

### 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 (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 flêché et fermé, compris entre un début et une fin.

SYMBOLE	DESIGNATION
	<b>Traitement</b> Opération ou calcul sur des données dont le résultat reste dans le microcontrôleur.
	<b>Sous-programme</b> Portion de programme considérée comme une simple opération.
	<b>Entrée-sortie</b> Mise à disposition d'une information (écriture sur un port de sortie) ou enregistrement d'une information (lecture d'un port d'entrée).
	<b>Embranchement</b> Test, exploitation de conditions variables impliquant le choix d'une parmi deux. Symbole utilisé pour représenter une décision.
	<b>Début, fin, interruption</b> Début, fin ou interruption d'un organigramme.
<b>Sens conventionnel des liaisons</b> Le sens général des lignes de liaison doit être : - de haut en bas, - de gauche à droite.	

### 3- STRUCTURES ALGORITHMIQUES FONDAMENTALES

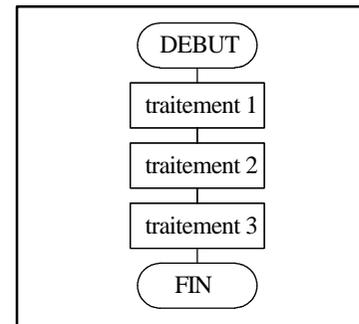
Les opérations relatives à la résolution d'un problème peuvent en fonction de leur enchaînement, être organisées selon trois familles de structures :

- 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

Début

---



---



---

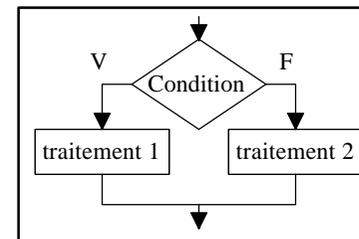
Fin

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

##### 3.2.1- LA STRUCTURE ALTERNATIVE COMPLÈTE

Organigramme



Langage algorithmique

---



---

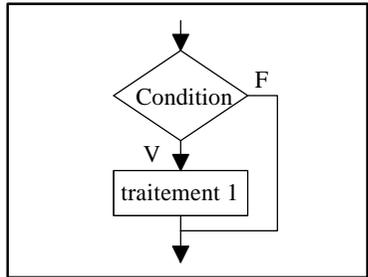


---

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

---



---

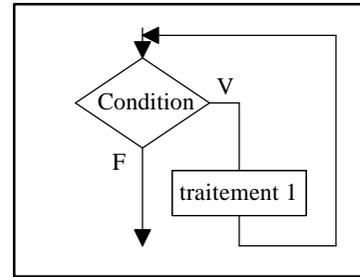


---

3.3.2• TANT QUE ... FAIRE ...

Dans cette structure on commence par tester la condition, si elle est vraie alors le traitement est exécuté.

Organigramme



Langage algorithmique

---



---



---

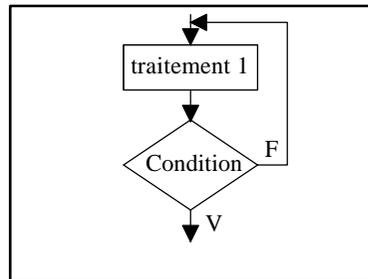
3.3• LES STRUCTURES RÉPÉTITIVES

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

---



---



---

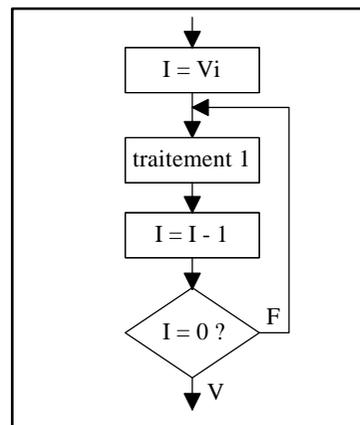
3.4• STRUCTURE RÉPÉTITION CONTRÔLÉE

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

---



---



---

## 4• EXERCICES D'APPLICATION

Pour les exemples suivants, on utilise un PIC 877A avec 8 leds reliées au PORT B et 8 capteurs reliés au PORT C.

### 4.1• ORDINOGRAMMES UTILISANT LES OPÉRATIONS DE LECTURE ET D'ÉCRITURE SUR UN PORT

#### 4.1.1• ECRITURE SUR UN PORT DE SORTIE

✍ On veut écrire un programme qui allume les 8 leds. Dessiner l'ordinogramme qui traduit ce fonctionnement :

#### 4.1.2• CLIGNOTANT

✍ On veut écrire un programme qui fait clignoter les 8 leds (allumées 500ms, éteintes 500ms, etc...)

Dessiner l'ordinogramme qui traduit ce fonctionnement :

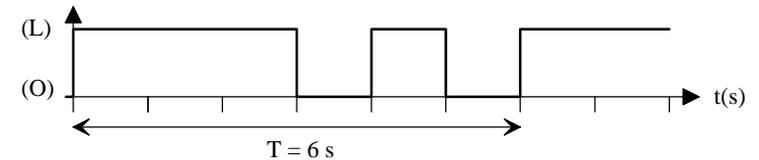
#### 4.1.3• RECOPIE D'UN PORT D'ENTRÉE SUR UN PORT DE SORTIE

✍ On veut écrire un programme qui lit l'état des 8 capteurs et le recopie sur les 8 leds. Ces opérations sont ensuite répétées en boucle toutes les 500ms. Dessiner l'ordinogramme qui traduit ce fonctionnement. On utilisera une variable nommée CAPTEURS pour mémoriser l'état des capteurs.

### 4.2• ORDINOGRAMME UTILISANT UNE STRUCTURE ALTERNATIVE

#### 4.2.1• BALISE MARITIME

La balise maritime est équipée à son sommet d'un système d'éclairage qui sert à guider les bateaux dans la nuit. Le signal lumineux émis par la balise est intermittent et possède un rythme propre qui permet de l'identifier. Le rythme est donné par la répartition des temps de lumière (L) et d'obscurité (O) :



Une cellule photoélectrique permet de réaliser la détection du jour et de la nuit : la balise va s'allumer automatiquement la nuit et s'éteindre le jour.

La cellule produit une information logique sur C0. C0 = '1' lorsqu'il fait nuit. On utilisera une variable NUIT pour mémoriser cette information.

La lampe qui s'allume et s'éteint selon le rythme défini sera reliée à D0.

✍ Dessiner à la page suivante l'ordinogramme qui traduit ce fonctionnement.

### 4.3• ORDINOGRAMME UTILISANT UNE STRUCTURE RÉPÉTITIVE

#### 4.3.1• DEFILEMENT

✍ On veut écrire un programme qui allume successivement chacune des 8 leds (B0 à B7) pendant 200ms. Ces opérations sont ensuite répétées en boucle. Dessiner l'ordinogramme qui traduit ce fonctionnement en utilisant la structure **tant que**. On utilisera une variable nommée DEFILE (de type entier) pour mémoriser l'information envoyée vers les leds.