

# **Chapitre 2 :**

# **Notion d'algorithme et de programme**

Université de Bejaia- 2020/2021

Département de Technologie

1ere année

Module : Informatique1

## 4.2 Structures de contrôle répétitives (itératives)

Ce type de structure est appelé « boucles »

### 4.2.1 Boucle pour (for )

**Syntaxe :**

```
pour <compteur> ← <valeur_initiale> à <valeur_finale> faire  
    <instruction(s)>  
finPour;
```

**En Pascal**

```
for <compteur>:=<valeur_initiale> to <valeur_finale> do  
begin  
    <instruction(s)>;  
end;
```

## Fonctionnement

- Pour chaque valeur de la variable <compteur > qui varie de la valeur initiale à la valeur finale avec un pas d'incrémentation égal à 1, le bloc <instruction(s)> sera exécuté.
- Quand la valeur initiale égale à la valeur finale, <instruction(s)> est exécuté et la boucle sera terminée.

## Remarques

- <compteur> est une variable de type ordinal (entier ou caractère)
- Le pas est par défaut de 1.
- Si le pas est de (-1), la syntaxe en Pascal devient :

```
for <compteur>:=<valeur_finale> downto <valeur_initiale> do  
  begin  
    <instruction(s)>;  
  End ;
```

- Le changement de la valeur du compteur est automatique ; le pas est de 1 ou de -1.

## Exemple

Algorithme Affichage

Variables i,n :entier

Debut

Ecrire ('donner n :')

lire(n)

Pour i←1 à n faire

    Ecrire (i, ' Bonjour à tous')

Finpour ;

Fin.

**Program** Affichage ;

**Uses** wincrt ;

**Var** i ,n : integer ;

**Begin**

Write('donner n :');

Read(n);

**For** i:= 1 **to** n **do**

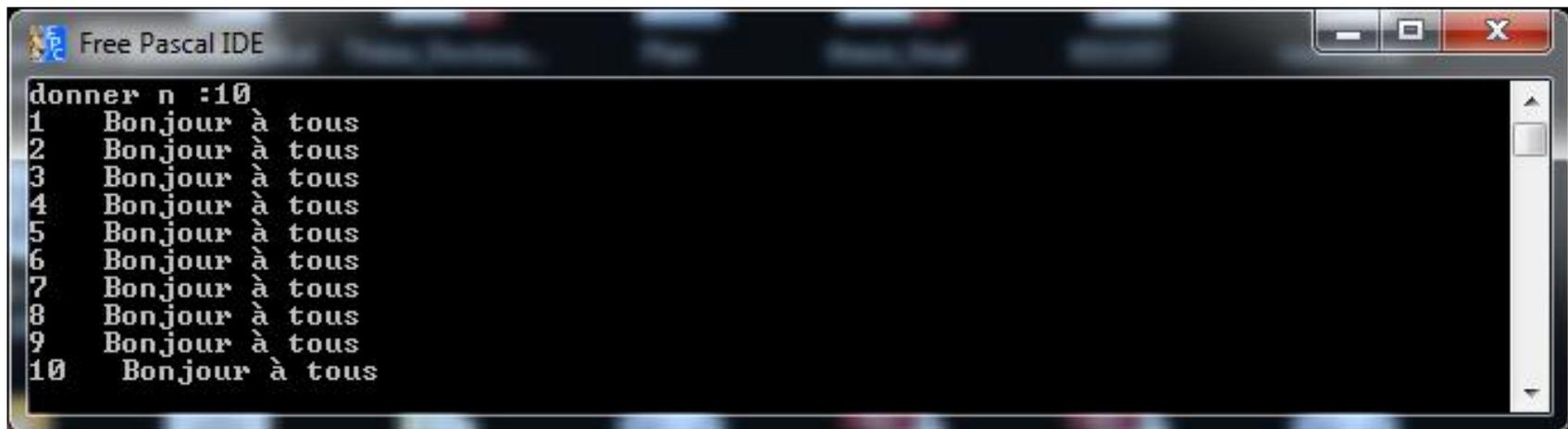
**Begin**

    Writeln(i, ' Bonjour à tous');

