

Cours d'algorithmique

I - Définition

- a. Définition d'un algorithme
- b. Définition d'un algorithme
 - 1 Symbole général
 - 2 Renvoi
 - 3 Sous- programme
 - 4 Entrée Sortie
 - 5 commentaires
 - 6 Branchement

II – L'algorithme

Le langage de description

III – Structure d'un algorithme

- a. Représentation :
 1. L'en-tête
 2. les déclarations
 3. le corps
 4. les commentaires

IV Déclaration de constantes, de variables, et de structures

- a. Les constantes
- b. Les variables
- c. Les structures

V Déclaration de procédures et de fonctions

- a. la procédure
- b. La fonction
 1. Les paramètres

VI Les types de base

1. L'entier
2. Le réel
3. Le booléen
4. Le caractère
5. La chaîne de caractères

VII Les opérateurs

- a. Opérateurs sur les entiers et les réels
- b. Opérateurs sur les entiers
- c. Opérateurs sur les caractères et les chaînes
- d. Priorité des opérateurs

VIII Les structures algorithmiques fondamentales

- a Caractéristique de la structure linéaire
- b Caractéristique de la structure alternative
- c Caractéristique de la structure de choix
- d Caractéristique de la structure itérative

I. DEFINITIONS

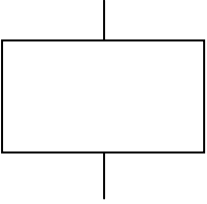

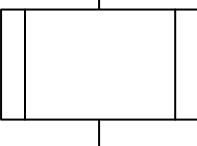

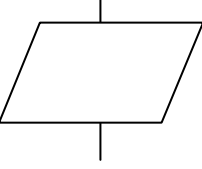
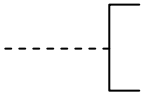
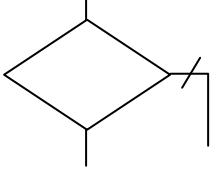
a. Définition d'un algorithme :

C'est un ensemble de règles opératoires rigoureuses, ordonnant à un processeur d'exécuter dans un ordre déterminé un nombre fini d'opérations élémentaires ; il oblige à une programmation structurée.

Un algorithme est écrit en utilisant un langage de description d'algorithme (LDA). L'algorithme ne doit pas être confondu avec le programme proprement dit.

b. Définition d'un algorithme :

C'est une représentation graphique de l'algorithme. Pour le construire, on utilise des symboles normalisés.

Symbole	Désignation	Symbole	Désignation
Symboles de traitement		Symboles auxiliaires	
	Symbole général Opération ou groupe d'opérations sur des données, instructions pour laquelle il n'existe aucune symbole normalisé		Renvoi Symbole utilisé deux fois pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée.
	Sous-programme Portion de programme considérée comme une simple opération		Début, fin, interruption Début, fin ou interruption d'un algorithme
	Entrée-Sortie Mise à disposition d'une information à traiter ou enregistrement d'une information à traiter		Commentaire Symbole utilisé pour donner des indications sur les opérations effectuées
Symbole de test		Les différents symboles sont reliés entre eux par des lignes de liaisons	
	Branchement Exploitation de conditions variables impliquant un choix parmi plusieurs		
Sens conventionnel des liaisons			
Le sens général des lignes de liaison doit être : <ul style="list-style-type: none"> ➤ De haut en bas ➤ De gauche à droite 			
Lorsque le sens général ne peut pas être respecté, des pointes de flèches à cheval sur la ligne indiquent le sens utilisé.			

