

Université Dr Moulay Taher Saida

Module Informatique1

Faculté technologie

Domaine : Sciences & Technologies - ST

Algorithmique

Présentée par : **Dr** Bekkouche Souad

2017/2018

1- Définitions :

1.1. Algorithmique :

Désigne la science des algorithmes, le premier livre qui a été apparu au (780-850) par Al Khawarizmi qui décrit des méthodes de calculs algébriques.

1.2. Programmation :

Est un lien intermédiaire entre le langage machine le langage humain, qui permet de traiter les données pour obtenir des résultats. Est un ensemble des règles traduit dans un langage de programmation compréhensible et lisible par la machine (ordinateur).

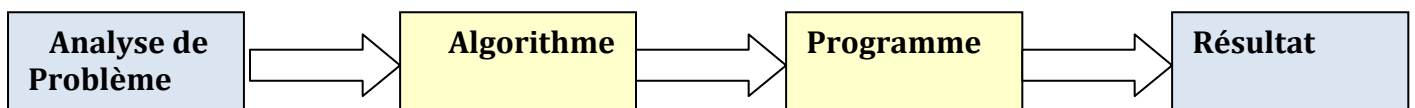
1.3. Définition de l'algorithme

- Est une suite ordonnée ou est une séquence finie des opérations(ou instructions) assimilé ou exécute pour résoudre un problème donné.
- Est une suite des instructions élémentaire réalisée pour restitue en sortie des résultats.

a- **Instruction** : est une action élémentaire exécutable par l'ordinateur.

b- **Un programme** :

Est un algorithme lisible de tout écrit ou saisie dans un langage de programmation (durant cette année on étudier le langage pascal) afin que la machine puisse de l'exécuter d'une manière efficace (donne des bons résultats) et rapide (prendre moins de temps).



➤ ***L'algorithme suit les étapes suivantes :***

- 1- Saisir les données déterminées d'une manière très précise nécessaires pour la résolution un problème donné.
- 2- Le traitement de données saisis pour résoudre le problème.
- 3- L'affichage des résultats obtenus après le traitement.

Un algorithme se compose d'un entête(le nom, déclarations, et d'un corps (Début, suite instruction, fin).

Nom : le nom de l'algorithme

Variable : déclaration des variables

Début

Instruction 1

Instruction2

.

.

..

Instruction N

Fin

1.3.1. La Validité d'un algorithme :

Pour être valide l'algorithme doit être répondre aux critères suivants :

- Le programme doit donner le bon résultat.
- Une fois sa tache est terminé le programme doit s'arrêter.
- Le temps d'exécution doit être acceptable.
- Il faut Le programme gérer mieux la mémoire de l'ordinateur.

1.3.2. Variable, affectation :

a- Variable :

Une variable est un objet identifié par un nom, qui va contenir des données.

Par exemple :

En pascal :

Variables A : Entier

Var A : integer ;

Variable moy : réel

Var moy : real

➤ **Une variable est définie par :**

- **Son identificateur ou son nom :**(le nom peut contenir des lettres ou des caractères), dans notre exemple A et moy.
- **Sa valeur (constante ou une variable) :** Si la valeur de variable n'est pas déterminée et peut être modifiée alors est une variable mais si la valeur de variable ne peut être modifiée donc est une constante.

D'après l'exemple donné la variable **A** et **moy** sont des variables et **C** est une constante.

- **Son type :** pour les déclarations des variables, il faut préciser le type des variables utilisées. on cite par la suite les différents types de variables :

Type numérique :

Par exemple :

Entier : 13

Réel : 14.8

Dans l'exemple donné Le type de A et moy sont entier et réel respectivement.

Type Alphanumérique :

- ✓ **Caractère (char en pascal) :** par exemple 'g','1'.
- ✓ **Chaîne de caractère (ensemble de caractères) par exemple : "Bekkouche "**

Variable Nom : chaîne de caractère [20]

En pascal :

Var Nom : string [20]

Remarque : le chiffre 20 désigne la longueur de la chaîne.

