Chapitre 2 :
Notion d'algorithme et de programme

Université de Bejaia- 2020/2021

Département de Technologie

1ere année

Module : Informatique1

1. Exemple introductif

- Si on veut résoudre une équation de 2eme degré

  $ax^2 + bx + c = 0 \qquad a, b, c \text{ sont des reèls, et } a \neq 0$

  par un ordinateur, comment expliquer à l'ordinateur ce qu'il doit faire pour résoudre ce problème et réaliser cette tâche?

- Les étapes à suivre pour réaliser cette tâche par un ordinateur sont :

1. Entrer par le clavier les valeurs des variables a,b et c.
2. Définir l'expression de Delta
3. Tester la valeur de delta

$si\ Delta < 0\ alors\ on\ affiche$ à $l'utilisateur\ que\ l'equation\ n'admet\ pas\ de\ solution\ reelle$

$si\ delta = 0\ alors\ on\ a\ une\ solution\ double\ ,\ on\ calcule\ x1 = -\frac{b}{2a}\ et\ on\ affiche\ x1$

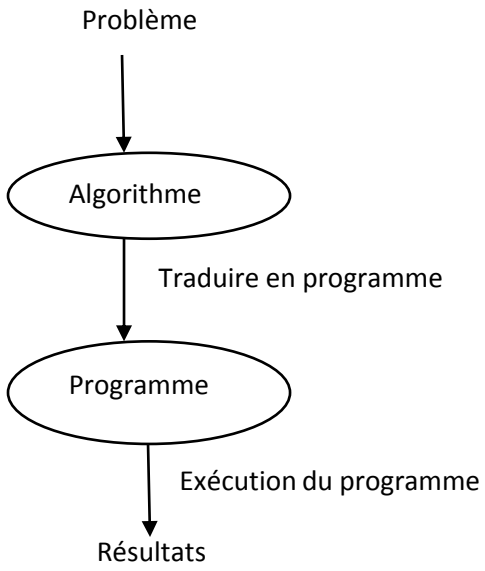
$si\ Delta > 0\ alors\ on\ a\ deux\ solutions$ à $calculer\ puis$ à $afficher$

- **Chaque étape s'appelle instruction ou action.**
- **L'ensemble d'étapes ou d'instructions s'appelle Algorithme.**

2. Du problème au programme

Algorithme

Un algorithme est une suite d'instructions permettant la résolution d'un problème donné



Instruction

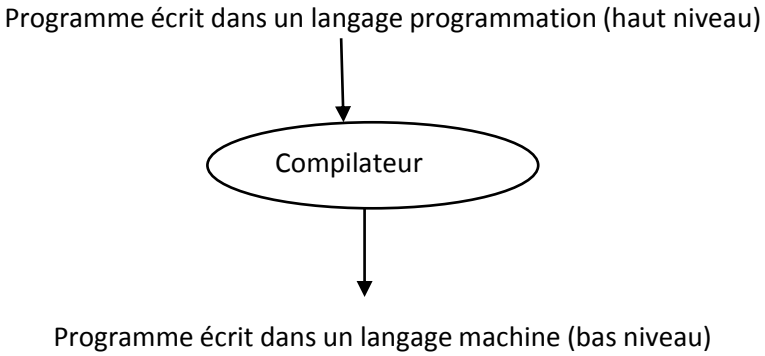
Une instruction ou une action est une étape de l'algorithme qui indique à la machine ce qu'elle doit faire (entrer des données, faire des calculs, afficher les résultats, ….).

Programme

Un programme est la description d'un algorithme dans un langage de programmation. Un programme est une suite d'instructions exécutées par un ordinateur

Langage de Programmation (langage évolué)

Un langage de programmation est un ensemble de règles syntaxiques à respecter utilisés pour écrire un programme, Il existe plusieurs langages de programmation, comme C/C++, PASCAL, Java, Fortran, Matlab, *etc.*



Compilateur

Un compilateur est un programme qui traduit d'autres programmes écrits dans un langage évolué en langage machine.

3.Structure d'un algorithme et d'un programme Pascal

Algorithme nom_Algorithme — Entête

Déclaration des constantes — Partie

Déclaration des variables — déclaration

Debut

- <instruction 1>;
- <instruction 2>;
- <instruction 3>; — partie traitements
- ..... — (Instructions)
- ……
- <instruction N>;

Fin.

Program nom_program;

Uses wincrt;

Déclaration des constantes;

Déclaration des variables;

Begin

Instructions ;

…..

end.

. Mots clés (Mots réservés)

Sont des mots définis dans le langage de programmation, comme **program, for,begin, end, if , var , then, else** …. En Pascal.

. Identificateur

Un identificateur est utilisé pour désigné un objet (nom_objet), il est composés d'une suite de caractère alphanumériques (alphabétiques de [a-z] et [A-Z] et numérique [0-9]) et tiré 8 '_' (trait souligné), il doit commencer par un une lettre et il doit pas être un mot clé du langage.

