

**Chapitre III:**

# **Les Structures Alternatives (Instructions Conditionnelles)**

(en algorithmique et en langage C)

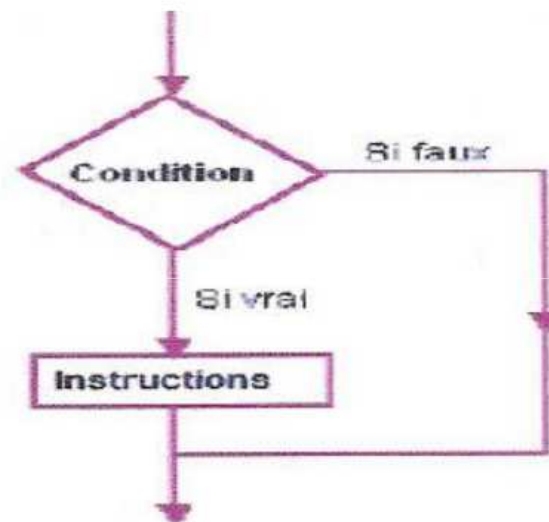
# Instructions à structure alternative SI

Son intérêt est de permettre de réaliser des choix dans un programme: **Faire telle chose SI telle condition est réalisée et SINON faire telle autre chose.**

## Structure alternative simple

*Syntaxe:*

```
SI condition ALORS
  Instruction 1
  Instruction 2
  ...
  Instruction N
FINSI
```



*Si la condition est vérifiée alors le bloc d'instructions sera exécuté, sinon il sera ignoré.*

*Exemple:*

```
Y ← X
Si X < 0 alors
  Y ← -X;
FinSi
```

# Instructions à structure alternative SI

## Structure alternative complète

*Syntaxe:*

**SI** condition **ALORS**

Instruction 1-1

Instruction 1-2

...

Instruction 1-N

**SINON**

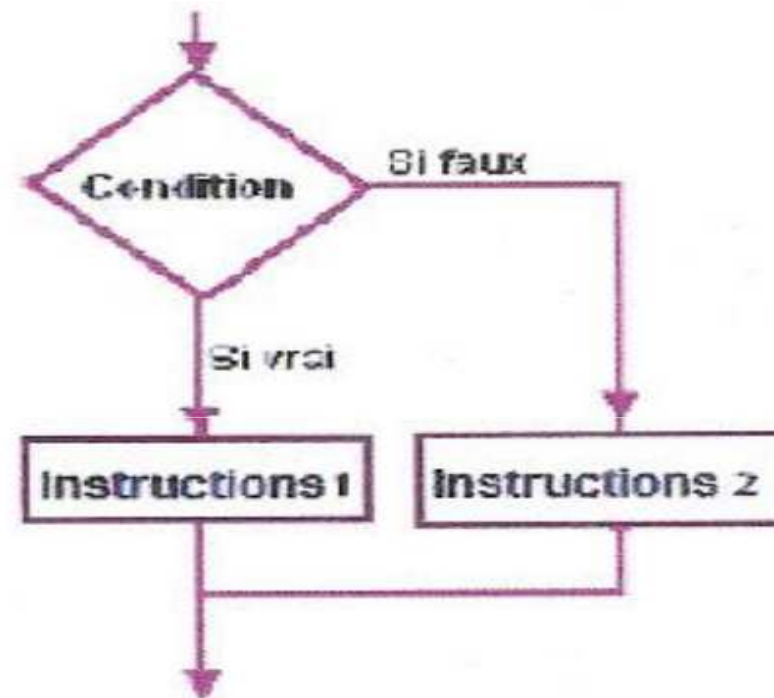
Instruction 2-1

Instruction 2-2

...

Instruction 2-M

**FINSI**



**Si** la condition est **vérifiée** alors le bloc d'instructions 1 sera exécuté et le bloc d'instructions 2 sera ignoré, **sinon** le bloc instructions 1 sera ignoré et le bloc instructions 2 sera exécuté.

# Instructions à structure alternative SI

## Exemples :

### Algorithme Test 1

**Variable** x : entier

**Début**

**Écrire** ('Saisir un entier x ')

**Lire** (x)

**Si** (x > 0) **alors**

**Écrire**('x est un nombre positif ')

**Finsi**

**Fin**

### Algorithme Test 2

**Variable** x : entier

**Début**

**Écrire** ('Saisir un entier x ')

**Lire** (x)

**Si** (x > 0) **alors**

**Écrire** (' x est un nombre positif ')

**Sinon**

**Écrire** (' x est un nombre négatif ou nul')

**Finsi**

**Fin**

# Instructions à structure alternative SI

## Structures alternatives imbriquées

Si on a plus que deux choix dans un traitement conditionnel, on utilise les instructions alternatives imbriquées.

*Syntaxe:*

```
SI condition1 ALORS
  Si condition2 Alors
    Instruction A
  Sinon
    Instruction B
  FinSi
SINON
  Si condition3 Alors
    Instruction C
  FinSi
FINSI
```

*Exemple:*

```
Début
  Ecrire('entrez un nombre:')
  Lire(n)
  si(n<0) alors
    Ecrire('ce nombre est négatif')
  sinon
    si ( n=0) alors
      Ecrire('ce nombre est nul')
    sinon
      Ecrire ('ce nombre est positif')
    finsi
  finsi
fin
```

## Instructions à structure alternative

### Exemple :

Dans l'exercice où on a calculé la moyenne générale, Afficher 'Admis' si un étudiant a une moyenne générale  $\geq 10$  et Afficher 'Ajourné' dans le cas contraire (Moyenne générale  $< 10$ ).

