

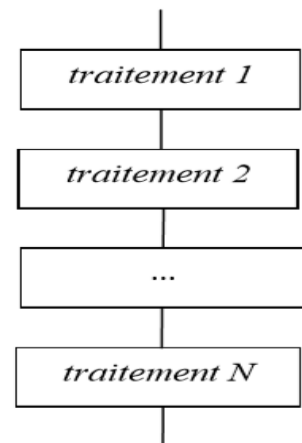
Les structures algorithmiques peuvent être organisées suivant quatre familles principales : Les structures linéaires, les structures alternatives, les structures répétitives et les structures de choix.

### I. LES STRUCTURES LINEAIRES

#### 1. Définition

Une STRUCTURE LINEAIRE (ou STRUCTURE SEQUENTIELLE) se caractérise par une suite de traitements à exécuter successivement, dans l'ordre énoncé.

#### 2. Organigramme (ou Algorigramme) : c'est la représentation graphique d'un algorithme



#### 3. Exemple :

Début de l'algorithme

Lire A

Lire B

Calculer  $A+B$

$C \leftarrow A+B$

Afficher C

Fin de l'algorithme

### II. Les structures alternatives

#### 1. DEFINITIONS

a. Une STRUCTURE ALTERNATIVE (ou STRUCTURE CONDITIONNELLE) n'offre que deux cas possibles à la poursuite de l'algorithme, qui s'excluent mutuellement. Selon qu'une condition est vraie ou fausse, on effectue un traitement ou un autre. On parle de traitements conditionnels. Une structure alternative est donc UNE STRUCTURE DE TEST.

#### b. Condition :

Une CONDITION est une expression logique booléenne, prenant la valeur « vrai » ou « faux ». Il existe des conditions simples et des conditions complexes.

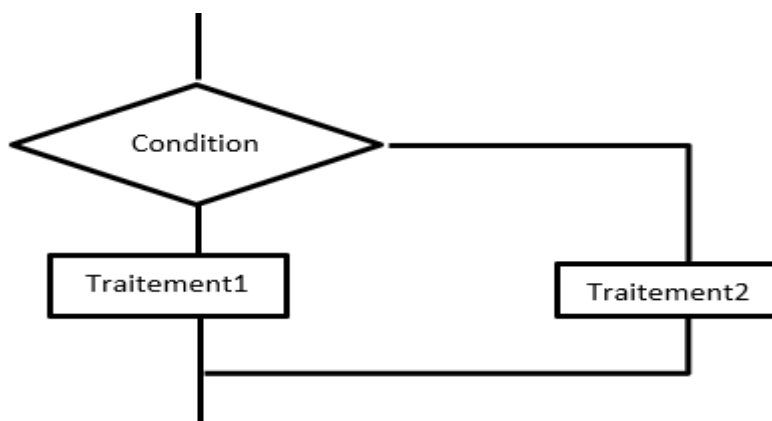
- ✓ Une condition simple, (par exemple :  $A=B$ ,  $A<B$ , ...)
- ✓ Une condition complexe est une combinaison logique de conditions simples, par exemple :  $A=B$  ET  $A<B$

## 2. STRUCTURE ALTERNATIVE COMPLETE

### a. Pseudo-code

Si condition Alors  
    Traitement 1 ;  
Sinon  
    Traitement 2 ;  
FinSi

### b. Organigramme :



### c. Exemple :

Début de l'algorithme  
Lire A  
Si  $A > 10$  alors  
    Afficher « élève admis »  
Sinon  
    Afficher « élève recalé »  
Fin de l'algorithme