**End** ;

**End.**

A l'exécution on aura le résultat suivant pour n=10 :



```
Free Pascal IDE
donner n :10
1  Bonjour à tous
2  Bonjour à tous
3  Bonjour à tous
4  Bonjour à tous
5  Bonjour à tous
6  Bonjour à tous
7  Bonjour à tous
8  Bonjour à tous
9  Bonjour à tous
10 Bonjour à tous
```

**Exercice 1** : écrire un algorithme/programme pascal qui calcule la somme suivante :  
 $S=1+2+3+\dots+n$ ; n:entier

```
Algorithme somme
Variables i, n, s : entier
Debut
Ecrire ('donner n :') ;
Lire (n)
 $S \leftarrow 0$  ;
Pour  $i \leftarrow 1$  à n faire
 $s \leftarrow s+i$ 
finpour ;
ecrire ('somme =', s) ;
Fin.
```

```
Program somme;
Uses wincrt ;
Var i ,n, s : integer ;
Begin
Write('donner n :');
Read(n);
S:=0;
For i:= 1 to n do
Begin
S:=s+i ;
End ;
Write(' somme =', s);
End.
```

## 4.2.2 Boucle Tantque (while)

### Syntaxe

**Tantque** <condition> **faire**

inst1 ;

inst2 ;

...

**Fintantque** ;

**While** <condition> **do**

**Begin**

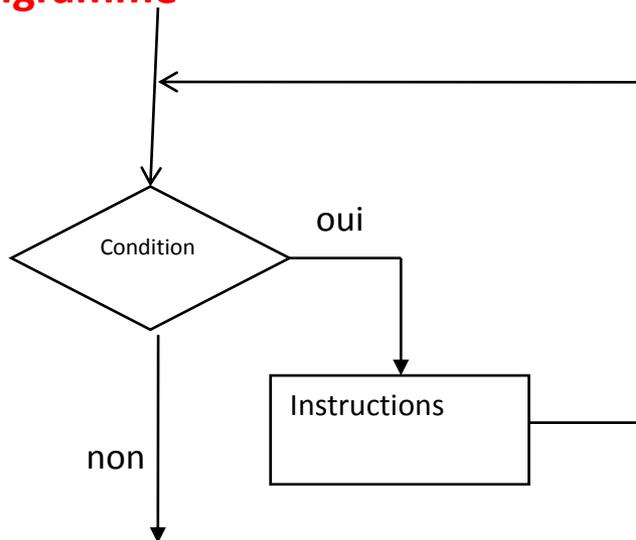
Inst1 ;

Inst2 ;

.....

**End** ;

### Organigramme



### Fonctionnement

1- On évalue la condition

2- Si la condition est vraie, on exécute les instructions puis on boucle et on répète **1** et **2** jusqu'à ce que la condition soit fausse.

