

# **Instructions itératives**

**NF01**

**Philippe TRIGANO**



# Plusieurs types de boucles

## ➤ **Boucle = Structure itérative**

- Itération = passage dans la boucle

## ➤ **Bornes définies**

- Incrémentation automatique du min au max

## ➤ **Bornes indéfinies**

- Au moins un passage dans la boucle
- Pas forcément d'itération



## Boucle à bornes définies

### ➤ Pour i variant de min à max ... fin\_pour

- Incrémentation ou décrémentation automatique
- En Pascal : **for**

**for** i:=<exp1> **to** <exp2> **do** <instruction>;

### ➤ Remarques

- La variable de boucle (ici i) doit être déclarée
- Elle doit être de type scalaire
  - ✓ C'est-à-dire entier, énuméré, intervalle ou caractère
- <instruction> peut être un bloc d'instructions
- Si **exp1** > **exp2**, <instruction> n'est pas exécutée

**On cherche à afficher à l'écran le carré  
des 5 premiers entiers :**

```
program carre;  
var  
    i : integer;  
begin  
    for i:=1 to 5 do  
        writeln('le carré de ', i, ' est :', i*i:3);  
end.
```

Exécution => le carré de 1 est : 1  
le carré de 2 est : 4  
le carré de 3 est : 9  
le carré de 4 est : 16  
le carré de 5 est : 25

# Portée

```
somme:=0;  
for i:=1 to 4 do  
    somme:=somme+sqr(i);  
writeln(somme);
```

## Exécution

**30**

```
somme:=0;  
for i:=1 to 4 do  
begin  
    somme:= somme+sqr(i);  
    writeln(somme);  
end;
```

## Exécution

**1**

**5**

**14**

**30**

# Avec des caractères

## Exemple :

```
for c:= 'a' to 'z' do write(c);
```

## Exécution ?

abcefg hijklmnopqrstuvwxyz



# Boucles à bornes indéfinies

- **Tant que** <expression> **faire**  
    <instruction>  
    **Fin\_tant\_que**

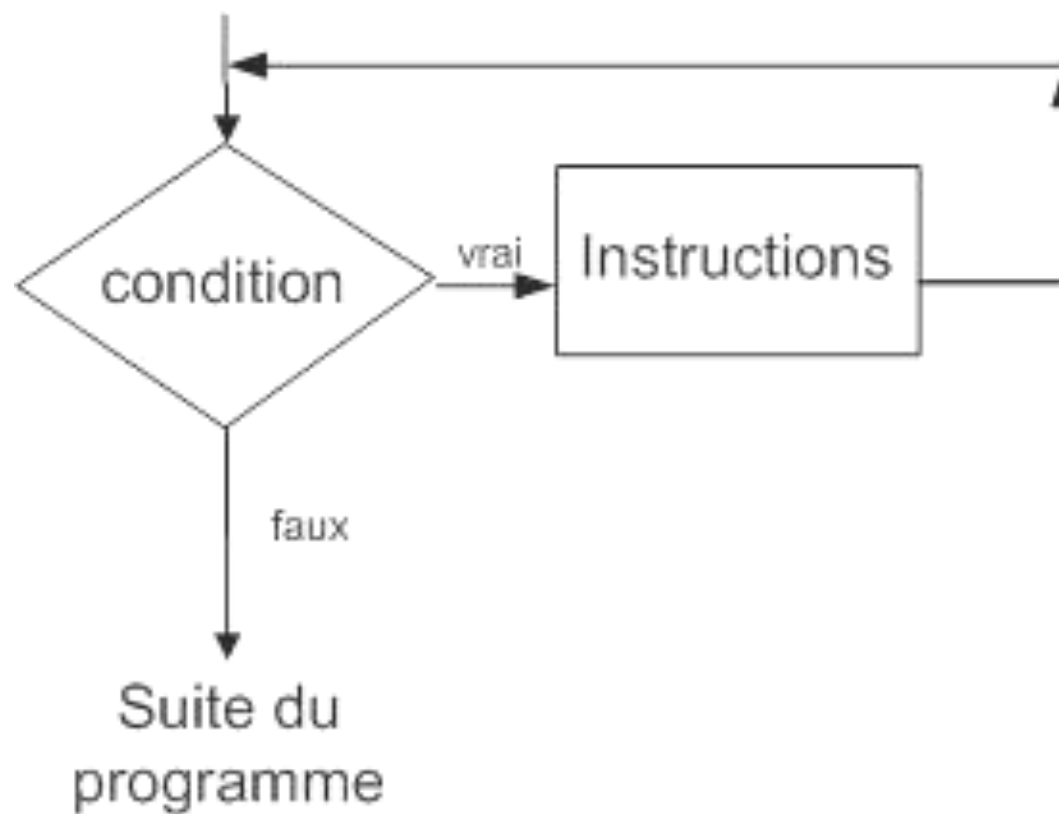
**while** <condition> **do** <instruction\_simple>;

