

# **Algorithmique :**

## **Langage**

## Table des matières

<u>1. Qualité d'un bon algorithme</u> .....	3
<u>2. Éléments représentatifs du langage algorithmique</u> .....	3
a) Les mots.....	3
<u>Les mots clés</u> .....	3
b) Mots instructions.....	4
c) Mots délimiteurs.....	4
<u>3. Représentation des algorithmes</u> .....	4
a) Représentation littérale.....	4
b) Représentation graphique ou algorithmique.....	5
<u>4. Structures algorithmiques fondamentales</u> .....	6
a) Structure linéaire ou séquentielle.....	6
b) Structures itératives.....	6
<u>Structure « REPETER... JUSQU'A... »</u> .....	7
<u>Structure « REPETER... TANT QUE... »</u> .....	8
<u>Structure « POUR... A... REPETER... »</u> .....	9
c) Structures alternatives.....	10
<u>Structure alternatives complète</u> .....	10
<u>Structure alternatives réduite</u> .....	11
<u>5. Structures emboîtées</u> .....	13
<u>Annexes</u> .....	14
<u>6. Index des illustrations</u> .....	14
<u>7. Index des exemples</u> .....	14

# 1. Qualité d'un bon algorithme

Pour obtenir un bon programme il faut partir d'un bon algorithme qui doit posséder entre autres, les qualités suivantes :

- être clair, facile à comprendre par tous ceux qui le lisent ;
- être le plus général possible pour répondre au plus grand nombre de cas possibles ;
- être d'une utilisation aisée même par ceux qui ne l'ont pas écrit (Messages pour l'introduction des données...) ;
- être conçu de manière à limiter le nombre d'opérations à effectuer et la place occupée en mémoire.

Une des meilleures façons de rendre un algorithme clair et compréhensible est d'utiliser un langage de description structuré n'utilisant qu'un petit nombre de structures indépendantes du langage de programmation utilisé.

## 2. Éléments représentatifs du langage algorithmique

### a) *Les mots*

On distingue trois familles de mots :

- les mots clés ;
- les mots instructions ;
- les mots délimiteurs.

#### Les mots clés

Les mots clés définissent la structure algorithmique utilisée et en voici quelques exemples :

- SI ... ALORS... SINON... : définissent un **structure alternative** ;
- REPETER... JUSQU'A... : définissent une **structure itérative**.

Un mot clé est toujours suivi :

- soit d'une expression conditionnelle écrite entre guillemets ;
- soit d'un ou plusieurs mots instructions.

```
SI « condition 1 »
  ALORS
    mot instruction 1
    mot instruction 2
  SINON
    REPETER
      mot instruction 3
      mot instruction 4
    JUSQU'A « condition 2 »
FIN SI
```

*Exemple 1: Exemple de structure algorithmique*

### *b) Mots instructions*

Se sont des verbes d'action qui caractérisent la nature des opérations à effectuer sur une ou plusieurs données.

Un mot instruction est toujours suivi entre guillemets :

- de la désignation de l'objet sur lequel il s'applique comme, par exemple, LIRE « Capteur S1 » ;
- éventuellement de la description de l'opération à appliquer à l'objet comme, par exemple, FAIRE « Compteur = Compteur - 1 ».

### *c) Mots délimiteurs*

Les mots délimiteurs fixent :

- les bornes d'ENTRÉE et de SORTIE de l'algorithme ;
- les bornes d'ENTRÉE et de SORTIE des différentes structures utilisées dans l'algorithme si ces bornes ne sont pas définies par la structure elle même.