## 3. Structure alternative réduite

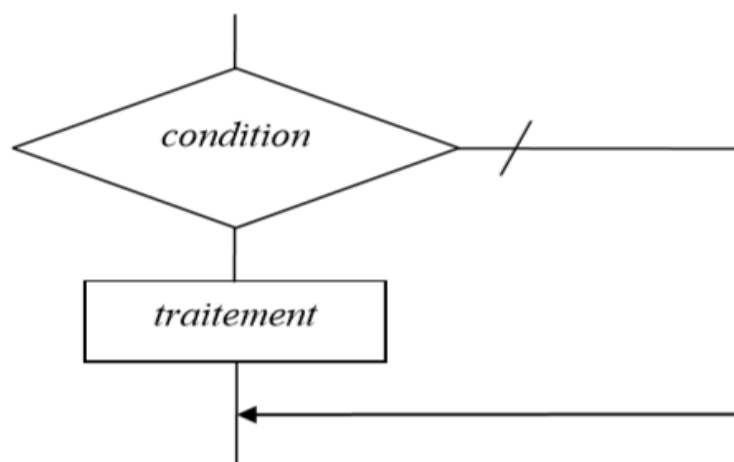
### a. Pseudo-code

Dans le cas où la condition n'est pas vérifiée, il n'y a pas à effectuer de traitement 2. On écrira ainsi la structure alternative « réduite » :

SI condition ALORS traitement

FINSI

### b. Organigramme :



c. Exemple :

Début de l'algorithme

Lire Moyenne

SI Moyenne=10 ALORS

Afficher « juste Moyen »

FINSI

Fin de l'algorithme

#### 4. STRUCTURES ALTERNATIVES IMBRIQUEES

##### a. Définition

Lorsque les conditions de tests sont nombreuses, le temps d'exécution devient plus couteux. Il serait ainsi bien plus rationnel de réduire le nombre de conditions via l'imbrication de tests. Les structures de tests imbriqués sont donc un outil indispensable à la simplification et à l'optimisation des algorithmes.

##### b. Exemple :

Un programme devant donner des appréciations pour les étudiants doit pouvoir choisir entre trois réponses possibles (recaler, passable ou mention).

Algorithme bulletin\_Reduite

Variable

Note : Entier

Début

Ecrire "Entrez la note :"

Lire note

Si note < 10 Alors

Ecrire "recaler"

FinSi

Si note >= 10 Et note < 12 Alors

Ecrire "passable"

Finsi

Si Temp >= 12 Alors

Ecrire "mention"

Finsi

Fin

Remarque : on oblige la machine à examiner trois tests successifs alors que tous portent sur une même chose, la note (la valeur de la variable note).

Il serait ainsi bien plus rationnel de réduire le nombre de tests :

## Algorithme bulletin\_Imbriqué

### Variable

Note : Entier

### Début

Ecrire "Entrez la note :"

Lire note

Si note < 10 Alors

Ecrire "recaler"

sinon

Si note < 12 Alors

Ecrire "passable"

sinon

Ecrire "mention"

Finsi

Finsi

### Fin

#### c. Structure fusionné (simplifié) :

Dans le cas de tests imbriqués, le Sinon et le Si peuvent être fusionnés en un SinonSi. On considère alors qu'il s'agit d'un seul bloc de test, conclu par un seul FinSi

## Algorithme bulletin\_Imbriqué

### Variable

Note : Entier

### Début

Ecrire "Entrez la note :"

Lire note

Si note < 10 Alors

Ecrire "recaler"

sinonSi note < 12 Alors

Ecrire "passable"

sinon

Ecrire "mention"

Finsi

### Fin

### 5. Structures alternatives et operateurs logiques :

a. Activité : est ce qu'on peut reconstruire la structure suivante avec l'opérateur ET ?

Si note < 0 OU note > 20 alors

Traitement1

Sinon

Traitement2

Finsi

b. Définition :

Toute structure de test requérant une condition composée faisant intervenir l'opérateur ET peut être exprimée de manière équivalente avec un opérateur OU, et réciproquement.

c. Pseudo-code

**Si A ET B Alors**  
    **Instructions 1**  
**Sinon**  
    **Instructions 2**  
**Finsi**

Équivaut à :

**Si NON A OU NON B Alors**  
    **Instructions 2**  
**Sinon**  
    **Instructions 1**  
**Finsi**

Remarque :

Cette règle porte le nom de transformation de Morgan , du nom du mathématicien anglais qui l'a formulée

d. Exemple :

Formulez un algorithme équivalent à l'algorithme suivant :

Si Total > TVA OU Total = 0 Alors

    Total ← Total + 1

Sinon

    Total ← Total – 1

Finsi

#### Solution

Si Total <= TVA ET Total <> 0 Alors

    Total ← Total – 1

Sinon

    Total ← Total + 1

Finsi

### III. LES STRUCTURES REPETITIVES

Deux cas sont à envisager : boucles itératives et boucles conditionnelles.

