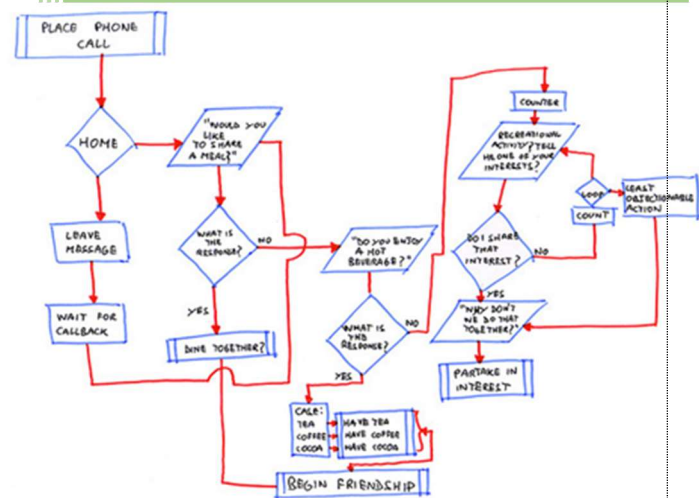


# Cours : Les modèles algorithmiques



## 1. DÉFINITION

Un algorithme est un ensemble de règles opératoires rigoureuses, ordonnant à un processeur d'exécuter dans un ordre déterminé une succession d'opérations élémentaires, pour résoudre un problème donné.

C'est un outil méthodologique général qui ne doit pas être confondu avec le programme proprement dit.

Un algorithme peut être :

- Représenté graphiquement par un **organigramme de programmation** (ou ordinogramme),
- Écrit sous forme littérale avec un **langage algorithmique**.

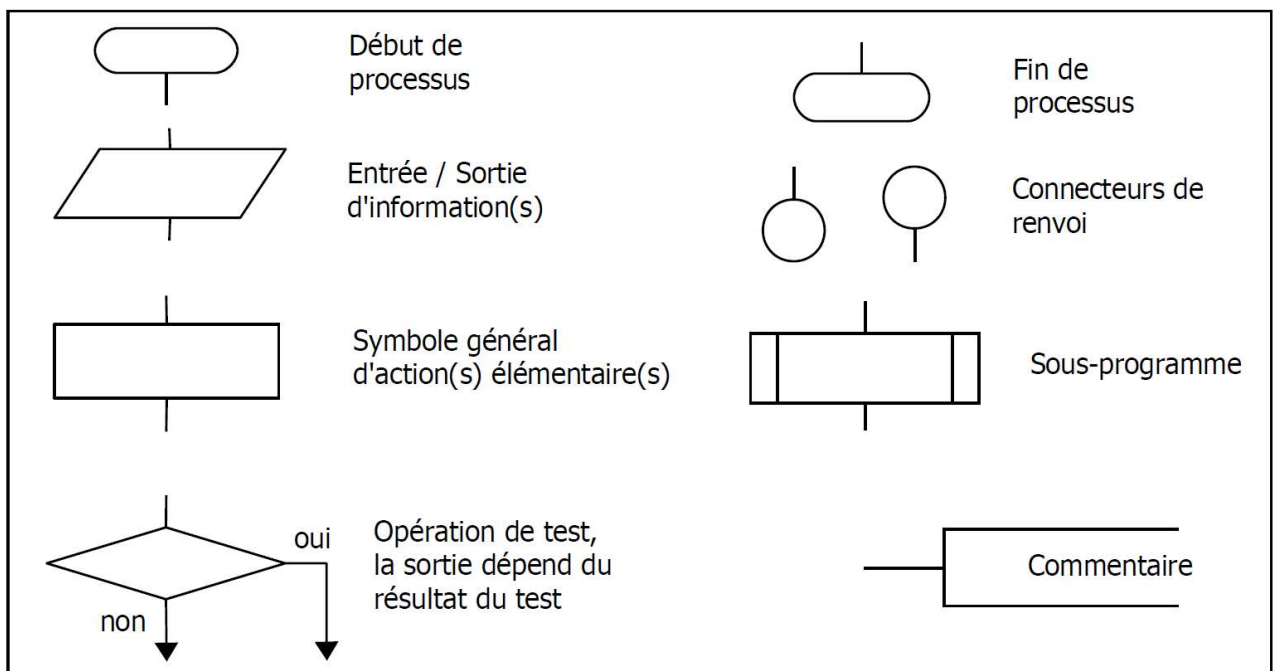
## 2. ORGANIGRAMME

L'organigramme est une représentation graphique normalisée utilisée pour analyser ou décoder un problème. Il utilise des symboles représentant des traitements, des liaisons, des données...