DEBUT et FIN sont les seuls mots délimiteurs et peuvent être suivi éventuellement d'un mot clé comme, par exemple, FIN SI

## **3. Représentation des algorithmes**

### *a) Représentation littérale*

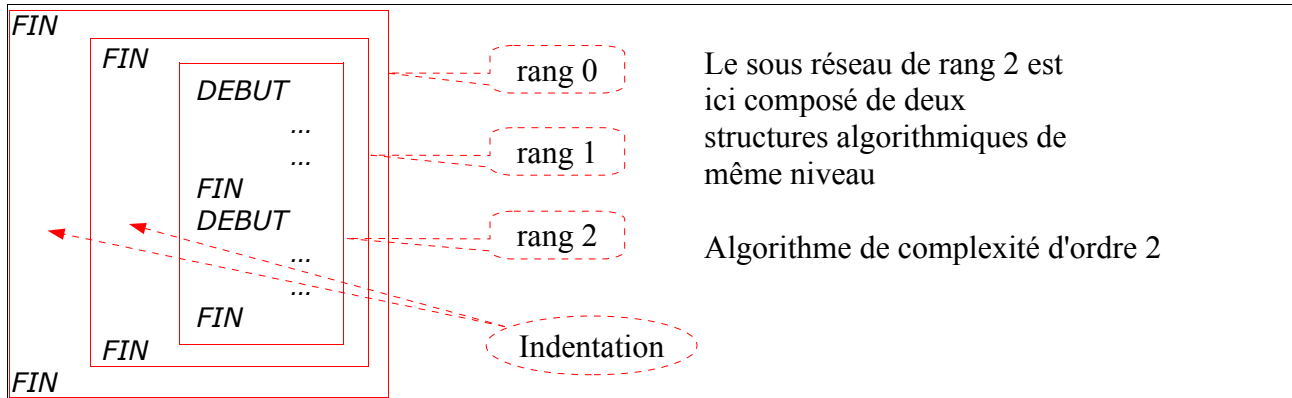
Aux règles précédemment définies nous ajouterons les règles d'écriture suivantes afin de faciliter la lecture de l'algorithme :

**L'écriture sera indentée** afin de faire apparaître l'algorithme comme un réseau principal **comportant** une **borne d'entrée** et une **borne de sortie** et chacune des structures qui le constitue comme un sous réseau présentant aussi une entrée et une sortie.

Règles :

- le dernier sous réseau ouvert doit être le premier fermé ;
- le nombre de fermetures doit être égal au nombre d'ouvertures ;

Le rang de la dernière paire de mots délimiteurs représente le degré de complexité de l'algorithme.



Exemple 2: Algorithme de complexité d'ordre 2

### b) Représentation graphique ou algorithmique

La représentation graphique permet une lecture aisée des algorithmes mais présente toutefois l'inconvénient de consommer une place importante. Elle utilise les symboles de la norme **NF Z 67-010** dont les principaux sont les suivants :

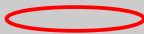
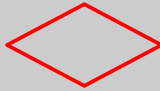






	Début, fin ou interruption d'un organigramme		Embranchement, représente une décision ou un aiguillage après un test
	Renvoi, utilisé deux fois pour assurer la continuité d'un organigramme scindé		Traitement, opération ou groupe d'opérations sur des données
	Commentaire, permet de donner des indications marginales		Entrée / Sortie, mise à disposition d'une information ou enregistrement d'une information
	Liaison, le sens doit être : <ul style="list-style-type: none"> <li>• de haut en bas ;</li> <li>• de gauche à droite.</li> </ul>		Procédé prédéfini, portion de programme considéré comme une simple opération

Illustration 1: Représentation algorithmique

## 4. Structures algorithmiques fondamentales

### a) Structure linéaire ou séquentielle

C'est une **suite d'actions** à exécuter successivement dans l'ordre de leur énoncé

Algorithme littérale	Algorithme graphique
Instruction 1 ; Instruction 2 ... Instruction n	

Exemple 3: Structure linéaire

Les mots instructions sont écrits au même rang sur des lignes successives.

Algorithme littérale	Algorithme graphique
DEBUT FAIRE « AC=1 » FAIRE « AC=2 » OUVRIR « vanne » FIN	

Exemple 4: Exemple de structure linéaire

### b) Structures itératives

Par itération on désigne toute répétition de l'exécution d'un traitement. Trois types de structures itératives sont à distinguer:

- la structure REPETER... JUSQU'A... ;
- la structure REPETER... TANT QUE... ;
- la structure POUR... A... REPETER...