Pour éviter la confusion entre un nombre et une chaîne (un ensemble de caractères), il faut mettre la chaîne entre guillemet " ".

Exemple :

245 est un nombre,mais "245" est un ensemble des caractères 2 ,4 et 5.

Type booléen :

Ce type ne peut prendre que deux valeurs logiques Vrai ou faux, le type booléen gère mieux l'espace mémoire.

b- Affectation :

Est une action élémentaire qui permet d'attribuer ou donner une valeur (constante ou une expression) à une variable. L'affectation est effectuée à l'aide de l'opérateur \leftarrow .

Syntaxe :

Identificateur \leftarrow valeur

Identificateur \leftarrow expression

En pascal Identificateur :=valeur ;

Exemple :

Somme \leftarrow 40

Somme \leftarrow Somme+4

Moyenne \leftarrow Somme/6

En pascal

Somme :=40 ;

Somme :=Somme+4 ;

Moyenne :=Somme/6 ;

Exercice:

Quelles sont les valeurs des variables X et Y :

X \leftarrow 2

Y \leftarrow -7

X \leftarrow X+3 ; Y \leftarrow X+2

X	y
2	-7
5	7

1.3.3. Opérateurs et expression :

Une expression est un ensemble de valeurs, reliées par des opérateurs arithmétiques ou logique.

A- Les opérateurs arithmétiques :

Pour effectuer des calculs arithmétiques on utilise les opérateurs suivants :

- + : l'addition.
- : la soustraction.
- * : la multiplication.
- / : La division.
- ^ : La puissance

Exemple :

$25^2 =$
 $12 * 3 + 5 =$
 $(12 * 3) + 5 =$
 $45 / 2 + 4 =$
 $(45 / 2) + 4 =$

B- Les opérateurs logiques :

Définition : on nomme une variable binaire toute variable qui ne peut prendre que deux valeurs ou deux états : l'état logique 0 qui peut être assemblé à l'absence d'une tension ou l'état logique 1 qui peut être assemblé à la présence d'une tension. pour effectuer des calculs logiques on utilise les opérateurs logiques suivants (et, ou ,xor,...).

Un opérateur logique est un opérateur mathématique qui permet de lier des variables binaires.

- **l'opérateur ET (and en anglais)**

a ET b est VRAI si et seulement si a est VRAI et b est VRAI. Cette loi est aussi notée $a.b$.

- **l'opérateur OU (OR en anglais)**

Elle est définie de la manière suivante : a OU b est VRAI si et seulement si a est VRAI ou b est VRAI. Cette relation logique est aussi notée $(. +)$

- **l'opérateur NON (NOT en anglais)**

La négation de a est VRAIE si et seulement si a est FAUX. La négation de a est notée (non- a , non a , not a).

<u>a</u>	<u>b</u>	A ET b	a ou b	Not a	Not b	A XOR b
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>

Saida

TP N°3
(Algorithmme : affectation)

Exercice N°1 :

1)

Algorithmme affectation

Variables A, B : entier

Début

A ← 1

B ← A+5

A ← 7

Fin

a- Quelles seront les variables A et B après l'acheminement les instructions

b-Réécrire cet algorithmme en programme pascal.

Solution :

A	1	7
B	6	6

Donc les nouvelles valeurs de A et B sont 7 et 6 respectivement.

Program affectation;

Var A,B :integer ;

Begin

A:=1;

B:=A+5;

A:=7;

Writeln('A=',A);

Writeln('b=',B);

Readln;

End.

Il faut ajouter les trios instructions pour l'affichage des nouvelles valeurs de A et B

2)

Variables A, B, C: Entier

A ← 4

B ← 14

C ← A+B

A ← B+C

B ← C

Fin

a-Quelles seront les valeurs des variables A, B et C après exécution les instructions suivantes :

A	4	32
B	14	18
C	18	18

Donc les nouvelles valeurs de A, B et C sont : 32,18 et 18 respectivement.

b-Réécrire cet algorithme en programme pascal.