**while** <condition> **do**  
begin  
    ...  
**end;**

- **Répéter**  
    <instruction>  
    **jusqu'à** <expression>

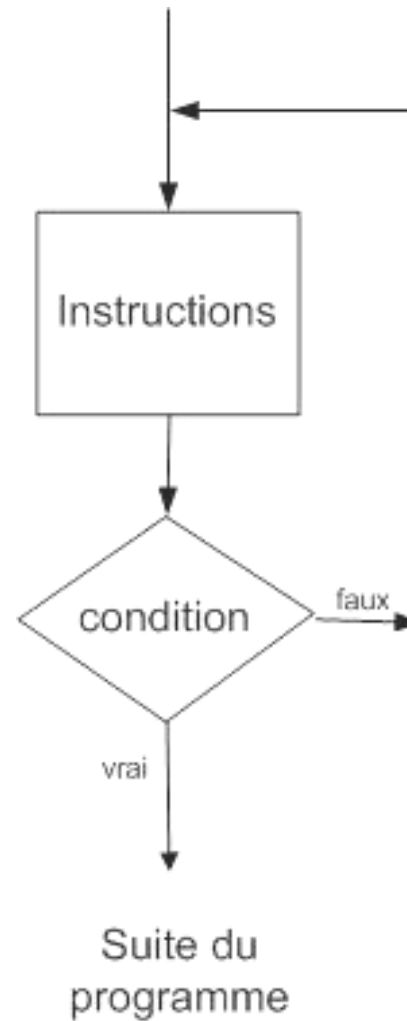
**repeat**  
    ...  
    ...  
**until** <condition>

# Tant que





# Répéter ... jusqu'à



# Exemple

- **Comment afficher le carré des n premiers entiers avec une boucle Tant que ou Répéter ... jusqu'à ?**

le carré de 1 est : 1  
le carré de 2 est : 4  
le carré de 3 est : 9  
le carré de 4 est : 16  
le carré de 5 est : 25

- **Algorithme ?**



```
program boucle_while;  
const  
    N=5;  
var  
    i: integer;  
  
begin  
    i:=1;  
    while i <= N do  
        begin  
            writeln('le carré de ', i, ' est :', sqr(i));  
            i:=i+1;  { incrémentation gérée par le programmeur }  
        end;  
    end.
```

Remarque : begin ... end obligatoire si plusieurs instructions

```
program boucle_repeat;  
const N=5;  
var  
    i: integer;  
  
begin  
    i:=1;  
    repeat  
        writeln('le carré de ', i, ' est :', sqr(i));  
        i:=i+1; { incrémentation gérée par le programmeur }  
    until (i > N);  
end.
```

Remarque : la boucle se termine quand la valeur de l'expression devient *true*



- Ecrire un programme qui calcule la factorielle d'un nombre  $n$  tapé au clavier par l'utilisateur.

```
program factorielle;  
{ calcul de n! }  
var  
    fact, n, i : integer;  
  
begin  
    readln(n);  
    fact:=1  
    for i:=1 to n do  
        fact:=fact*i;  
    writeln('factorielle ', n, ' =', fact);  
end.
```

# Imbrication

- **Exercice : Comment afficher une matrice unité ?**

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

# Imbrication

```
program unite;  
const  
    N=8;  
var  
    i:integer;  
begin  
    for i:=1 to N do  
        begin  
            for j:=1 to N do  
                if j=i then write('1')  
                    else write('0');  
            writeln;  
        end;  
    end;  
end.
```