Un organigramme bien représenté doit être fléché et fermé, compris entre un début et une fin, afin d'indiquer l'ordre de déroulement des actions élémentaires.

Par convention, le déroulement d'un organigramme (algorithme, ordinogramme...) se fait du haut vers le bas, si cela n'est pas respecté, il faut flécher le sens du déroulement.

Les principaux symboles normalisés :



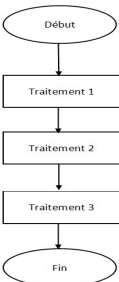
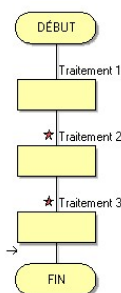
## 3. STRUCTURES ALGORITHMIQUES FONDAMENTALES

Les opérations relatives à la résolution d'un problème peuvent en fonction de leur enchaînement, être organisées selon trois familles de structures :

- Structures linéaires
- Structures alternatives
- Structures répétitives.

### 3.1. LA STRUCTURE LINÉAIRE OU SÉQUENCE

La structure linéaire se caractérise par une suite d'actions à exécuter successivement dans l'ordre de leur énoncé.

Organigramme	Langage algorithmique	Flowcode
	Début Traitement 1 Traitement 2 Traitement 3 Fin	

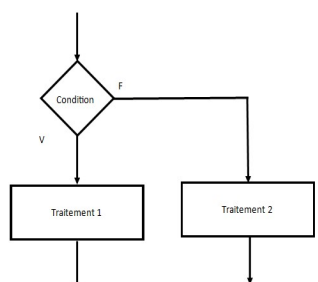
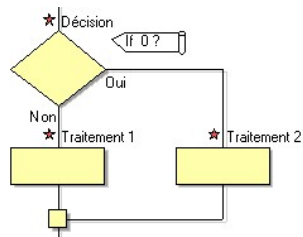
### 3.2. LA STRUCTURE ALTERNATIVE OU SÉLECTION

Une structure alternative n'offre que deux issues possibles s'excluant mutuellement.

Les structures alternatives définissent une **fonction de choix** ou de **sélection** entre l'exécution de l'un ou de l'autre des deux traitements.

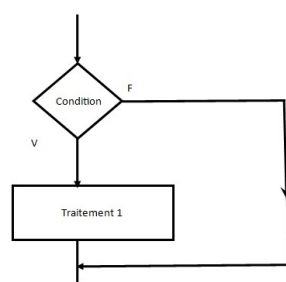
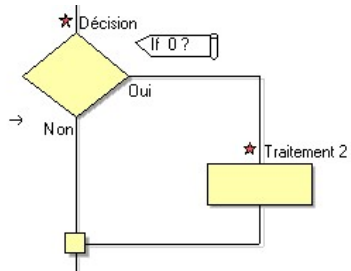
Également désignées par **structures conditionnelles**, elles sont représentatives du **saut** ou rupture de séquence.

#### 3.2.1. LA STRUCTURE ALTERNATIVE COMPLÈTE

Organigramme	Langage algorithmique	Flowcode
	<b>Si</b> (Condition) <b>alors</b> Traitement 1 <b>Sinon</b> Traitement 2 <b>Fin Si</b>	

#### 3.2.2. LA STRUCTURE ALTERNATIVE RÉDUITE

La structure alternative réduite se distingue de la précédente par le fait que seule la situation correspondant à la validation de la condition entraîne l'exécution du traitement. La situation opposée conduit à la sortie de la structure.