Program affectation2

Var A ,B ,C :integer ;

Begin

A:=4;

B:=14;

C:=A+B;

A:=B+C;

B:=C;

Writeln('A=',A);

Writeln('B=',B);

Writeln('C=',C);

Readln;

End.

Exercice N°2

1-Quel est le résultat de l'algorithme suivant :

Algorithme affectation3

Variabes Val,Pr : entier

Début

Val ← 440

Pr ← Val*2

Ecrire (Val)

Ecrire (Pr)

Fin

Solution :

Val=440 donc Pr=440x2=880

2-Traduire cet algorithme en programme pascal

Program affectation3 ;

Var Val, Pr :integer ;

Begin

Val :=440 ;

Pr :=Val*2 ;

Writeln('Val=',Val) ;

Writeln('Pr=',Pr) ;

Readln ;

End.

Exercice N°3 :

1-Quel est le résultat de l'algorithme suivant :

Algorithme affectation4

Variables : A, B, C : Chaîne [15]

Début

A ← 'Bon'

B ← 'jour'

C ← 'A+B' **il y'a une différence entre 'A+B' et A+B(la première nous donne un message A+B et la deuxième nous donne la concaténation de deux chaînes Bon et jour qui affiche la chaîne Bonjour**

Ecrire (C)

Fin

2-Traduire cet algorithme en programme pascal

Program affectation4 ;

Var A ,B,C :string[15]

Begin

A :='Bon' ;

B :='jour' ;

C:=concat(A,B);

Writeln('C=',C);

Readln;

End.

Exercice N°4

1-Ecrire un programme permettant de permuter les valeurs de deux variables A et B.

2-Ecrire un programme qui transfère à B la valeur de A, à C la valeur de B et à A la valeur de C.

Solution :

1)

Algorithme permutation

Variables A, B, C : entiers

Début

Lire(A)

Lire(B)

C ← A

A ← B

B ← C

Ecrire(A)

Ecrire(B)

Fin

```

En pascal :
Program permutation
Var A ,B,C :integer ;
Begin
Writeln('A=',A);
Writeln('B=',B);
C:=A;
A:=B;
B:=C;
Writeln('A=',A);
Writeln('B=',B);
Readln;
End.

```

En pascal

Program transfer

Var A ,B,C :integer ;

Begin

2)**Algorithme transfer**

Variables A, B, C : entiers

Début

Lire(A)

Lire(B)

Lire(C)

B←A

C←B

A←C

Ecrire(A)

Ecrire(B)

Ecrire(C)

Fin

1.3.4. Les instructions de base :

Sont des instructions de communication ou de dialogue entre l'utilisateur par le clavier (saisir les données pour le traitement)et la machine ou l'ordinateur(par les périphériques de sortie (Ecran ou m'imprimante).les instructions de base sont basé sur :

a- La lecture de données : permet de saisir les données en entrée**Syntaxe :****Lire (identificateur1, identificateur2,.....identificateur N)****Exemple :**

Lire(X)

Lire(x,y,z)

En pascal :

Read(X)

Read(x,y,z)

Exercice : écrire un algorithme qui permet de lire les variables entières A et B.

Algorithme Affiche

Variables A,B :entier

Début

Lire(A)

Lire(B)

Fin

```
En pascal :  
Program affiche ;  
Var A, B : integer ;  
Begin  
Writeln('A=',A);  
Read(A);  
Writeln('B=',B);  
Readln(B);  
End.
```

b- L'écriture de données :

Permet à l'utilisateur d'afficher des messages ou des résultats obtenus après l'exécution des programmes.

Exercice : écrire un algorithme qui permet d'afficher le message bonjour.

