



LES STRUCTURES ITÉRATIVES





Table des matières

Objectifs	5
I - Les structures itératives	7
A. Pré requis.....	7
B. Les structures itératives.....	7
1.La boucle « POUR ».....	7
2.La boucle « TANTQUE ».....	8
3.La boucle « REPETER ».....	9



Objectifs

Ce cours a pour objectif de vous permettre de :

- Comprendre l'utilité d'une structure itérative.
- Connaître les différentes structures itératives disponibles en algorithmique.
- Résoudre des problèmes nécessitant l'utilisation des structures itératives.

Les structures itératives

Pré requis	7
Les structures itératives	7

A. Pré requis

Avant de suivre cette leçon, vous devez avoir pour acquis

- La connaissance de la structuration d'un algorithme et la connaissance des cinq propriétés d'un algorithme de la leçon 1.
- Connaître les différents opérateurs et les instructions existants de la leçon 2.
- Connaître les différentes structures conditionnelles de la leçon 3.

B. Les structures itératives



Définition

Les structures itératives ou de répétitions ou boucles vont permettre de répéter une instruction un certain nombre de fois, sous certaines conditions.

Elles permettent d'effectuer une action autant de fois que nécessaire.

On distingue à cet effet trois structures de répétitions à savoir :

- **POUR**
- **TANTQUE**
- **REPETER**

1. La boucle « POUR »

La boucle « POUR » permet de répéter une instruction un nombre donné de fois jusqu'à un maximum. Elle se caractérise par le fait que l'on doit connaître à l'avance le nombre d'itérations que l'on va devoir effectuer.



Syntaxe

POUR *i* ← val-début **A** val-arrivé **Faire**

Instruction

Finpour



Remarque

- La boucle « POUR » commence par le mot réservé **POUR** et s'achève par un **finpour**.
- La variable **i** est appelé variable de la boucle et commence par une valeur minimum (val-début) à une valeur maximum (val-arrivé).
- La variation ou l'incrémentation de **i** (c'est-à-dire la valeur suivante de **i**) s'effectue automatiquement par pas de 1 (c'est-à-dire à chaque pas on fait **i = i+1**).



Exemple

Ecrire un algorithme qui affiche les 20 premiers entiers positifs.

Algorithme Nombreentiers

Var

I : entier

Début

POUR *i* ← 1 à 20 faire

Afficher *i*

Finpour

Fin

Cette boucle affichera successivement les nombres 1, ...,20, on effectue donc 20 itérations.

2. La boucle « TANTQUE »

La boucle « **TANTQUE** » permet d'effectuer des itérations sous certaines conditions, avant d'effectuer une itération la boucle vérifie si la condition est vraie. Lorsque la condition est fausse alors la boucle passe la main.



Syntaxe

Initialisation

TANTQUE Condition **Faire**

Instruction

variation

Fintantque



Remarque

- La boucle « TANTQUE » commence par l'**initialisation** de la variable qui servira à poser la condition et s'achève par un **Fintantque**.
- L'initialisation consiste à attribué une valeur de départ à une variable (elle peut se faire par affectation ou par saisie).
- La condition est celle qui permet si elle est vraie de faire une itération.

- La variation consiste à aller à la valeur suivante de la variable initialisée
- La boucle « TANTQUE » sans variation devient une **boucle infinie**.



Exemple

Ecrire un algorithme qui affiche les 20 premiers entiers positifs.

Algorithme Nombreentiers

Var

I : entier

Début

I ← 1 // Initialisation

TANTQUE I ≤ 20 **Faire**

Afficher I

I ← I + 1 // valeur suivante de I

Fintantque

Fin

Cette boucle affichera successivement les nombres 1, ..., 20, on effectue donc 20 itérations.

3. La boucle « REPETER »

La boucle « REPETER » permet d'effectuer au moins une itération avant de vérifier la condition et elle continue l'itération jusqu'à ce que la condition soit vraie avant de sortir de la boucle.

Syntaxe

REPETER

Instruction

JUSQU'A Condition

Remarque

- La boucle « REPETER » commence par le mot réservé REPETER et s'achève par un JUSQU'A.
- L'instruction est exécutée au moins une fois

Exemple

Ecrire un algorithme qui permet de saisir un nombre entier positif et ceux jusqu'à ce que la condition soit vérifiée.

Algorithme Nombrepositif

Var

Nbre : entier

Début

REPETER

Afficher "Entrer un nombre entier positif"

Saisir Nbre
JUSQUA Nbre >0
Fin

