Chapitre 2 : Notions d'algorithme et de programme

1. Introduction

- un ordinateur est une machine électronique, qui permet de traiter des informations selon des séquences d'instructions prédéfinies (un **programme).**
- L'objectif => Apprendre à écrire des programmes.
- **Etapes de résolution d'un problème mathématique en utilisant l'outil informatique**

La résolution d'un problème mathématique en algorithmique passe par six étapes fondamentales :
1. Analyse du problème.
2. Etablir l'algorithmique.
3. Programmation (utilisation d'un langage de programmation).
4. Vérification et correction du programme.
5. Exécution du programme.

Figure 1 : Cycle de développement d'un programme

Problèmes :
- Addition de deux nombres,
- Calcul de la factorielle d'un nombre,
- Dessiner la courbe d'une fonction,
- Calculer la moyenne des notes des étudiants.

1.1 Algorithme :

Un algorithme est une séquences (suite) d'action élémentaire qui, exécutées par un processeur bien définis réalisera un travail bien précis.

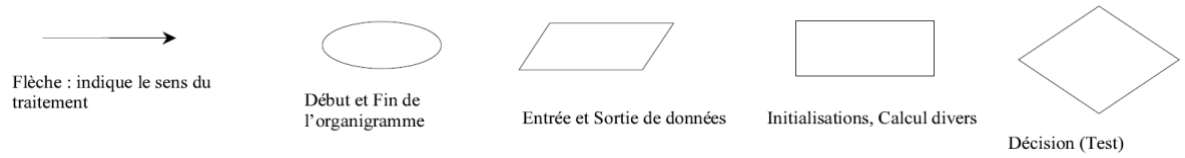
Structure Générale d'un Algorithme
Un algorithme est composé de :

Parties principale	exemple
a. L'entête : qui contient le nom de l'algorithme permettant de l'identifier.	addition;
b. Les déclarations de constantes, variables, structure ..... : cette partie de l'algorithme permet de déclarer tout type de variable, constante ou structure.	**const**a, b, pi; **var** x, y, delta; **tableau** A, B, T1;
c. Les déclarations de fonctions et procédures..... : dans cette partie toutes les fonctions et les sous programmes du problème sont déclarées.	**fonct** f1, f2, fon;
d. Corps de l'algorithmique..... : cette partie de l'algorithme contient toutes les instructions permettant de traiter et de résoudre le problème posé.	**début de l'algorithme** instruction1; instruction2; instruction3; **fin de l'algorithme**

1.2 Organigramme:

Un organigramme est un schéma symbolique conventionnel qui illustre les étapes d'un algorithme et leurs relations.

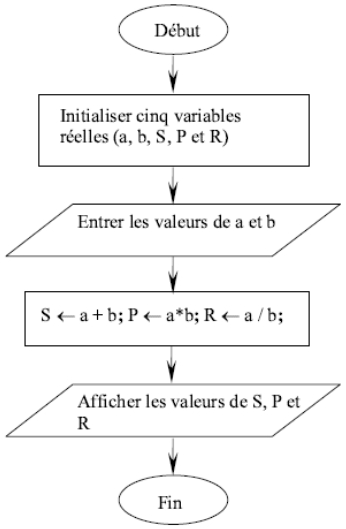
- Nous utilisons l'organigramme parce qu'une représentation graphique aide à la compréhension.



1.3 Programme :

Un programme est un enchaînement d'instruction, écrit dans un langage de programmation, exécutées par un 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. Exemple : **Fortran, Pascal, C++, Java, …**

Exemple : Addition, produit et rapport de deux nombres réels a et b

Algorithme	Organigramme	Programme-4-1 Fortran
Algorithme arithmétique; a, b, S, P, R : réels; **Début** **Lire** (a,b); S ← a + b; P ← a * b; R ← a / b; **Ecrire** (S, P, R); **Fin.**		**Program** arithmetic **real** a, b, S, P, R **read**(*,*) a, b S = a + b P = a * b R = a / b **write**(*,*) S, P, R **end** **Remarque :** les écritures suivantes sont équivalentes : **read**(*,*) a, b **et read***, a, b **write**(*,*) S, P, R **et print***, S, P, R

1.3.1 Structure générale d'un programme Fortran

1. Le nom du programme est introduit par le mot clé **program**, le nom du programme n'est pas obligatoire en Fortran.
2. Dans la partie déclarative du programme les variables utilisées doivent être déclarées avec leurs types correspondant. Dans le programme précédent les deux variables Tf (température en degré Fahrenheit) et Tc (température en degré Celsius) sont des réelles.
3. Dans la partie corps du programme le problème posé est traité et résolus.
4. la fin du programme est marquée par le mot clé **end** (tout simplement) ou bien **end program**<nom du programme>

Program nom du programme
Type de constante, **Paramètre** (identificateur= valeur)
Type des variables identificateur, identificateur
Instruction **1**
Instruction **2**
……
Instruction **n**
End Program nom du programme

1.4 Langages de programmation

Un langage de programmation est un langage artificiel comprenant un ensemble de caractère, de symbole et de mots régis par des règles qui permettent de les assembler, utilisé pour donner des instructions à une machine.

Les langages de programmation les plus utilisés sont : PASCAL, FORTRAN, MATLAB, C, C++ …..etc.

2. Eléments préliminaire d'algorithme-Organigramme-Fortran

2.1 Déclaration des variables et des constantes

Dans la partie déclarative toutes les variables et constantes du problème à traiter doivent être identifiées par un identificateur et déclarées avec leur types.