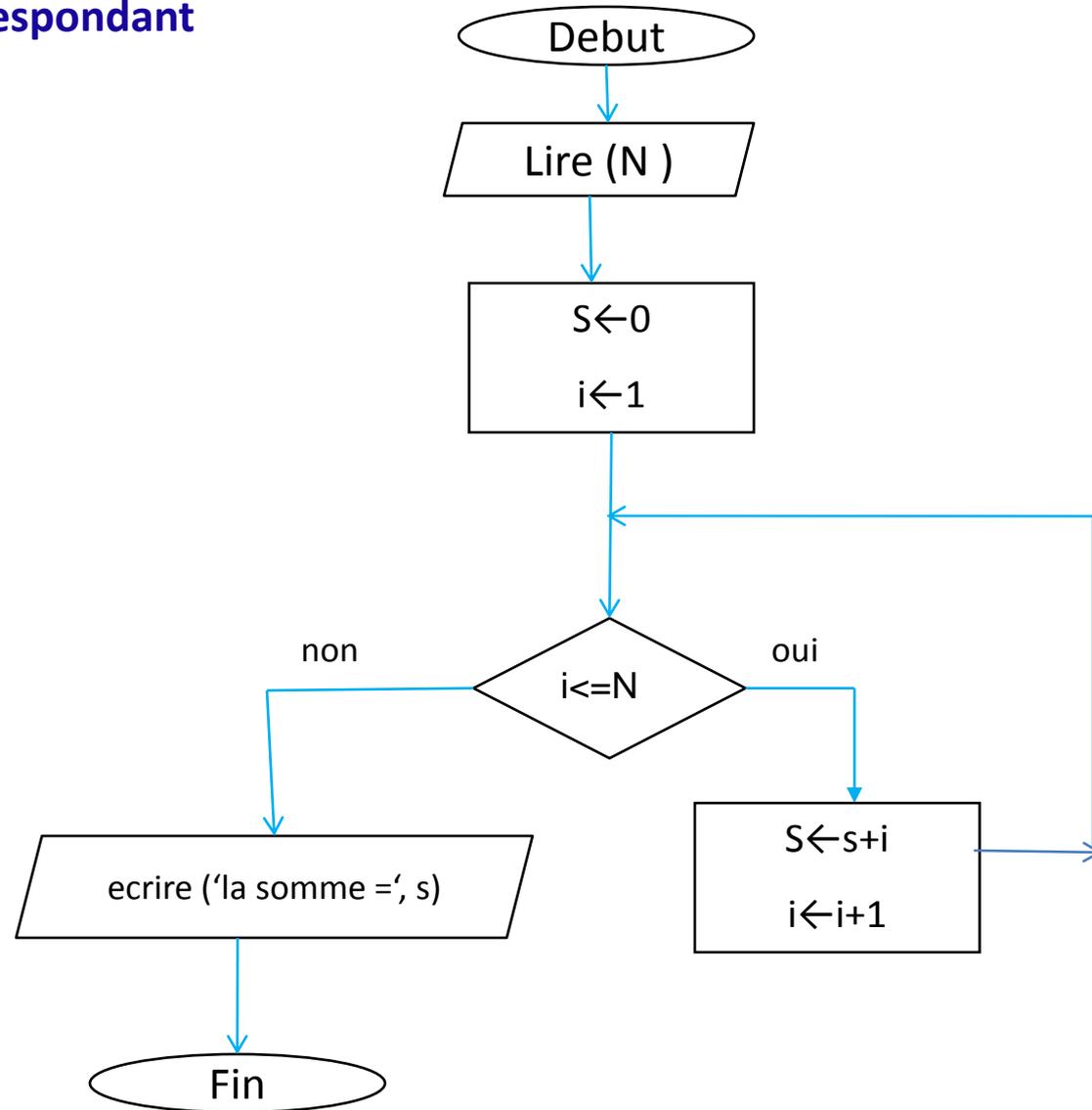
3- Si la condition est fausse, les instructions ne seront pas exécutées et la boucle sera terminée.

**Exercice 2** : écrire un algorithme/programme pascal qui calcule la somme suivante :  
 $S=1+2+3+\dots+n$ ; n:entier, en utilisant la boucle tantque

```
Algorithme somme
Variables i, n, s : entier
Debut
Ecrire ('donner n :')
  Lire (n)
 $S \leftarrow 0$ 
 $i \leftarrow 1$ 
Tantque  $i \leq n$  faire
   $s \leftarrow s+i$ 
   $i \leftarrow i+1$ 
fintantque;
ecrire ('somme =', s)
Fin.
```

```
Program somme;
Uses wincrt ;
Var i ,n, s : integer ;
Begin
Write('donner n :');
Read(n);
S:=0;
i:=1;
while i<= n do
Begin
S:=s+i ;
i:=i+1;
End ;
Write(' somme =', s);
End.
```