Dans les deux premier cas, le nombre de répétitions n'est pas connu à l'avance et dépend d'un ou plusieurs événements extérieurs.

Dans le dernier cas, le nombre de répétitions est connu à l'avance, il est consigné.

*Structure « REPETER... JUSQU'A... »*

Algorithme littérale	Algorithme graphique
<p><b>REPETER</b></p> <p>Instruction 1</p> <p>...</p> <p>Instruction n</p> <p><b>JUSQU'A</b> « condition vrai »</p>	

Exemple 5: Structure « REPETER... JUSQU'A... »

En anglais : REPEAT... UNTIL

Le traitement est exécuté une première fois dès l'entrée dans la structure, il se répète jusqu'à ce que la condition soit vérifiée.

**Exemple**

A l'apparition d'un défaut on provoque simultanément la mise sous tension :

- d'un signal lumineux H1 ;
- d'un avertisseur sonore H2.

L'arrêt de l'avertisseur sonore se fait lorsque l'opérateur acquitte le défaut par une action sur le bouton poussoir S1

Algorithme littérale	Algorithme graphique
<p>DEBUT</p> <p>FAIRE « H1=1 »</p> <p><b>REPETER</b></p> <p>FAIRE « H2=1 »</p> <p><b>JUSQU'A</b> « S1=1 »</p> <p>FAIRE « H2=0 »</p> <p>FIN</p>	

Exemple 6: Exemple de structure « REPETER... JUSQU'A... »

**Structure « REPETER... TANT QUE... »**

Algorithme littérale	Algorithme graphique
<p><b>REPETER</b></p> <p>Instruction 1</p> <p>...</p> <p>Instruction n</p> <p><b>TANT QUE</b> « condition vrai »</p>	

Exemple 7: Structure « REPETER... TANT QUE... »

En anglais: REPEAT... WHILE...

Le traitement est exécuté une première fois dès l'entrée dans la structure, il se répète tant que la condition est vérifiée.

Les deux structures itératives « REPETER... JUSQU'A... » et « REPETER... TANT QUE... » sont strictement équivalentes.

**Exemple**

A l'apparition d'un défaut on provoque simultanément la mise sous tension :

- d'un signal lumineux H1 ;
- d'un avertisseur sonore H2.

L'arrêt de l'avertisseur sonore se fait lorsque l'opérateur acquitte le défaut par une action sur le bouton poussoir S1

Algorithme littérale	Algorithme graphique
<p>DEBUT</p> <p>FAIRE « H1=1 »</p> <p><b>REPETER</b></p> <p>FAIRE « H2=1 »</p> <p><b>TANT QUE</b> « S1=0 »</p> <p>FAIRE « H2=0 »</p> <p>FIN</p>	

Exemple 8: Exemple de structure « REPETER... TANT QUE... »



Structure « POUR... A... REPETER... »

Algorithme littérale	Algorithme graphique
<p><b>POUR</b> <math>V=V_i</math> <b>A</b> <math>V=V_f</math> <b>PAS</b>=<math>P</math>  <b>REPETER</b>              Instruction 1              ...              Instruction n  <b>FIN POUR</b></p>	

Exemple 9: Structure « POUR... A... REPETER »

En anglais: FOR... TO... STEP... NEXT...

Dans cette structure la sortie de la boucle d'itération s'effectue lorsque le nombre de répétitions est atteint.

