

1. Définitions

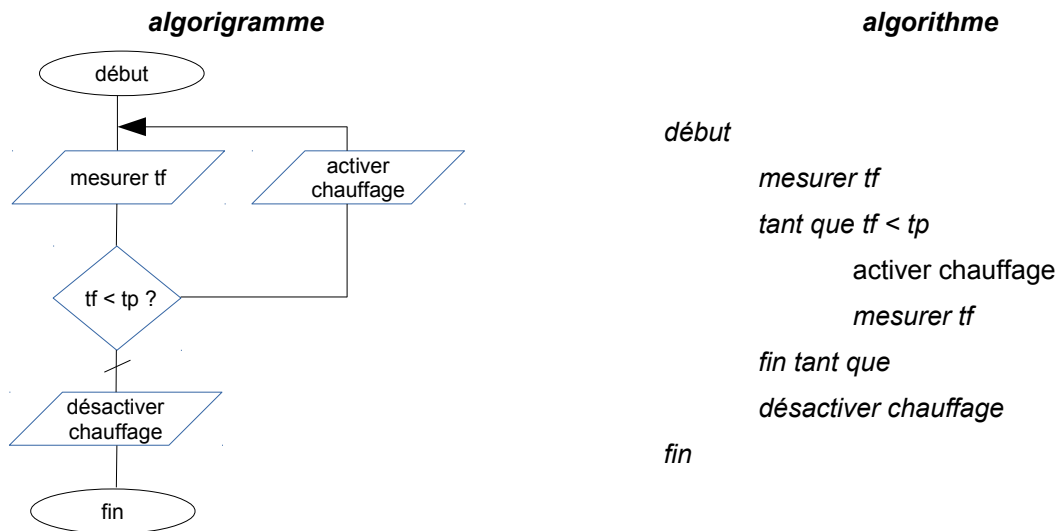
Un **algorithme** est une description en langage naturel de la suite des actions effectuées par un programme. L'algorithme utilise un ensemble de mots clés (**début**, **fin**, **faire**, **tant que**, **répéter**, **jusqu'à**, ...). L'avantage de ce langage est sa transcription facile en langage de programmation dit évolué (Basic, Pascal, C, Java, ...).

Un **algorithme** correspond à une représentation graphique normalisée de l'enchaînement des opérations et des décisions effectuées par le programme.

Les **symboles** décrivant un algorithme sont définis par des **normes** (NF Z 67-010 et ISO 5807) :

SYMBOLE	DÉSIGNATION	SYMBOLE	DÉSIGNATION
	début ou fin d'un algorithme		Test ou Branchement conditionnel décision d'un choix parmi d'autres en fonction des conditions
	symbole général de « traitement » opération sur des données, instructions, ... ou opération pour laquelle il n'existe aucun symbole normalisé		sous-programme appel d'un sous-programme
	entrée / sortie Lecture d'une entrée ou écriture vers une sortie		Liaison Les différents symboles sont reliés entre eux par des lignes de liaison. Le cheminement va de haut en bas et de gauche à droite. Un cheminement différent est indiqué à l'aide d'une flèche
	commentaire		Boucle (symbole de la norme ISO5807) Utilisé pour réaliser des boucles conditionnelles. On note la condition de réalisation de la boucle dans les symboles de début ou de fin suivant la position de l'opération de test.

Exemple : un four à micro-ondes chauffe pendant un temps de fonctionnement t_f , jusqu'à ce que t_f atteigne le temps t_p , programmé par l'utilisateur.



2. Les variables

Dans un programme informatique, il est souvent nécessaire de stocker provisoirement des valeurs (nombre, texte, etc.). Pour stocker une information au cours d'un programme, on utilise une **variable** qui sera stockée dans la mémoire du système micro-programmé.

Une variable est constituée d'un **nom**, et éventuellement d'une **valeur initiale** donnée au moment de sa **déclaration**.

Une variable peut être de **type** :

- **numérique** : elle peut être déclarée comme un octet, un entier ou un réel :

Type de variable numérique	Plage de valeurs
octet (byte)	0 à 255
entier simple (integer)	-32 768 à 32 767
entier long (long integer)	-2 147 483 648 à 2 147 483 647
réel simple (float)	-3,40x10 ³⁸ à -1,40x10 ⁴⁵ pour les valeurs négatives 1,40x10 ⁻⁴⁵ à 3,40x10 ³⁸ pour les valeurs positives
réel double (double)	-1,79x10 ³⁰⁸ à -4,94x10 ⁻³²⁴ pour les valeurs négatives 4,94x10 ⁻³²⁴ à 1,79x10 ³⁰⁸ pour les valeurs positives

- **caractère** (char) : correspond à un caractère alphanumérique (lettre, chiffre, etc.) ;
- **chaîne de caractères** (string) : plusieurs caractères alphanumériques ;
- **booléen** (boolean) : on y stocke uniquement les valeurs logiques VRAI et FAUX ou 0 et 1.

