

Chapitre 2 :

Notion d'algorithme et de programme

Université de Bejaia- 2020/2021

Département de Technologie

1ere année

Module : Informatique1

4. Structures de contrôle

4.1 Structures de contrôle conditionnelles

4.1.1 Structure conditionnelle simple

Syntaxe

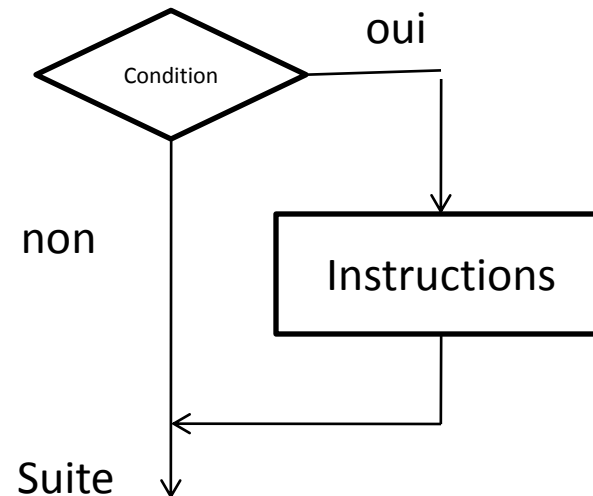
En Algo Si <Condition> alors
 <instruction(s)>
finsi;

En Pascal if <condition> then
begin
 <instruction(s)>;
end;

Fonctionnement

Si la condition est vraie, on exécute le bloc d'instructions, sinon, le bloc d'instructions ne sera pas exécuté.

Organigramme



Remarque : Lorsque le bloc d'instructions est composé d'une seule instruction, les deux mots clés **begin** et **end** sont facultatifs (ne sont pas nécessaires).

Exemple

```
Algorithme exemple;  
Variables x,y : entier;  
Debut  
Lire(x,y);  
Si x>y alors  
  x←x+10;  
Finsi;  
y←y-5;  
Ecrire(x, ' ', y);  
fin.
```

```
program exemple;  
Uses wincrt;  
Var x,y : integer;  
begin  
  Read (x,y);  
If x>y then  
  begin  
    x := x+10 ;  
  End;  
  y:=y-5;  
  write( x, ' ', y);  
end.
```

Déroulement pour x=7 et y=3

Instructions	x	y	Affichage
Lire(x,y)	7	3	
Si x>y vrai x←x+10	17		
y←y-5		-2	
Ecrire(x, ' ', y)			17 -2

Déroulement pour x=4 et y=6

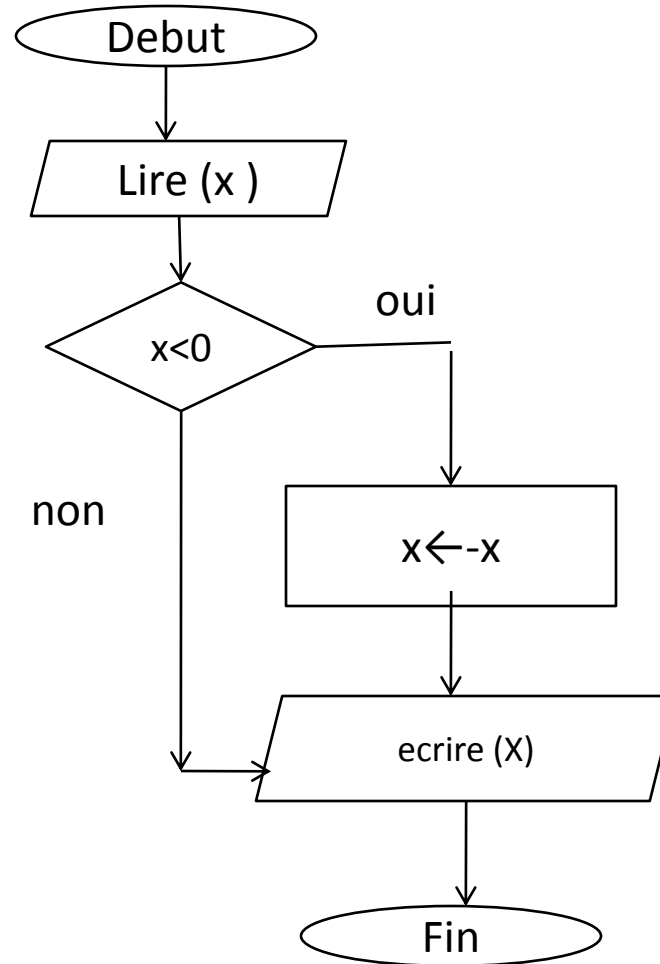
Instructions	x	y	Affichage
Lire(x,y)	4	6	
Si x>y faux			
y←y-5		1	
Ecrire(x, ' ', y)			4 1