Algorithme affiche2

Début

Ecrire ("Bonjour")

Fin

```
En Pascal  
Program affiche2  
Begin  
Writeln('Bonjour') ;  
Readln ;  
End.
```

Université Dr Moulay Tahar

Module : Informatique 1
2017/2018

Saida

TP N°4
(Algorithmes simples)

Exercice N°1 :

- Ecrire un Algorithme qui affiche le message suivant :

```
*****
Je suis un étudiant en sciences de la Technologie
Le module informatique est intéressant.
*****
```

- Enregistrer le programme sous le nom affiche dans le répertoire créé au début.
Remarque : Ici vous avez créé votre programme source : affiche.pas (vérifier dans le répertoire).
- Compiler votre programme à l'aide de la commande compile du menu compile.
Remarque : Ici vous avez créé votre programme exécutable : affiche.exe (vérifier dans le répertoire) - La compilation est la phase dans laquelle le compilateur Pascal corrige les erreurs.

Solution :

Algorithme affiche

Début

Ecrire (" *****Je suis un étudiant en sciences de la
Technologie.....*****")

Fin

En pascal :

Program affiche ;

Writeln('*****Je suis un étudiant en sciences de la
Technologie *****');

Readln ;

End.

Exercice N°2 :

- Donner des exemples sur les types : **Real, Boolean, Char, String.**
- Elaborer un algorithme qui permet de saisir un nombre réel N et puis de l'afficher.

Exemple :

Real: 23.45

Integer: 23

Boolean: true or false

Char:'Z'

String:'Bekkouche'

2)

Algorithme affiche

Variables N:reel

Début

Ecrire ('donner la valeur de N')

Lire(N)

Fin

En pascal:

```
Program affiche ;
Var N: real;
Begin
Writeln('donner la valeur de N');
Readln(N) ;
Readln ;
End.
```

Exercice N°3 :

- Ecrire un programme qui permet de calculer la surface d'un cercle de rayon R.

Algorithme surface

Variables S, R : entier

Début

Lire(R)

$S \leftarrow (R/2) * (3,14)^2$

Écrire(S)

Fin

Program surface;

Var S, R: integer;

Begin

Writeln('R=');

Readln(R);

S:=(R/2)*sqr(3.14);

Writeln('la surface S=',S);

Readln;

End.

Exercice N°4 :

- Ecrire un programme en pascal qui permet de résoudre une équation au 1 degré ($ax+b=0$, $a \neq 0$).

Algorithme équation

Variables a,b,x :reel

Début

Lire(a)

Lire(b)

Écrire ('la solution de l''eq =' , -b/a)

Fin

```
En pascal
Program equation;
Var a,b,x :real ;
Begin
Writeln('a=');
Readln(a);
Writeln('b=');
Readln(b);
Writeln('la solution de l''eq',-b/a);
Readln ;
End.
```

2. Les structures conditionnelles

Si on tombe dans une situation ou on veut choisir entre deux possibilités

Par exemple :

Après l'évaluation d'examen j'aurai la note :

Si la note ≥ 10 alors je serai admet sinon je serai ajourné

Donc en algorithmique les structures conditionnelles nous offre cette possibilité.

a- Les structures simples :

Syntaxe :