## Organigramme correspondant

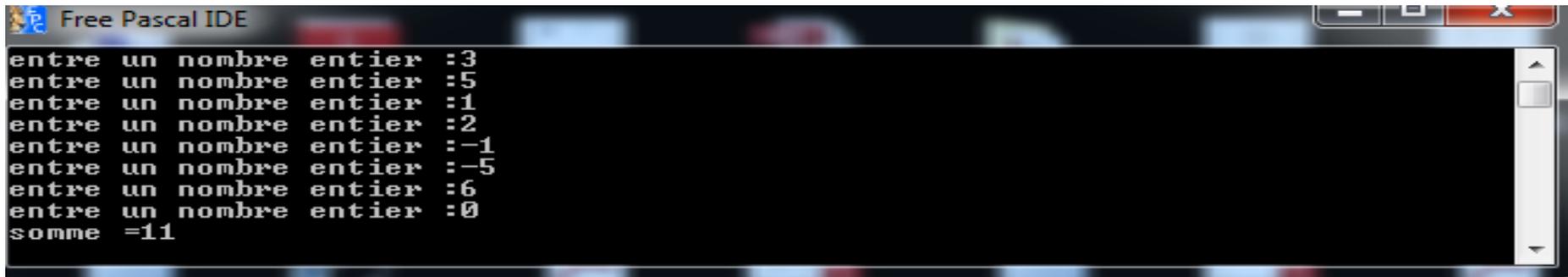


**Exercice 3** : écrire un algorithme/programme qui lit une suite de nombres entiers (on ne connaît pas leurs nombres) mais la lecture s'arrête dès qu'on introduit un zéro, puis qui calcule et affiche la somme des nombres lus.

```
Algorithme somme
Variables x, s : entier
Debut
S ← 0
Ecrire (' entrer un nombre entier : ')
Lire (x)
Tantque x≠0 faire
    S ← s+x
    Ecrire ('entrer un nombre entier : ')
    Lire(x)
fintantque;
ecrire ('somme =', s)
Fin.
```

```
Program somme;
Uses wincrt ;
Var x, s : integer ;
Begin
S:=0;
Write('entrer un nombre entier :');
Read(x);
while x<> 0 do
Begin
S:=s+x ;
Write('entrer un nombre entier :');
Read(x);
End ;
Write(' somme =', s);
End.
```

**Exécution :**



The screenshot shows the Free Pascal IDE window with the following output in the console:

```
entre un nombre entier :3
entre un nombre entier :5
entre un nombre entier :1
entre un nombre entier :2
entre un nombre entier :-1
entre un nombre entier :-5
entre un nombre entier :6
entre un nombre entier :0
somme =11
```

## 4.2.3 Boucle répéter (repeat)

### Syntaxe

#### Repetier

<Instruction (s)>

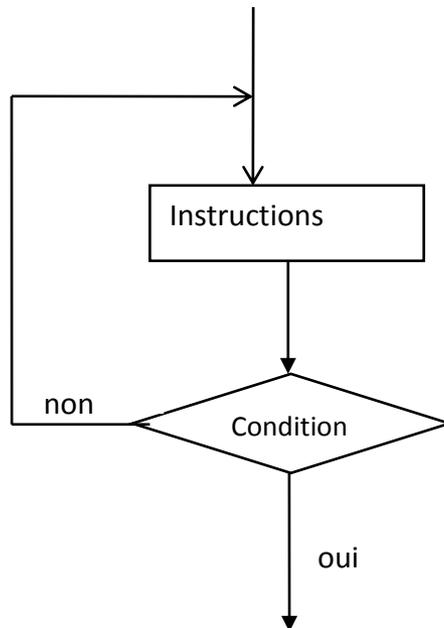
**Jusqu'à** < condition > ;

#### Repeat

<Instruction (s)> ;