D'où l'emploi d'une variable  $V$  dite de contrôle d'itération et définie par :

- sa **valeur initiale** :  $V_i$
- sa **valeur finale** :  $V_f$
- son **pas de variation** :  $P$

Cette structure s'utilise si le nombre  $NR$  de répétitions est connu :

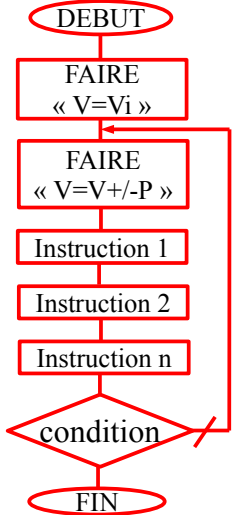
$$NR = \frac{|V_f - V_i|}{|P|}$$

Exemple :

- Pour  $V_f=5$  ;  $V_i=5$  ;  $P=+1$  :  $NR=5$
- Pour  $V_f=0$  ;  $V_i=8$  ;  $P=-2$  :  $NR=4$

### Exemple

Le guichetier d'une salle de cinéma tape sur son clavier le nombre de billets demandé par le client. Le distributeur déroule alors les billets en sectionne la bande après le dernier billet :

Algorithme littérale	Algorithme graphique
<p>DEBUT</p> <p>TAPER « Nombre de billets NB »</p> <p><b>POUR</b> <math>V=0</math> <b>A</b> <math>V=NB</math></p> <p><b>REPETER</b></p> <p style="padding-left: 40px;">DELIVRER « 1 billet »</p> <p><b>FIN POUR</b></p> <p>COUPER « Bande »</p> <p>FIN</p>	

Exemple 10: Exemple de structure « POUR... A... REPETER »

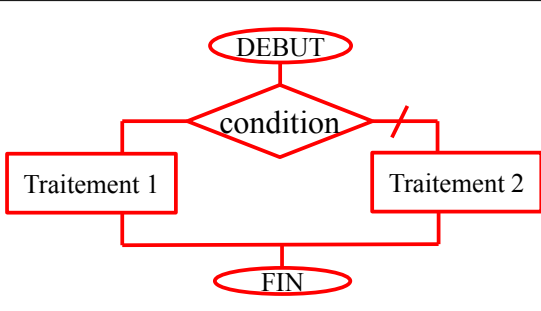
### c) Structures alternatives

Ces structures désignent toute situation n'offrant que deux issues possibles s'excluant mutuellement.

Il existe deux types de structures alternatives :

- la structure alternative complète ;
- la structure alternative réduite.

#### Structure alternatives complète

Algorithme littérale	Algorithme graphique
<p><b>SI</b> « condition vrai »</p> <p><b>ALORS</b></p> <p style="padding-left: 40px;">traitement 1</p> <p><b>SINON</b></p> <p style="padding-left: 40px;">traitement 2</p> <p><b>FIN SI</b></p>	

Exemple 11: Structure « SI... ALORS... SINON »

En anglais : IF... THEN... ELSE...

L'exécution d'un des deux traitements dépend du résultat d'un test :

- si le test est **VRAI** le premier traitement est exécuté.
- si le test est **FAUX** c'est le deuxième traitement qui s'effectue.

**EXEMPLE**

Sur une chaîne de conditionnement un dispositif de tri permet de diriger les caisses de masse supérieure ou égale à 20Kg vers le tapis 1 et les autres vers le tapis 2 en comptabilisant le nombre de caisses.

Algorithme littérale	Algorithme graphique
<p>DEBUT</p> <p>PESER "Caisse"</p> <p><b>SI</b> "P&gt;=20Kg"</p> <p><b>ALORS</b></p> <p>AIGILLER « vers tapis 1 »</p> <p>FAIRE « C1=C1+1 »</p> <p><b>SINON</b></p> <p>AIGILLER « vers tapis 2 »</p> <p>FAIRE « C2=C2+1 »</p> <p><b>FIN SI</b></p> <p>FIN</p>	<pre> graph TD     DEBUT([DEBUT]) --&gt; PESER[PESER « caisse »]     PESER --&gt; D{P &gt;= 20Kg}     D --&gt; A1[AIGILLER « vers tapis 1 »]     D --&gt; A2[AIGILLER « vers tapis 2 »]     A1 --&gt; F1[FAIRE « C2=C2+1 »]     A2 --&gt; F2[FAIRE « C2=C2+1 »]     F1 --&gt; FIN([FIN])     F2 --&gt; FIN     </pre>

Exemple 12: Exemple de structure « SI... ALORS... SINON »