#### 1. BOUCLES CONDITIONNELLES

## Activité

Variable

Réponse : Caractère

Début

Ecrire "Voulez-vous un café ? (O/N)"

Lire Reponse

Si Reponse <> "O" et Reponse <> "N" Alors

Ecrire "Saisie erronée. Recommencez"

Lire Reponse

FinSi

Fin

Dans cet exemple le test est réalisé une seule fois, donc si l'utilisateur a saisi une valeur erronée pour la deuxième fois le test ne sera pas effectué. → Donc on utilise les boucles.

a. La boucle TANT QUE ... FAIRE

- i. Définition : dans ce cas le nombre de répétitions n'est pas connu à l'avance, donc on répète autant de fois que la condition est vérifiée.
- ii. Syntaxe

Tanque condition faire

Traitement

FinTanque

iii. Algorithme

iv. Remarques :

- ✓ Le nombre de répétitions dépend de la condition.
- ✓ Si la condition n'est pas vérifiée au début, alors le traitement ne sera pas exécuté du tout.
- ✓ Si la condition est vérifiée au début et elle reste toujours vérifiée, alors on entre dans une boucle infinie. Dans ce cas, il s'agit d'une erreur majeure car un programme ne doit pas boucler indéfiniment.

v. Exemple :

Variable

Réponse : Caractère

Début

Ecrire "Voulez-vous un café ? (O/N)"

Lire Reponse

TantQue Rep <> "O" et Rep <> "N"

Ecrire "Saisie erronée. Recommencez"

Lire Rep

FinTantQue

Fin

Remarque : Si on suppose que l'utilisateur saisi toujours une valeur erronée, alors → boucle infinie, mais c'est une possibilité faible.

b. La boucle REPETER ... JUSQU'A ...

i. Définition : Une variante de BOUCLE CONDITIONNELLE consiste à répéter un traitement jusqu'à ce qu'une certaine condition soit vérifiée.

ii. Syntaxe :

Répète

Traitement

Jusqu' à condition

iii. Algorithme :

iv. Remarque : Dans ce type, le test est effectué à la fin de la boucle, donc le traitement est exécuté au moins une fois.

## 2. BOUCLE ITERATIVE

### Activité

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 8) :

1 X 8 = 8  
2 X 8 = 16  
3 X 8 = 24  
4 X 8 = 32  
5 X 8 = 40  
6 X 8 = 48  
7 X 8 = 56  
8 X 8 = 64  
9 X 8 = 72  
10 X 8 = 80

### SOLUTION

Variables N, i en Entier

Début

Ecrire "Entrez un nombre : "

Lire N

Ecrire "La table de multiplication de ce nombre est : "

$i \leftarrow 0$

Tantque  $i < 10$

$i \leftarrow i + 1$

Ecrire N, " x ", i, " = ",  $N*i$

i Suivant

Fin

a. Définition : consiste à répéter un certain traitement un nombre de fois fixé à l'avance.

b. Syntaxe :

Pour  $i \leftarrow 1$  à N

Traitement

FinPour

En général on a :

**Pour Compteur  $\leftarrow$  Initial à Final Pas ValeurDuPas**

**... Instructions ...**

**Compteur suivant**

c. Algorithme

d. Remarque : La variable est un compteur, dont la valeur augmente automatiquement de 1 à chaque tour.

e. Exercice :

✓ Réécrit l'exercice précédent en utilisant la boucle Pour.