II. L'ALGORITHME

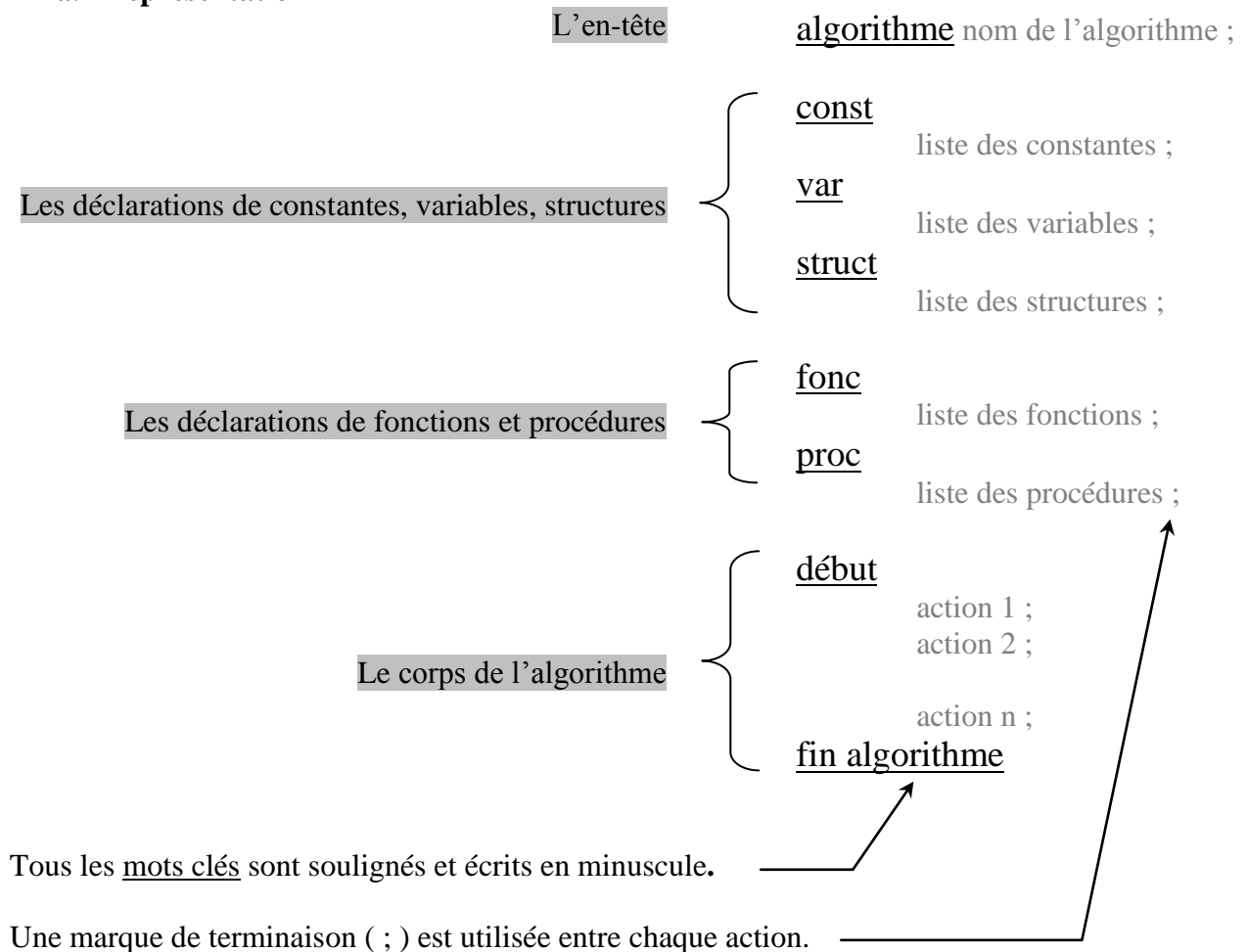
a. Le Langage de description d'algorithme

Ce langage utilise un ensemble de *mots clés* et de *structures* permettant de décrire de manière complète, claire, l'ensemble des opérations à exécuter sur des données pour obtenir des résultats ; on n'hésitera donc pas à agrémenter l'algorithme de nombreux commentaires.

L'avantage d'un tel langage est de pouvoir être facilement transcrit dans un langage de programmation structuré (Pascal, C, ...).

III. STRUCTURE D'UN ALGORITHME

a. Représentation



1. L'en-tête

Il permet tout simplement d'identifier un algorithme

2. Les déclarations

C'est la liste exhaustive des objets, grandeurs utilisés et manipulés dans le corps de l'algorithme ; cette liste est placée en début d'algorithme.

3. Le corps

Dans cette partie de l'algorithme, sont placées les tâches (instructions, opérations) à exécuter.

4. Les commentaires :

Pour permettre une interprétation aisée de l'algorithme, il faut utiliser des commentaires.

Mot Clé : co ceci est un exemple fco

IV. DECLARATION DE CONSTANTES, DE VARIABLES ET DE STRUCTURES.

a. Les constantes

Elles représentent des chiffres, des nombres, des caractères, des chaînes de caractères, ... dont **la valeur ne peut pas être modifiée** au cours de l'exécution de l'algorithme.

