

ALGORITHME

Table des matières

I.	Definition	2
II.	Types de boucles	2
1.	La structure « pour »	2
2.	La structure « tant que »	3

I. **Definition**

Une boucle est une structure de l'algorithmique qui permet de répéter un traitement plusieurs fois pour une même données.

Exemple :

Ecrire un algorithme qui permet d'afficher le mot "bonjour" 50 fois.

50 : nombre de répétition.

Le traitement à répéter le mot "Bonjour".

Remarque : on dit le nombre de répétition ou bien le nombre d'itérations (qui signifié le passage d'un pas à l'autre dans une boucle).

II. **Types de boucles**

On distingue deux types de boucles : La structure « pour » et la structure « tant que »

1. La structure « pour »

Propriétés

- On utilise la structure « pour » quand le nombre d'itération est connu à l'avance.
- Le compteur est initialisé à 1.
- Le pas d'incrémentation =1

Syntaxe :

Pour compteur allant de (valeur initiale) à (valeur finale) faire

Σ Instructions

Fin pour

Exemple :

Pour i allant de 1 à 3 faire

Ecrire ("bonjour")

Fin pour

En pascal

```
For i :=1 to 3 do  
Begin  
Writeln('bonjour');  
End;
```

2. La structure « tant que »

Contrairement à la boucle « pour », la structure « tant que » permettent de faire des itérations tant que la condition est vérifiée.

Propriétés

Le nombre d'itérations qui n'est pas connu à l'avance, permet de vérifier si la condition est vraie pour exécuter le bloc d'instructions, si la condition est fausse on sort de la boucle.

Syntaxe

```
Tant que (condition) faire  
    ∑ Instructions  
Fin tant que
```

Exemple

```
S ← 0  
Tant que ( i<=5 )faire  
    S ← S+i ← i ← i+1  
Fin tant que  
Ecrire(s)
```

Cet algorithme s'arrête dès que le compteur $i > 5$

En Pascal:

```
S:=0;  
While (i<=5) do
```

```
Begin S: =S+1; i:=i+1; end;
```

```
Writeln('la somme=',S);
```

Remarque : Si on connaît le nombre de répétition à traiter on utilise la boucle « pour » mais si on connaît la condition mais on ne connaît pas le nombre de répétition on utilise la boucle « tant que »