### SOLUTION

**Variables N, i en Entier**

**Début**

Ecrire "Entrez un nombre : "

Lire N

Ecrire "La table de multiplication de ce nombre est : "

Pour i ← 1 à 10

Ecrire N, " x ", i, " = ", N\*i

i Suivant

**Fin**

✓ Comparer  
les deux  
boucles : la  
boucle tanque

est plus générale que la boucle Pour.

### 3. BOUCLE ITERATIVE IMBRIQUE

a. Remplir le tableau suivant en utilisant la boucle POUR

Etudiant1	Etudiant2	Etudiant3	Etudiant4	Etudiant5
-----------	-----------	-----------	-----------	-----------

Pour i de 1 à 5

Ecrire « entrer le nom de l'étudiant : »

Lire valeurs

FinPour

b. Remplir le tableau suivant en utilisant la boucle POUR

	Etudiant1	Etudiant2	Etudiant3	Etudiant4	Etudiant5
Note1					
Note2					
Note 3					

Pour i de 1 à 5

Ecrire « entrer le nom de l'étudiant : »

Lire nom

Pour j de 1 à 3

Ecrire « entrer la note : »

Lire note

FinPour

FinPour

## IV. Les structures de choix

### 1 Activité :

Ecrire un algorithme permettant d'afficher le mois en toute lettre selon son numéro saisi au clavier :

❖ La seule structure convenable vu précédemment est la structure fusionnée :

Algorithme Mois

Variables n : Entier

Début

Ecrire ("Donner le numéro du mois: ")

Lire (n)

Si n = 1 alors

Ecrire (« janvier ») ;

Sinon si n = 2 alors

Ecrire (« Fevrier ») ;

.....

Sinon si n = 12 alors

Ecrire (« December ») ;

Sinon

Ecrire (« erreur »)

FinSi

FIN

### 2 Remarques :

Dans cet exemple on remarque que :

- ✓ La variable n est de type discrète (Entier ou Caractère).
- ✓ Les valeurs de n sont des constantes (pas de variables).

Si ces réglés sont vérifiées, La structure de choix (listes de valeurs) est plus efficace qu'une série de Si en cascade (car l'expression de La structure de choix (listes de valeurs) n'est évaluée qu'une seule fois et non en chacun des Si)

### 3 Définitions :

La structure de choix est une simplification d'écriture de plusieurs alternatives imbriquées. Deux formes existent :

- La structure de choix (listes de valeurs)
- La structure de choix (conditions)

La forme acceptée par la plupart des langages de programmation est celle avec listes de valeurs.

### 4 Syntaxe de La structure de choix (listes de valeurs) :

Selon sélecteur faire

cas 1 : Instructions1

cas 2 : Instructions 2

cas 3 : Instructions 3

....

cas n : Instructions n

Sinon

Autre cas

FinSelon

## 5 Remarques :

- ✓ Lorsque l'ordinateur rencontre cette instruction, il vérifie la valeur de la variable de sélection (sélecteur) et il la compare aux différentes valeurs.
- ✓ Les valeurs sont évaluées dans l'ordre, les unes après les autres, et dès qu'une de celle-ci est vérifiée l'action associée est exécutée. On peut utiliser une instruction Sinon (facultative), dont l'action sera exécutée si aucune des valeurs évaluées n'a été remplie.

## 6 Exercice :

- ✓ Donnez la Version structure de choix (listes de valeurs) de l'activité précédente:

Algorithme Mois

Variables N : Entier

Début

Ecrire ("Donner le numéro du mois: ")

Lire (N)

Selon N faire

1 : Ecrire ("Janvier")

2 : Ecrire ("Février")

3 : Ecrire ("Mars")

4 : Ecrire ("Avril")

5 : Ecrire ("Mai ")

6 : Ecrire ("Juin")

7 : Ecrire ("Juillet")

8 : Ecrire ("Août")

9 : Ecrire ("Septembre")

10 : Ecrire ("Octobre")

11 : Ecrire ("Novembre")

12 : Ecrire ("Décembre")

Sinon

Ecrire ("Le numéro saisi est incorrecte: ")

FinSelon

Fin