Exercice 1: Ecrire un algorithme/programme pascal qui calcule la valeur absolue d'un nombre réel x donné par un utilisateur (sans utiliser la fonction prédéfinie `abs`).

```
Algorithme exercice1;  
Variable x : reel;  
Debut  
Lire(x);  
Si x<0 alors  
  x ← -x;  
Finsi;  
Ecrire(x);  
fin.
```

```
program exercice1;  
Uses wincrt;  
Var x:real;  
begin  
  Read (x);  
  If x<0 then  
    begin  
      x := -x ;  
    End;  
  write( x);  
end.
```

Organigramme correspondant



4.1.2 Structure conditionnelle alternée(double)

Syntaxe

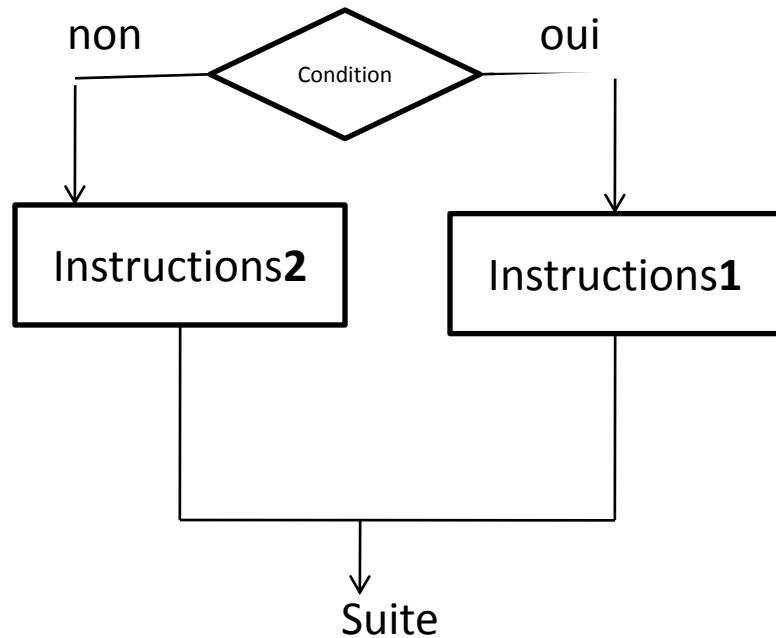
En Algo Si <Condition> alors
 <instruction(s)1>
sinon
 <instruction(s)2>
finsi;

En Pascal if <condition> then
begin
 <instruction(s)1>;
end
else
begin
 <instruction(s)2>;
end;

Fonctionnement

Si la condition est vraie on exécute le premier bloc, sinon on exécute le second.