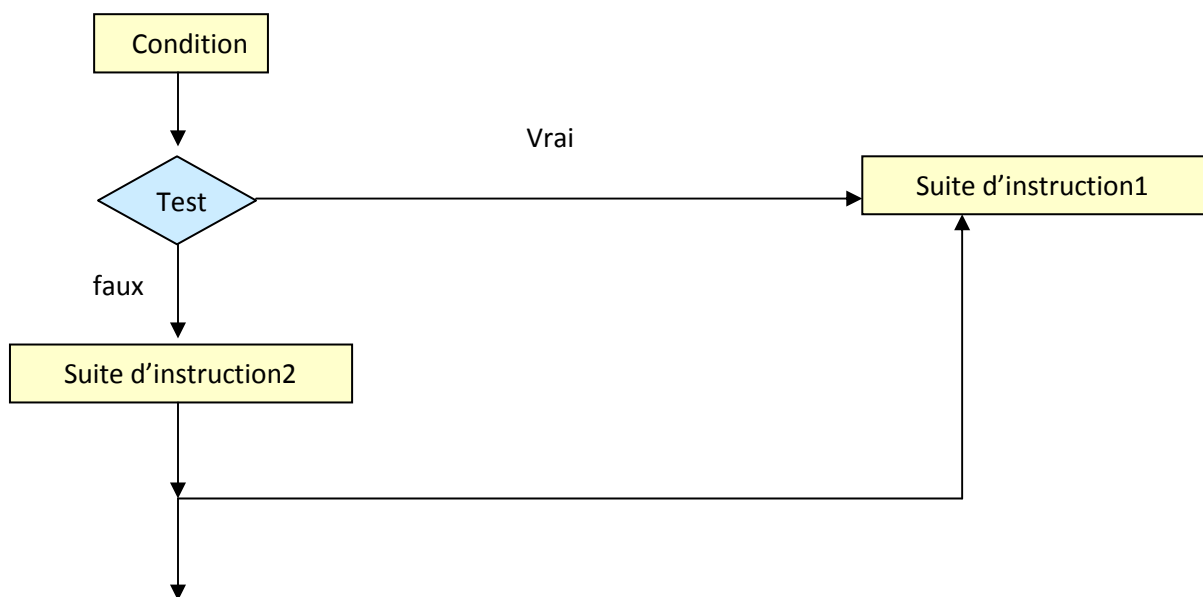
Si (conditions) alors instruction A

Sinon instruction B

En Pascal:

If (condition) then instruction A

Else instruction B



La condition est une action sous forme une comparaison situé entre deux parenthèse nous donne que deux valeur soit vrai ou fausse : si la condition est vérifiée donc nous donne la valeur vrai mais si la condition n'est pas vérifiée nous donne la valeur fausse.

Exemple : écrire un algorithme qui permet de calculer et d'afficher la valeur absolue d'un nombre saisi..

```

Algorithme Calcul
Variables x : entier
Lire(x)
Si(X>0) alors écrire ("la valeur absolue= ",+x)
Sinon écrire ("la valeur absolue= ",-x)
Fin si
Fin
  
```

<pre> En pascal : Program calcul ; Var x : integer ; writeln('x=') ; Readln(x) ; If(x>0) then writeln('la valeur absolue est',+x) Else witeln('la valeur absolue est ',-x); Readln ; End. </pre>
--

2.2. Structures conditionnels avec une condition composée :

Une condition peut être complexe dans le cas où on utilise des conditions reliées par des opérateurs logiques : 'ET'(and en pascal) , 'OU'(or en pascal) et 'NON'(NOT en pascal).

Exemple : Si $(A \geq 2)$ ET $(A \leq 10)$ alors $A \leftarrow A + 12$; A compris entre 2 et 10

En pascal : `if (A>=2) and (A<=10) then A :=A+12 ;`

2.3. Structures conditionnels imbriquées :

Syntaxe :

Si (condition1) alors

Si (condition2) alors instruction1

Fin si

 Sinon si (condition3) alors instruction 2

 Sinon si (condition 4) alors instruction3

Fin si

 Fin si

 Fin si

Exemple :

Algorithme calcul

Variables n : entier

Ecrire ("entrer un nombre ")

Lire(n)

Si (n<0) alors écrire ("ce nombre est négatif")

Sinon

 Si(n=0) alors écrire("ce nombre est nul")

 Sinon écrire ("ce nombre est positif")

 Fin si

 Fin si

Fin

En pascal :

```
Program calcul ;
```

```
Var n :integer ;
```

```
Writeln('entrer un nombre') ;
```

```
Readln(n) ;
```

```
If (n<0) then writeln('ce nombre est  
negatif)
```

```
Else
```

```
    If (n=0) then writeln('ce nombre est nul')
```

```
    Else writeln('ce nombre est positif');
```

```
Readln ;
```

```
End.
```

Rq : Le sinon se rapporte toujours le dernier si qui n'a pas encore de sinon associé.