Mot clé : const

b. Les variables

Elles peuvent stocker des chiffres, des nombres, des caractères ou chaîne de caractères, ... dont **la valeur peut être modifiée** au cours de l'exécution du programme.

Mot clé : var

Les constantes et les variables sont définies dans la partie déclarative par deux caractéristiques essentielles, à savoir :

- **Le type** : Il détermine la nature de la variable ou de la constante (entier, réel, booléen, chaîne de caractères).
- **L'identificateur** : c'est le nom de la variable ou de la constante. Il est composé de lettres et de chiffres.

Exemple

const

origine c'est 10 ;
posinitiale c'est 0 ;

var

Entier température, consigne ;

c. Instruction d'affectation

Permet de mettre une valeur ou le résultat d'un calcul dans une variable,

Mot clé : ← symbole d'affectation

L'affectation se fait en deux temps :

- 1) Evaluation de l'expression située à droite du symbole
- 2) Affectation du résultat à l'identificateur de variable

Identification de variable ← Valeur

Exemple :

$Y \leftarrow 2 \times x + 3$ ←

d. Les structures

Elles permettent de rassembler plusieurs variables ou constantes sous un même identificateur ; on parle aussi d'entité ou d'objets.

Mot clé : struct ... fstruct

Exemple

struct

```
    disque c'est  
           entier abs,ord ;  
           entier rayon ;  
           entier couleur ;
```

fstruct

V. DECLARATION DE PROCEDURE ET DE FONCTIONS

a. La procédure

C'est un ensemble d'instructions référencé par un nom, et dont l'exécution est provoquée par le simple énoncé de ce nom. La procédure peut recevoir ou renvoyer des valeurs.

Mot clé : proc identificateur (...)

b. La fonction

Comme pour la procédure, l'exécution d'une fonction est provoquée par la simple évocation de son nom. La fonction peut renvoyer au programme principal un résultat. C'est une procédure simplifiée.

Mot clé : fonc identificateur (...)

1. Les paramètres (...)

- Les paramètres entrants peuvent être consultés (et éventuellement modifiés) à l'intérieur de la procédure.
- Les paramètres sortants dont la valeur est déterminée à l'intérieur de procédure et utilisable après l'appel de la procédure
- Les paramètres mixtes, qui ont une valeur à l'entrée dans la procédure, valeur qui peut être modifiée à l'intérieur de celle-ci, la modification étant répercutée à l'extérieur de la procédure.

VI. LES TYPES DE BASE

Nous avons vu qu'une des deux caractéristiques des constantes et de variables était leur type. Nous détaillerons 5 types de base :

1. L'entier

Notation

45, 36, - 564, 0 en décimal
45h,0F1Bh,64h en hexadécimal
%10101110, %1001 en binaire

Mot clé : entier

2. Le réel

Notation

-3.67,4.258,-542.0,18.36 e⁻⁶

Mot clé : réel

3. Le booléen

Il ne peut prendre que deux états : VRAI ou FAUX

Mot clé : booléen

4. Le caractère

Notation

A, a, *, 7, z

Mot clé : car

5. La chaîne de caractères

Notation

« électronique », « charge »

Mot clé : chaîne

Exemple de traduction

Entier	INTEGER
Réel	REAL
Le booléen	BOOL
Le caractère	CHART
La chaîne de caractères	STRING

VII. LES OPERATEURS

a. Opérateurs sur les entiers et les réels

Arithmétiques	
+	Addition
-	Soustraction
*	Multiplication
/	Division
DIV	Division entière
↑	Puissance

Comparaisons	
>	Supérieur
<	Inférieur
≥	Supérieur ou égal
≤	Inférieur ou égal
=	Egal
≠	Différent

b. Opérateurs sur les entiers et les booléens

Fonctions logiques	
<u>Et</u>	Fonction ET
<u>Ou</u>	Fonction OU
<u>Oux</u>	Fonction OU exclusif
<u>Non</u>	Fonction NON
<u>Non et</u>	Fonction NON ET
<u>Non ou</u>	Fonction NON OU
>>	Décalage à droite
<<	Décalage à gauche

Fonctions de comparaison pour les booléens	
=	Egal
≠	Différent

c. Opérateurs sur les caractères et les chaînes

Fonctions de concaténation	
+	Concaténation

Fonctions de comparaison pour les chaînes	
>	Supérieur
<	Inférieur
=	Egalité
≠	Différent

d. Priorité des opérateurs

Priorité à la multiplication et à la division

VIII LES STRUCTURES ALGORITHMIQUES FONDAMENTALES

Les opérateurs élémentaires relatives à la résolution d'un problème peuvent, en fonction de leur enchaînement être organisées suivant quatre familles de structures algorithmiques fondamentales.