Organigramme

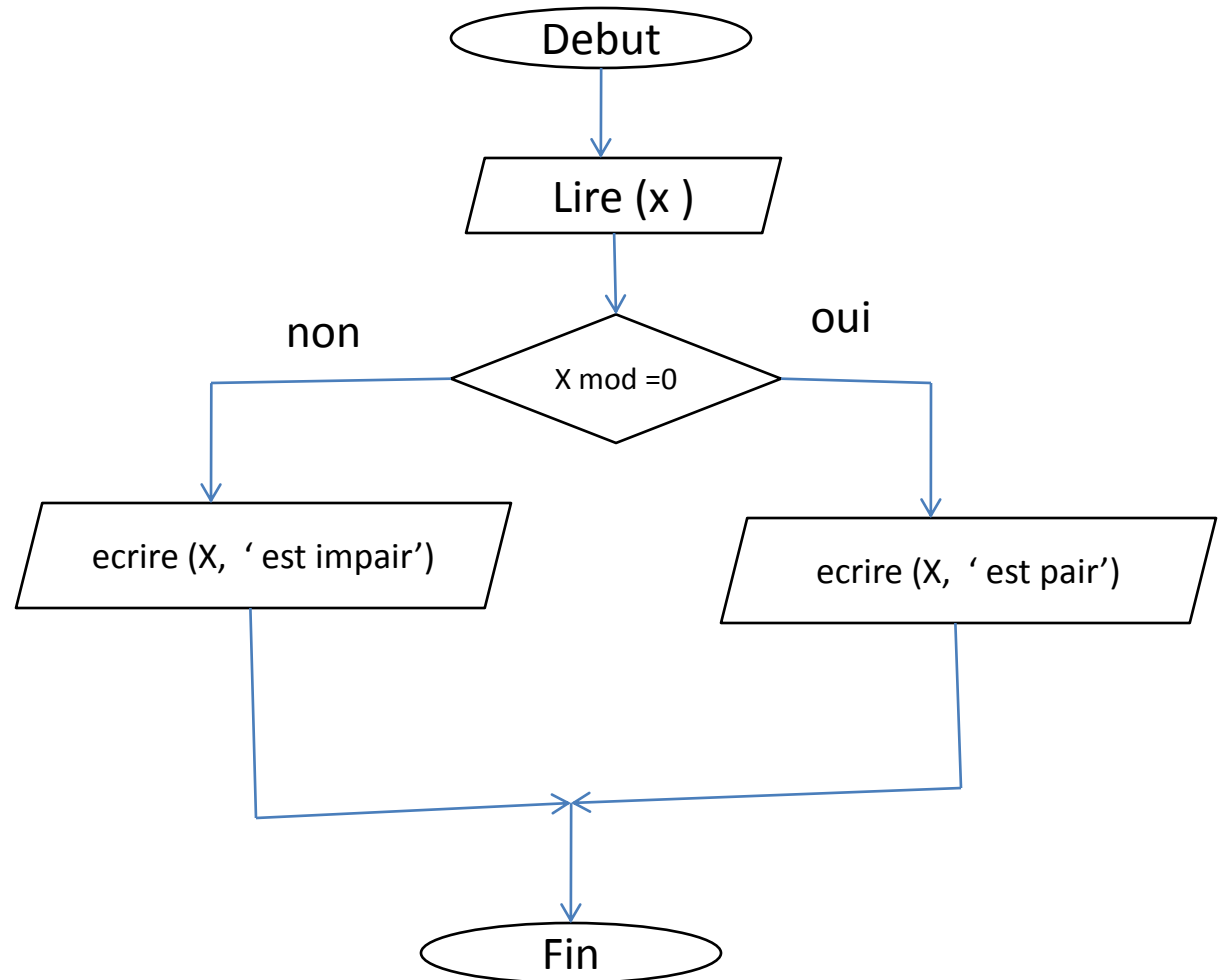


Exercice 2 : Ecrire un algorithme /programme qui vérifie si un nombre entier x est pair ou impair .

```
Algorithme exercice2;  
Variable x : entier;  
Debut  
Lire(x);  
Si x mod 2 =0 alors  
    ecrire ( x, ' est pair ' )  
Sinon  
    ecrire (x, ' est impair ' )  
Finsi;  
fin.
```

```
program exercice2;  
Uses winCRT ;  
Var x :integer ;  
begin  
Read (x);  
If x mod 2=0 then  
    Write (x, ' est pair ' )  
Else  
    write( x, ' est impair ' );  
end.
```

Organigramme correspondant



➤ Structure conditionnelle alternée imbriquée

Syntaxe

En Algo Si <Condition1> alors
 <instruction(s)1>
sinon
 Si <Condition2> alors
 <instruction(s)2>
 sinon
 <instruction(s)3>
 finsi;
finsi;