Structure alternatives réduite

Algorithme littérale	Algorithme graphique
<p><b>SI</b> « condition vrai »</p> <p><b>ALORS</b></p> <p>traitement 1</p> <p><b>FIN SI</b></p>	<pre> graph TD     DEBUT([DEBUT]) --&gt; D{condition}     D --&gt; T[Traitement 2]     T --&gt; FIN([FIN])     D --&gt; FIN     </pre>

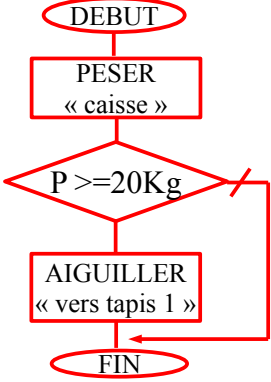
Exemple 13: Structure « SI... ALORS... »

En anglais : IF... THEN...

Seule la situation correspondant à la validation de la condition entraîne l'exécution du traitement dans le cas où la condition n'est pas satisfaite, le traitement n'est pas exécuté et la structure est abandonnée.

### EXEMPLE

Sur une chaîne de conditionnement un dispositif de tri permet de diriger les caisses de Masse supérieure ou égale à 20Kg vers le tapis 1, les autres continuent d'avancer sur le même tapis 2.

Algorithme littérale	Algorithme graphique
<p>DEBUT</p> <p>PESER "Caisse"</p> <p><b>SI</b> "P&gt;=20Kg"</p> <p>    <b>ALORS</b></p> <p>        AIGUILLER « vers tapis 1 »</p> <p>        FAIRE « C1=C1+1 »</p> <p>    <b>SINON</b></p> <p>        AIGUILLER « vers tapis 2 »</p> <p>        FAIRE « C2=C2+1 »</p> <p><b>FIN SI</b></p> <p>FIN</p>	 <pre>graph TD; DEBUT([DEBUT]) --&gt; PESER[PESER &lt;&lt; caisse &gt;&gt;]; PESER --&gt; DECISION{P &gt;= 20Kg}; DECISION --&gt; AIGUILLER[AIGUILLER &lt;&lt; vers tapis 1 &gt;&gt;]; AIGUILLER --&gt; FIN([FIN]); DECISION --&gt; AIGUILLER;</pre>

Exemple 14:

Exemple de structure « SI... ALORS... »