Exemple : x1, Delta, y, min sont des identificateurs correctes ou valides

1k, min-1, begin sont invalides

. Types simples

type	En algorithmique	En pascal	Exemple de valeur
entier	entier	integer	2020, 12, 56, 1, -5 ,-10
reèl	reel	real	99.99
caractère	caractere	char	'a' , '1', '+'
Chaine de caractères	chaine	string	'11', 'bonjour'
logique	booleen	boolean	Vrai, faux en algorithmique True, false en Pascal

3.1. Partie Déclaration

3.1.1 Déclaration de variables

Une variable correspond à un emplacement mémoire désigné par un identificateur dont une valeur est stockée, cette valeur peut être modifiée durant l'exécution du programme. Une variable est déclaré en respectant la syntaxe suivante :

En algorithmique: variable id_variable : type_variable

En Pascal: **Var** id_variable : type_variable **;**

Exemple

En Algo : variable x : entier
y:reel

En pascal: **Var** x : integer;
y: real;

3.1.2 Déclaration de constantes

Une constante correspond à un emplacement mémoire désigné par un identificateur dont la valeur ne change pas durant l'exécution du programme. Une constante est déclaré en respectant la syntaxe suivante :

En algorithmique: constante id_constante=valeur_constante

En Pascal: **const** id_constante=valeur_constante;

Exemple

En Algo : constante pi=3.14

En pascal: **const** pi=3.14;

3.2.Partie Instruction

3.2.1 Les opérateurs

Opérateur arithmétiques	Opérateurs logiques	Opérateurs relationnels
+, -, * div(division entière), / (division réelle), Mod (reste de la division entière) , -unaire	Et ---> and (en pascal) Ou ---> or Non ---> not	= ----> =(en pascal) < ---> < > ---> > ≤ ---> <= ≥ ---> >= ≠ ---> <>

Priorité des opérateurs

1 les parenthèses () (()), on commence par les plus internes
2 les fonctions
3 - unaire , non
4 *, /, div, mod, et
5 +, -, ou
6 <, >,=,≠, ≥, ≤

Quelques fonctions prédéfinies

$|x|$ ⟶ Abs((x)

$\sqrt{x}$ ⟶ Sqrt(x)

e^x ⟶ Exp(x)

x^2 ⟶ Sqr(x)

3.2.2 Les expressions

- **Une expression arithmétique** est constituée d'opérandes numériques reliés par des opérateurs arithmétiques.

Exemple : 7*5+(8+2-4-4/2)

- **Une expression booléenne** (ou expression logique) est une expression dont le résultat est de type booléen.

Elle peut comporter des opérateurs arithmétiques, des opérateurs de relation et des opérateurs booléens

Exemple : (5+2>8-6)et(7<9)

3.2.3 Instruction d'affectation

Une affectation consiste à mettre une valeur (immédiate, constante, variable ou calculée à travers une expression) dans une variable (espace mémoire).

Syntaxe :

en algo Id_variable <---- valeur

En pascal id_variable:=valeur;

Remarque

Id_variable et valeur doivent avoir le même type

Exemple 1

a <----3;
b <--- 4;
c <---1;
Delata <---b*b-4*a*c
i<--- i+1 (instruction d'incrémentation)
j<--- j-1 (instruction de décrémentation)

a	b	c	Delta
3	4	1	4

Exemple 2

Soit l'algorithme suivant :

Algorithme affectation
Constante a=5
Variables x,y : entier
Debut
X<----5
Y<----x
X<----x+y
Y<----y+2*a
fin.

Program affectation ;
Uses wincrt;
Const a=5 ;
Var x,y: integer;
begin
X :=5;
Y := x;
X := x+y;
Y := y+2*a;
end.

Déroulement

Instruction	x	y
X<----5	5	
Y<----x	5	5
X<----x+y	10	5
Y<----y+2*a	10	15

3.2.4 Instruction d'entrée sortie (lecture/Affichage)

3.2.4.1 Instruction de lecture : permet d'affecter une valeur tapée à une variable

Syntaxe :

En algo	lire(id-var) ,	lire(id_var1, id_var2)
En pascal	read(id_var);	read(id_var1, id_var2);

Exemple :

Lire(x); read(x);
Lire(y,z); read (y,z);

3.2.4.2 Instruction d'affichage: permet d'afficher un message, une valeur d'une variable ou une valeur d'une expression

Syntaxe :

En algo	ecrire(id-var) ,	ecrire(expression),	ecrire('message')
En pascal	write (id_var);	write(expression);	write('message');

Exemple :

ecrire(x); ecrire(x,y,z); ecrire('x=',x); ecrire(x+y); ecrire (' bonjour');

Exercice:
Ecrire un algorithme puis un programme Pascal qui calcule et affiche la somme et la moyenne de deux nombres entiers x et y.

```
Algorithme exercice
Variables x,y , s : entier
          M:reel
Debut
Lire(x,y)
S<----x+y
M<---s/2
Ecrire(S,M)
fin.
```

```
program exercice ;
Uses wincrt;
Var  x,y , S : integer;
          M:real;
begin
Read (x,y);
S := x+y ;
M:=S/2;
write( S, M);
end.
```

Organigramme

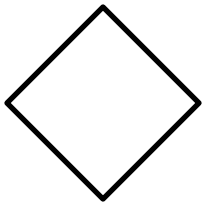
Un organigramme est la représentation graphique des actions d'un algorithme

Les symboles utilisés pour construire un organigramme sont:

Debut et fin

Instructions d'affectations

Instructions de lecture et d'affichage



Tests

Liaison

connecteur

Organigramme exercice