Exemple2 : remplir le tableau suivant

Algorithme test

Variables b, x : entier

Lire(b)

x ← 1

Si (b>=0) alors

si (b==0) alors x ← 3

Sinon x ← 2

fin si

fin si

écrire(x)

fin

b	-1	0	1
x	1	3	2

2.4. Structures de choix

Syntaxe

:

Cas variable parmi :

Valeur 1 :instruction1

Valeur2 :instruction2

.

.

.

Valeur n : instruction n

Fin

Exemple :

Cas nombre parmi

0 : écrire ("nul")

2,4,6 : écrire ("pair")

1, 3,5 : écrire ("impair")

Fin

En pascal

Case variable of

Valeur1 :instuction1

Valeur2 :instruction2.

.

.

Valeur n : instruction n

End ;

En pascal :

Case nombre of

0 :writeln('nul') ;

2,4,6 :writeln('pair') ;

1,3,5 :writeln('impair');

End;

Université de Dr Moulay Tahar
Saida

2017/2018

Module : Informatique 1

Niveau : 1ère année

Fiche TP N°5
Les Structures Conditionnelles

Exercice N°1 : l'instruction conditionnelle (**ifthenelse**)

Transformer l'algorithme suivant en programme pascal, puis donner le résultat de l'exécution :

```
Algorithme test
Variables x , y : réels
Début
Lire(x)
Si x > 5
Alors affecter à y la valeur 2 x
Sinon affecter à y la valeur 3 x
FinSi
Ecrire(y)
Fin
```

```
Program test;
Var x,y :real ;
Begin
Writeln('x=');
Readln(x);
If(x>5) then y:=2*x
Else y:=3*x;

Writeln('y=',y);
Readln;
```

x	4	-3	6	-2
y	12	-9	12	-6

affecter à y la valeur 2 x signifié : $y \leftarrow 2 * x$

Exercice N°2 : l'instruction conditionnelle (**ifthenelse**)

Ecrivez un programme qui donne le maximum de trois nombres saisis au clavier.

Effectuez des tests pour : (2 ,5 ,8), (3, 1 ,3), (8, -6, 1).

Algorithme max
 Variables a,b,c :entier
 Début
 Lire(a)
 Lire(b)
 Lire(c)
 Si (a>b) alors max ←a
 Sinon max← b
 Si(c>max) alors max← c
 Fin si
 Fin si
 Ecrire (max)
 fin

```

Program maximum;
Var a,b,c :integer ;
Begin
Writeln('a=');
Readln(a);
Writeln('b=');
Readln(b);
Writeln('c=');
Readln(c);
If(a>b) then max:=a
Else max:=b
If(c>max)then max:=c;
Writeln('le maximum=',max);
Readln;
End.
  
```

Exercice N°3 : l'instruction conditionnelle (ifthenelse) avec plusieurs instructions

Ecrire un programme qui calcule la racine d'un nombre avec un test au préalable si le nombre est positif ou négatif.

Algorithme racine
 Variables n : entier
 Début
 Lire (n)
 Si (n<0) alors écrire ('on ne peut pas calculer la racine')
 Sinon R ←(N)^{1/2}
 Ecrire(R)
 Fin

```

Program racine ;
Var N :integer ;
Begin
Writeln('N=');
Readln(N);
If (n<0) then writeln('on ne peut
pas calculer la racine')
Else R :=sqrt(N) ;
Writeln('la racine=',N) ;
Readln ;
End.
  
```

Exercice N°4 (case of)

On utilise la structure de choix multiple **case of**, écrire un programme qui affiche "pair", "impair", ou "nul", en fonction d'un nombre donné.