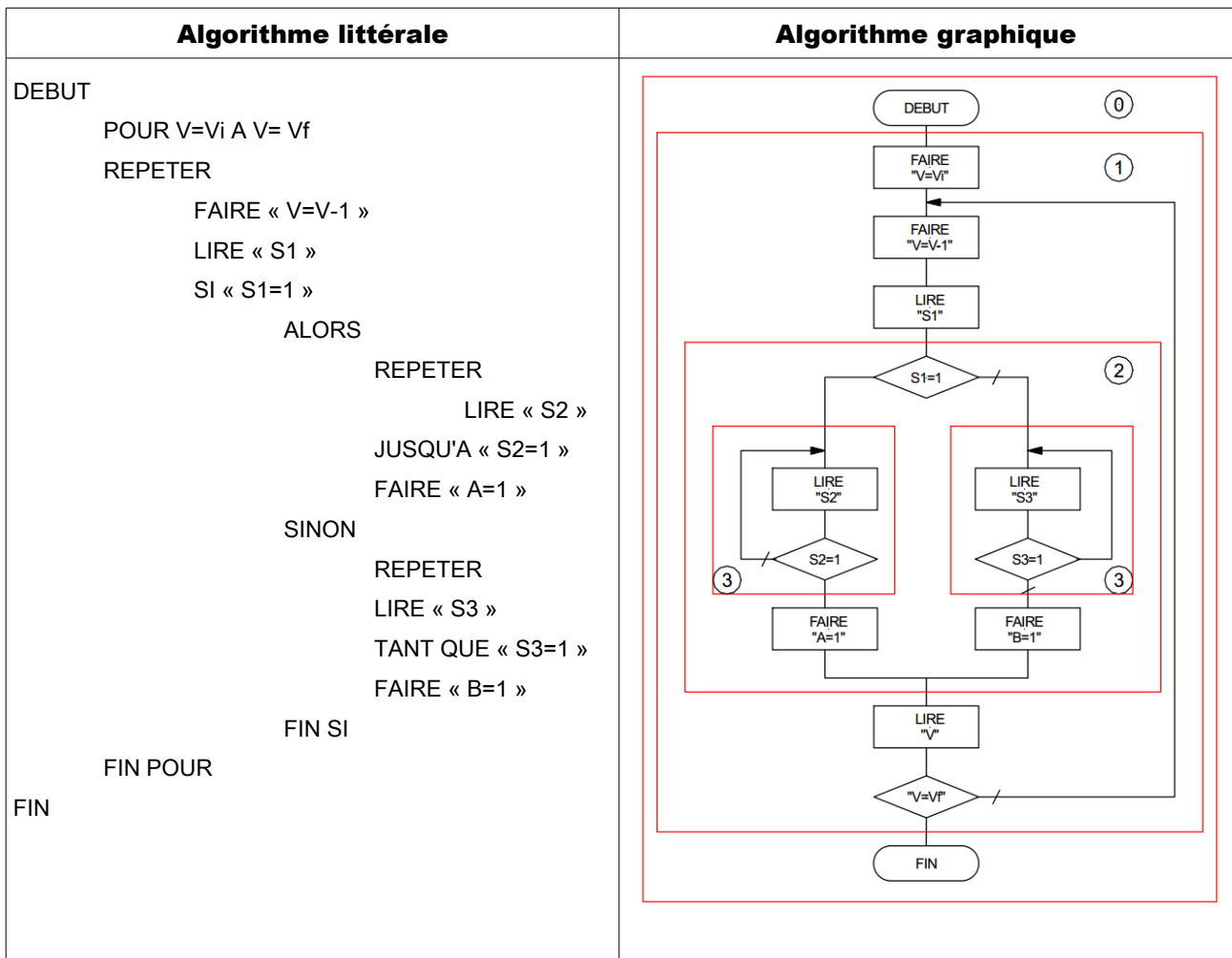
## 5. Structures emboîtées

Le niveau de difficultés des problèmes posés par l'automatisation des processus industriels conduisent souvent à rédiger des algorithmes de degré de complexité d'ordre supérieur à 1 utilisant les structures fondamentales en combinaisons emboîtées.

L'algorithme suivant intègre dans un ordre d'emboîtement décroissant :

- une structure itérative POUR... A... REPETER...
- une structure alternative COMPLETE
- deux structures itératives REPETER... JUSQU'A... et REPETER... TANT QUE...

Son degré de complexité est d'ordre 3



Exemple 15:

Exemple de structures emboîtées

# Annexes

## 6. Index des illustrations

### Index des illustrations

Illustration 1: Représentation algorithmique.....	5
---	---

## 7. Index des exemples

### Index des exemples

Exemple 1: Exemple de structure algorithmique.....	3
Exemple 2: Algorithme de complexité d'ordre 2.....	5
Exemple 3: Structure linéaire.....	6
Exemple 4: Exemple de structure linéaire.....	6
Exemple 5: Structure « REPETER... JUSQU'A... ».....	7
Exemple 6: Exemple de structure « REPETER... JUSQU'A... ».....	7
Exemple 7: Structure « REPETER... TANT QUE... ».....	8
Exemple 8: Exemple de structure « REPETER... TANT QUE... ».....	8
Exemple 9: Structure « POUR... A... REPETER ».....	9
Exemple 10: Exemple de structure « POUR... A... REPETER ».....	10
Exemple 11: Structure « SI... ALORS... SINON ».....	10
Exemple 12: Exemple de structure « SI... ALORS... SINON ».....	11
Exemple 13: Structure « SI... ALORS... ».....	11
Exemple 14: Exemple de structure « SI... ALORS... ».....	12
Exemple 15: Exemple de structures emboîtées.....	13