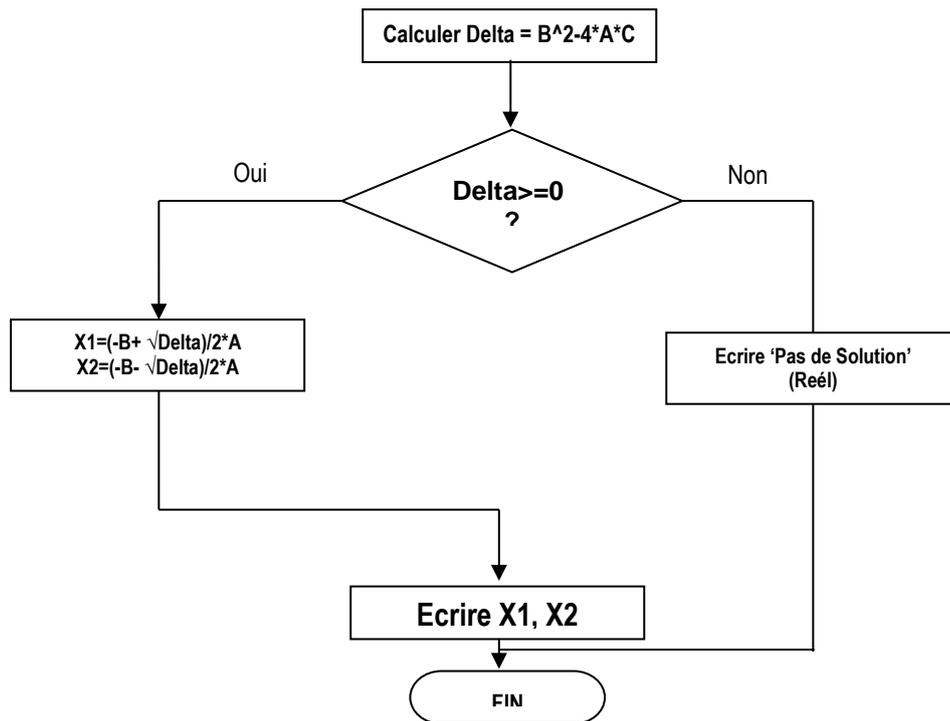
- Un **Algorithme** est une description en **Langage Naturel** de la suite des actions effectuées par un programme structuré.
- Un **Algorithme** est écrit en utilisant un **Langage de Description d'Algorithmes (LDA)**.

L'algorithme ne doit pas être confondu avec le programme proprement dit (tel que Pascal, Basic, Cobol, C,...)

- **Organigramme** : Traduction **graphique** de l'algorithme. Aussi appelé **Algorigramme**, il utilise des formes géométriques pour représenter des actions ou autres (des structures).



#### V.4.4. Exemple d'un Organigramme (Résolution d'une équation du second degré)



#### V.4.4. Caractéristiques

##### Un Algorithme :

- ne dépend pas de l'ordinateur
  - ne dépend pas du langage de programmation
  - ne dépend pas du programmeur
  - est indépendant des détails de l'implémentation.
- ... etc