Déclaration des constantes

Algorithme	Programme Fortran	Exemple Fortran
Const pi=3.141593	Type de la constante, **PARAMETER** :: g=9.81	Ex1 : **real, PARAMETER**:: g=9.81 Ex2 : **integer, PARAMETER**:: theta=45

Déclaration des variables :

Algorithme	Programme Fortran	Exemple Fortran
Var x, y, z : type ; **Ex1 :** **Var** x, y, z : réel ;	Type de la variable, identificateurs (séparés par des virgules)	Ex1 : **real** x, a, delta Ex2 : **integer,** theta, x1, T Ex3 : **character** lettre, section, groupe

2.2 Identificateur

Toutes les variables et constantes du problème à traiter

Type des identificateurs

Type entier : il s'agit des nombres entiers…..-6,-5,-4,-3,-2,-1,0, 1, 2, 3, 4, 5,6,……en fortran on les appelle integer.

Algorithme	Programme Fortran
Var x1, y, z : entier ;	**Integer,** x1, y, z

Type réel : il s'agit des nombres réels appartenant à R, en fortran on les appelle real.

Algorithme	Programme Fortran
Var z2, alpha, B, T : réel ;	**real** x1, y, z2, alpha, B, T

Type caractère : les caractères sont en général les lettres (de a à z) majuscules ou minuscules, les chiffres (de 0 à 9), les ponctuations (la virgules ', ', le point-virgule ';', le point ' .' …..).

Algorithme	Programme Fortran
Var salle, groupe, section : caractère ;	**character** salle, groupe, section

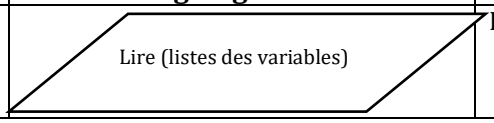
Type chaine de caractères : les chaines de caractères sont un ensemble de caractères.

Algorithme	Programme Fortran
Var nom, prénom, adresse: chaîne de caractère ;	**Character** * n nom, prenom, adress

n dans **character*n** est un nombre entier positif permet de définir la langueur de la chaine de caractère (i .e. nombre de caractère composant la chaine.

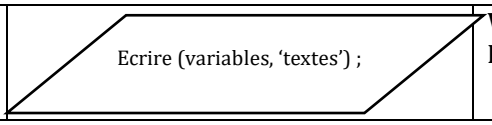
2.3 Action élémentaires

Action de lecture de données(Saisie) : permet d'introduire les données utilisées dans le problème.

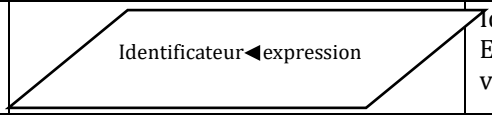
Algorithme	Organigramme	Exemple Fortran
Lire (Listes des variables) ;		**Read (*,*)** Id1, Id2, Id3

Action d'écriture (d'affichage) de résultats : permet d'écrire ou d'afficher les résultats d'un traitement.

Algorithme	Organigramme	Exemple Fortran

Ecrire (variables, 'textes') ;		**Write (*,*)** x1, T, delta **Print***, x1, T, delta

Action d'affectation : Permet d'affecter une valeur à une variable.

Algorithme	Organigramme	Exemple Fortran
Identificateur ← expression ;		Identificateur=expression Ex : x1=a*b (et on lit **reçoit** la valeur du produit a fois b.

Remarque :

L'expression désigne une valeur, exprimée par composition d'opérateurs appliqués à des opérandes, qui sont : des valeurs, des constantes, des variables, des appels à des fonctions ou des sou-expression : Ex1 : f=2*x, Ex2 : z=3*cos (alpha), Ex3 : B*exp (-x*x)

2.4 Opérateurs

- Opérateurs Arithmétiques

Ils permettent l'exécution des opérations usuelles :

+ Addition
- Soustraction
***** multiplication
/ Division
****** Exponentiation

- Opérateurs de relation

Ils expriment une condition arithmétique qui peut être vraie ou fausse :

.EQ. « Égal à »
.NE . « Différent de »
.LT . « Inférieur à »
.LE . « Inférieur ou égal à »
.GT. « Supérieur à »
.GE. « Supérieur ou égal à »

- Opérateurs Logiques

Ils sont au nombre de 3 :

.NOT. Complémentation
.OR. Réunion ou addition logique
.**AND.** Intersection ou multiplication logique

3. Structures de contrôle

Une structure de contrôle sert à contrôler le déroulement d'un traitement qui peut s'exécuter de différentes manières, à savoir, séquentiellement, alternativement ou répétitivement.

3.1 Traitement séquentiel

C'est une suite d'instruction qui s'exécute 'une à la suite de l'autre.

Algorithme	Programme en langage Fortran
Début Instruction1 ; Instruction 2 ; Instruction 3 ; Instruction 4 ; …… ; Instruction N ; Fin.	Instruction1 Instruction 2 Instruction 3 Instruction 4 …….. Instruction N end

3.2 traitement Alternatif

On utilise le traitement alternatif lorsqu'on a le choix d'exécuter telle séquence d'instruction si une condition donnée est vraie, et une autre séquence, si la condition est fausse. Le bloc peut être composé d'une ou plusieurs instructions.

On distingue deux types d'alternatif, l'alternatif simple et l'alternatif composé.

3.2.1 Alternatif simple :

Dans ce cas si la une condition est vérifiés alors une séquence d'instruction est exécuter sinon nr rien faire.