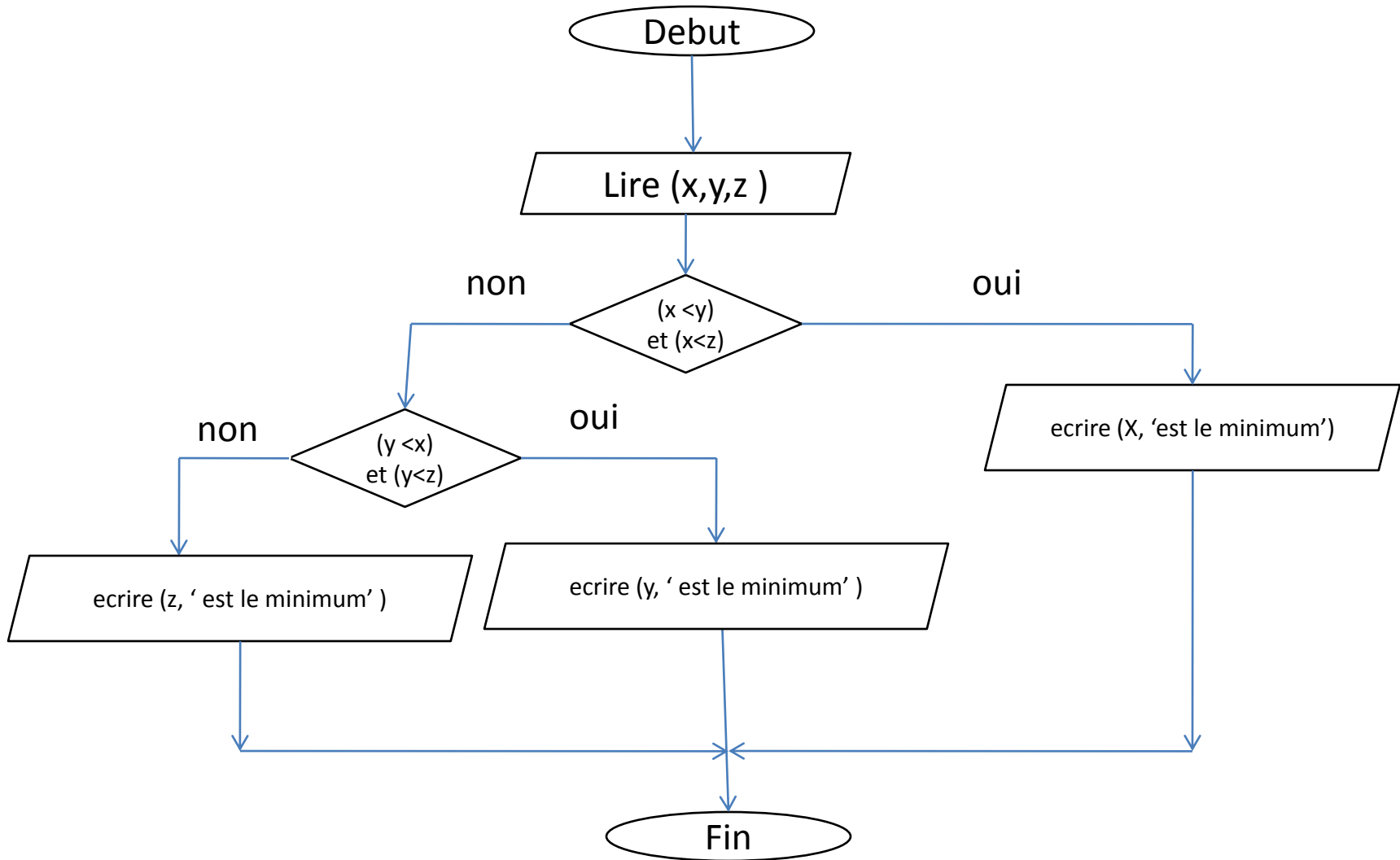
En Pascal if <condition1> then
begin
 <instruction(s)1>;
end
else
begin
 if <condition2> then
 begin
 <instruction(s)2>;
 end
 else
 begin
 <instruction(s)3>;
 end;
end;
end;

Exercice 4: écrire un algorithme/programme pascal qui affiche le minimum de trois nombres entiers x,y et z.

```
Algorithme exercice4;  
Variable x ,y ,z: entier;  
Debut  
Lire(x,y,z);  
Si (x <y) et (x<z) alors  
    écrire ( x, ' est le minimum ' )  
Sinon  
    Si (y <x) et (y<z) alors  
        écrire(y, ' est le minimum')  
    sinon  
        écrire(z, 'est le minimum')  
    finsi;  
Finsi;  
Fin.
```

```
Program exercice4;  
Uses wincrt;  
Var x ,y ,z: integer;  
begin  
read(x,y,z);  
if (x <y) and (x<z) then  
    write ( x, ' est le minimum ' )  
else  
    if(y <x) and (y<z) then  
        write(y, ' est le minimum')  
    else  
        write(z, 'est le minimum') ;  
End.
```

Organigramme correspondant



Exercice 5 : Ecrire un algorithme/programme Pascal qui permet de résoudre une équation de 2eme degré $ax^2 + bx + c = 0$ *a, b, c sont des reèls, et a ≠ 0*

```
Algorithme exo5
Variables a,b,c,delta,x1,x2:reel
Debut
Lire(a,b,c);
Delta← b*b-4*a*c
Si delta>0 alors
  x1←(-b-sqrt(delta))/(2*a)
  x2 ← (-b+sqrt(delta))/(2*a)
  ecrire (x1, x2)
Sinon
  Si delta=0 alors
    x1 ← -b/(2*a)
    ecrire (x1)
  sinon
    ecrire(' pas de solution réelle ')
  finsi ;
Finsi;
Fin.
```

```
Program exo5;
Uses wincrt;
Var a,b,c,delta,x1,x2:real;
Begin
  Read(a,b,c);
  Delta:=b*b-4*a*c;
  if delta>0 then
    begin
      x1:=(-b-sqrt(delta))/(2*a);
      x2 :=(-b+sqrt(delta))/(2*a);
      write (x1, x2);
    end
  else
    if delta=0 then
      begin
        x1:= -b/(2*a);
        write (x1);
      end
    else
      write(' pas de solution réelle ');
  End.
```

Organigramme correspondant