- Structures linéaires
- Structures alternatives
- Structures de choix
- Structures itératives (ou répétitives)

a. Caractéristiques de la structure linéaire

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

Mot clé : faire action

b. Caractéristique de la structure alternative

La structure alternative n'offre que deux issues possibles à la poursuite de l'algorithme et s'excluent mutuellement.

On peut rencontrer deux types de structures alternatives :

1. Une structure alternative complète

Dans cette structure l'exécution d'un des deux traitements distincts ne dépend que du résultat d'un test effectué sur la condition qui peut être une variable ou un événement ;

- Si la condition est vérifiée seul le premier traitement est exécuté ;
- Si la condition n'est pas vérifiée seul est effectué le second traitement

Notation :

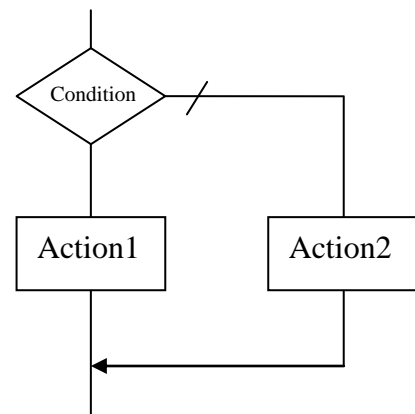
si condition alors

Action1 ;

Sinon

Action2 ;

fsi ;



2. Une structure alternative réduite

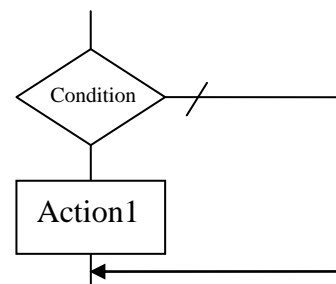
La structure alternative réduite se distingue de la précédente par le fait que seule la situation correspondant à la validation de la condition entraîne l'exécution du traitement, l'autre situation conduisant systématiquement à la sortie de la structure.

Notation :

si condition alors

Action ;

fsi ;



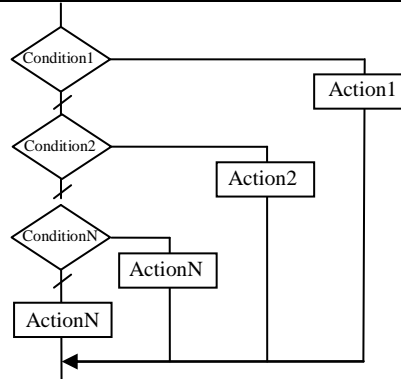
c. Caractéristique de la structure de choix

La Structure de choix permet, en fonction de plusieurs condition de type booléen, d'effectuer des actions différentes suivant les valeurs que peut prendre une même variable

Notation :

suivant valeur faire
valeur1 : action1 ;
valeur2 : action2 ;

valeur N : actionN ;
sinon action N+1
fsuivant ;



d. Caractéristique de la structure itérative

La structure itérative répète l'exécution d'une opération ou d'un traitement.

On considérera 2 cas :

Premier cas : le nombre de répétitions n'est pas connu ou est variable.

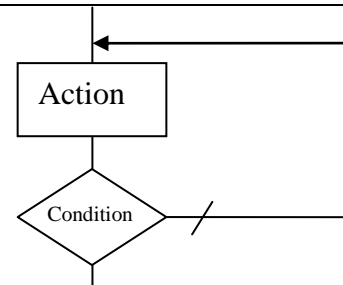
Structure REPETER JUSQU'A

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

L'ACTION EST EXECUTEE AU MOINS UNE FOIS

Notation :

répéter
action ;
Jusqu'à condition vraie ;



Structure TANT QUE ... FAIRE ...

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

L'ACTION PEUT NE JAMAIS ETRE EXECUTEE.

Notation :

tant que condition faire
action ;
Ftant que ;