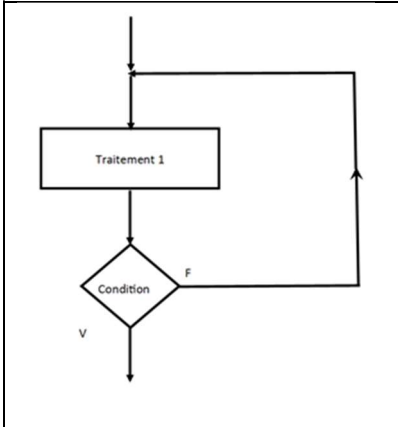
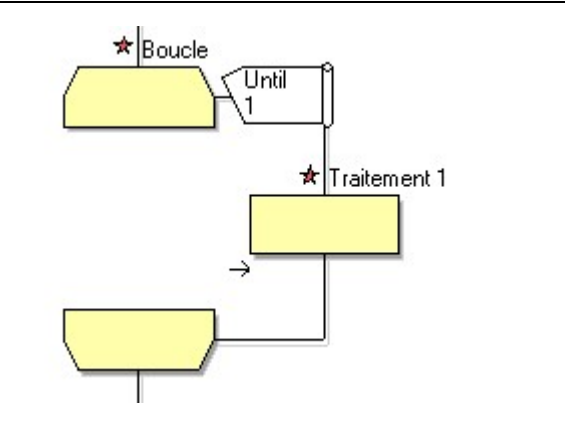
Organigramme	Langage algorithmique	Flowcode
	<b>Si</b> (Condition) <b>alors</b> Traitement 1 <b>Fin Si</b>	

### 3.3. LES STRUCTURES RÉPÉTITIVES

Une structure répétitive ou itérative répète l'exécution d'un traitement.

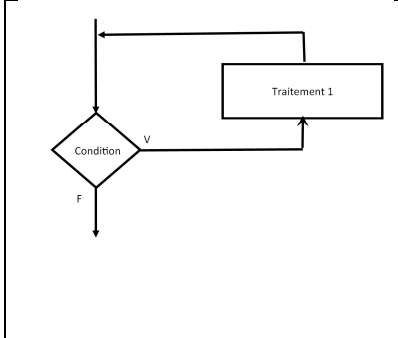
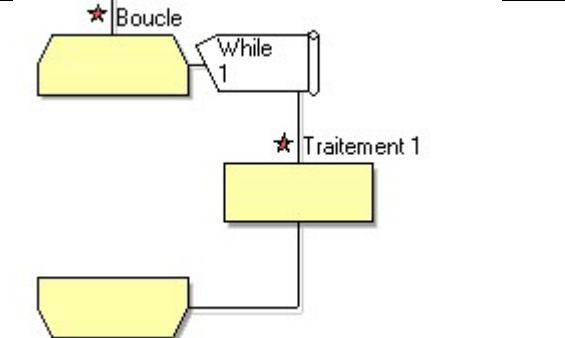
#### 3.3.1. REPETER ... JUSQU'À ...

Dans cette structure le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée.

Organigramme	Langage algorithmique	Flowcode
	<b>Répéter</b>  Traitement 1  <b>Jusqu'à ce que (Condition)</b>	

#### 3.3.2. TANT QUE ... FAIRE ...

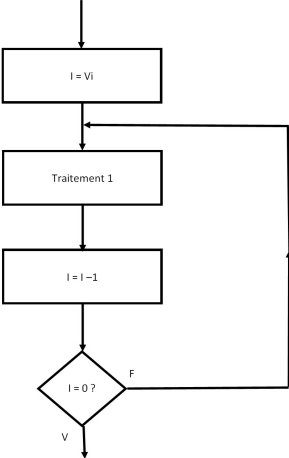
Dans cette structure on vérifie d'abord la condition avant d'exécuter le traitement. Le traitement sera répété tant que la condition est vraie

Organigramme	Langage algorithmique	Flowcode
	<b>Tant que (Condition) faire</b>  Traitement 1  <b>Fin Tant que</b>	

### 3.4. STRUCTURE RÉPÉTITION CONTRÔLÉE

Dans cette structure la sortie de la boucle d'itération s'effectue lorsque le nombre souhaité de répétitions est atteint. D'où l'emploi d'une variable de boucle (indice I) caractérisée par :

- Sa valeur initiale,
- Sa valeur finale,
- Son pas de variation

Organigramme	Langage algorithmique	Flowcode
	<pre><b>Pour</b> I = Vi jusqu'à 0 <b>faire</b>     Traitement 1 <b>Fin Pour</b></pre>	