Dans un algorithme, la **déclaration de variables** se fait avant le mot clé début :

```

variable g de type entier
variables prixHT, tauxTVA, prixTTC de type réel
début
...
fin
    
```

3. Les différentes structures algorithmiques

Il existe trois structures algorithmiques différentes :

- la structure **linéaire** ou séquentielle ;
- la structure **alternatives** ou **conditionnelle** ;
- les structures **répétitives** ou itératives.

3.1. La structure linéaire

Les actions s'exécutent successivement dans l'ordre d'écriture.

Exemple d'une structure linéaire - un feu tricolore placé à un carrefour suit un même cycle à l'infini :

- feu vert allumé, feu orange éteint, feu rouge éteint ;
- temporisation de 20s
- feu vert éteint, feu orange allumé, feu rouge éteint ;
- temporisation de 5s
- feu vert éteint, feu orange éteint, feu rouge allumé ;
- temporisation de 20s

Écrire l'algorithme puis l'algorigramme.

algorigramme	algorithme

3.2. La structure alternative ou conditionnelle

Elle offre deux possibilités suivant une **condition**. L'exécution d'un des deux traitements dépend du résultat d'un test effectué sur une condition :

- si la condition est vraie, seul le premier traitement est exécuté;
- si la condition n'est pas vérifiée, seul est effectué le second traitement.

algorithme	algorithme
	<pre> si condition alors action1 sinon action2 fin si </pre>

Exemple d'une structure alternative complète – tri de sacs

A la sortie de l'atelier de conditionnement d'une usine de fabrication d'engrais, un même convoyeur à bande transporte indifféremment des sacs de 25 Kg et 50 Kg. Un dispositif de tri automatique dirige ces sacs vers deux zones distinctes de stockage, D1 et D2.



algorithme	algorithme
	<pre> si condition alors action1 sinon action2 fin si </pre>

La structure conditionnelle peut aussi se trouver sous la forme **réduite** : si la condition n'est pas vérifiée aucune action n'est exécutée.

algorithme	algorithme
	<pre> si condition alors action1 fin si </pre>

3.3. LES STRUCTURES RÉPÉTITIVES

Il existe différents types de structures répétitives : soit le nombre de répétitions est connu, soit il est inconnu.

3.3.1. Le nombre de répétitions n'est pas connu

répéter ... tant que ...		tant que ... faire ... fin tant que	
Le traitement est exécuté une première fois puis sa répétition se poursuit tant que la condition est vérifiée.		On commence par tester la condition, si elle est vérifiée alors le traitement est exécuté tant que cette condition est vérifiée.	
algorithme	algorithme	algorithme	algorithme
	<p>répéter</p> <p>action</p> <p>tant que condition vraie</p>		<p>tant que condition vraie</p> <p>faire action</p> <p>fin tant que</p>
L'action est toujours exécutée au moins une fois		L'action peut ne jamais être exécutée	

<p>Exemple : alimentation en eau d'un lave-linge Une électrovanne assure le remplissage de la cuve d'un lave-linge tant que le NH ne détecte pas le niveau haut.</p>		<p>Exemple : chauffage d'un four Un four doit être chauffé à une température t de 500°C.</p>	
algorithme	algorithme	algorithme	algorithme

3.3.2. **Le nombre de répétitions est connu**

La sortie de la boucle d'itération s'effectue lorsque le **nombre souhaité de répétitions est atteint**.

On utilise une variable (ou indice) de comptage d'itération, caractérisé par sa valeur initiale et sa valeur finale.

Si la valeur finale est inférieure à la valeur initiale, la structure est dite décroissante.

Si la valeur finale est supérieure à la valeur initiale, la structure est dite croissante.

structure POUR croissante		structure POUR décroissante	
algorithme	algorithme	algorithme	algorithme
	<p>pour indice de V_i à V_f action fin pour</p>		<p>pour indice de V_i à V_f action fin pour</p>

Exemple – conditionnement de bouteilles

Dans une usine de fabrication de jus de fruits les bouteilles sont conditionnées par six, après contrôle, sous un film plastique rétractable.

algorithme	algorithme