**Until** <condition> ;

### Organigramme



### Fonctionnement

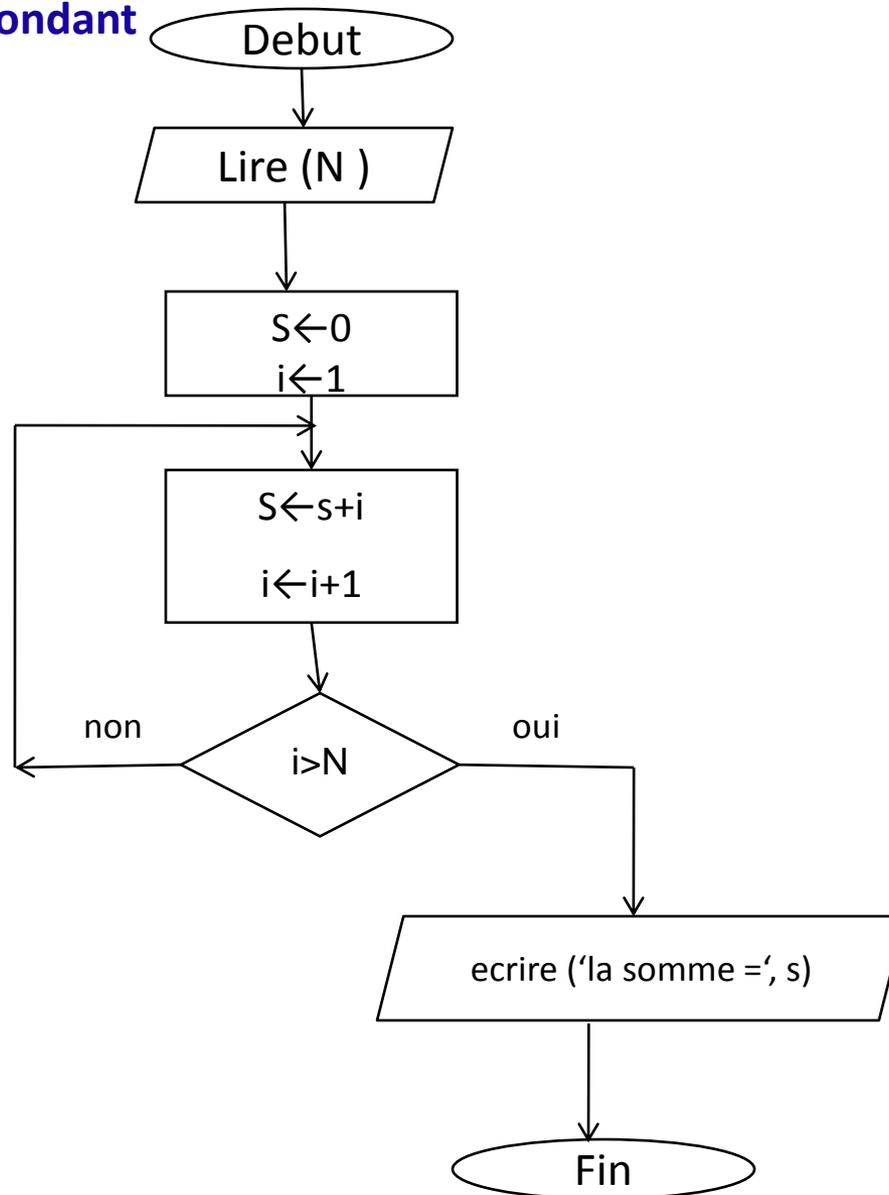
- 1- On exécute les instructions.
- 2- On évalue la condition, Si la condition est fausse, on répète **1** et **2** (on exécute les instructions puis on évalue la condition) jusqu'à ce que la condition soit vraie.
- 3- Si la condition est vraie, la boucle sera terminée.

**Exercice 4** : écrire un algorithme/programme pascal qui calcule la somme suivante :  
 $S=1+2+3+\dots+n$ ; n:entier, en utilisant la boucle repeter

```
Algorithme somme
Variables i, n, s : entier
Debut
Ecrire ('donner n :')
Lire (n)
 $S \leftarrow 0$ 
 $i \leftarrow 1$ 
repeter
   $s \leftarrow s+i$ 
   $i \leftarrow i+1$ 
jusqu'à  $i > n$ ;
ecrire ('somme =', s)
Fin.
```

```
Program somme;
Uses wincrt ;
Var i ,n, s : integer ;
Begin
Write('donner n :');
Read(n);
S:=0;
i:=1;
repeat
S:=s+i ;
i:=i+1;
until i>n;
Write(' somme =', s);
End.
```

## Organigramme correspondant



**Exercice 5** : Réécrire l'algorithme de l'exercice 3 en utilisant la boucle répéter.

```
Algorithme somme
Variables x, s : entier
Debut
S ← 0
Repete
  Lire ('entrer un nombre entier : ')
  Lire(x)
  S ← s+x
Jusqu'à x=0;
Ecrire ('somme =', s)
Fin.
```

```
Program somme;
Uses winCRT ;
Var x, s : integer ;
Begin
S:=0;
repeat
  Write('entrer un nombre entier :');
  Read(x);
  S:=s+x ;
Until x=0;
Write(' somme =', s);
End.
```