### Solution :

.....(*Reste de l'algorithme*)

**Si** (MG  $\geq 10$ ) **alors**

**Écrire** ('Admis')

**Sinon**

**Écrire** ('Ajourné')

**Finsi**

**FIN**

## Instructions à structure alternative

### Exercice :

**Écrire l'algorithme qui permet de calculer le maximum de deux entiers quelconques.**

## Instructions à structure alternative

### Solution :

**Titre :** Maximum

**Variable** a ,b, max : entier

**Début**

**Écrire** ('Saisir deux entiers a et b')

**Lire**(a, b)

**Si** (a > b) **alors**

**max** ← a

**Sinon**

**max** ← b

**Finsi**

**Écrire** ('le maximum de ', a, ' et de ', b, ' est : ', max)

**Fin**



## Instructions à structure alternative

### Exercice :

*Écrire l'algorithme qui permet de déterminer le salaire mensuel d'un commercial sachant que ce salaire comporte un montant fixe de 4000 DHS et une commission qui représente 5% du chiffre d'affaire réalisé par mois si ce chiffre est  $< 30000$  et de 10 % dans le cas contraire .*

# Algorithme

## ... Suite de l'algorithme

**Si** (CA < 30000) **alors**

Com ← CA \* 0.05

**Sinon**

Com ← CA \* 0.1

**Finsi**

Sal ← Com + M

**Écrire** ('Le salaire mensuel est de : ', Sal , 'DHS')

**FIN**

## Instructions à structure alternative

### Exercice :

*Compliquons un peu l'énoncé:*

*La commission est calculée de la manière suivante :*

- *Commission = 15% du CA quand  $CA > 100000$*
- *Commission = 10% du CA quand  $30000 < CA \leq 100000$*
- *Dans le cas contraire pas de commission*

*Écrire l'algorithme qui permet de déterminer le salaire mensuel.*

# Algorithme

## ... Suite de l'algorithme

**Si** (CA > 100000) **alors**

Com ← CA \* 0.15

**Sinon Si** (CA > 30000) **alors**

Com ← CA \* 0.1

**Sinon**

Com ← 0

**Finsi**

**Finsi**

Sal ← Com + M

**Écrire** ('Le salaire mensuel est de : ', Sal , 'DHS')

**FIN**

# Conditions composées

**Une condition composée est une condition formée de plusieurs conditions simples reliées par des opérateurs logiques: ET, OU et NON**

*Exemples :*

- x compris entre 2 et 6 :  $(x \geq 2) \text{ ET } (x \leq 6)$
- n divisible par 3 ou par 2 :  $(n \text{ Mod } 3=0) \text{ OU } (n \text{ Mod } 2=0)$

# Instructions à structure alternative

## Exercice :

**Écrire l'algorithme qui permet d'afficher le jour de la semaine d'un nombre donné.**

Par exemple, si l'utilisateur entre le nombre 2, l'algorithme affichera « MARDI ».

## Instructions à structure alternative *Selon*

Lorsque l'on doit comparer une même variable avec plusieurs valeurs, comme par exemple :

```
si a=1 alors
```

```
    instruction1
```

```
sinon si a=2 alors
```

```
    instruction2
```

```
sinon si a=4 alors
```

```
    instruction4
```

```
sinon
```

```
    ...
```

```
    finsi
```

```
finsi
```

```
finsi
```

On peut remplacer cette suite de si par l'instruction *Selon*.

## Instructions à structure alternative *Selon*

**L’instruction *Selon cas* constitue une structure alternative à la forme en bloc [Si ... Alors ...Sinon...] et permet de formuler de manière plus simple le choix du groupe d’instructions.**



# Instructions à structure alternative *Selon*

## Syntaxe :

**Selon** variable **Faire**

**Cas** Valeur 1:

*Actions 1*

**Cas** Valeur 2, Valeur3, Valeur 4:

*Actions 2*

**Cas** Valeur 5 à Valeur 7:

*Actions 3*

..

**Cas** valeur n:

*Actions n*

**Sinon**

*Actions n+1*

**FinSelon**

**Valeur 1, . . . , Valeur n** sont des constantes de type scalaire (**entier ou caractère**)  
action *i* est exécutée si **variable = valeur i** (on quitte ensuite l'instruction cas)  
**Action n+1** est exécutée si quelque soit *i*, **variable ≠ Valeur i**

## Instructions à structure alternative *Selon*

### Exercice :

**Écrire l'algorithme qui permet de déterminer le nombre de jours d'un mois d'une année donnée**

## Instructions à structure alternative *Selon*

....**Reste de l'Algorithme**

**Selon M Faire**

**Cas 2:**

Action 1

**Cas 1, 3, 5, 7, 8, 10 , 12:**

Action 2

**Cas 4, 6, 9, 11:**

Action 3

**Sinon**

**Écrire ( 'Attention : Mois Incorrect ' )**

**FinSelon**

**FIN**

## **Instructions à structure alternative Selon**

### **Exercice :**

**Un club de sport désire automatiser sa gestion. Les tarifs annuels d'inscription sont définis ainsi :**

**De 0 à 2 ans ne sont pas autorisés à s'inscrire**

**De 3 à 5 ans : gratuit**

**De 6 à 11 ans: 1000 DHS**

**De 12 à 26 ans : 1500 DHS**

**Plus de 26 ans : 2000 DHS**

## Instructions à structure alternative Selon

....Reste de l'Algorithme

**Selon** age **Faire**

**Cas** 1, 2:

**Écrire** ('Vous n'êtes pas autorisé à vous inscrire')

**Cas** 3..5:

**Tarif** ← **0**

**Cas** 6..11:

**Tarif** ← **1000**

**Cas** 12..25:

**Tarif** ← **1500**

**Sinon**

**Tarif** ← **2000**

**FinSelon**

**Écrire** (' le montant à payer en DHS est : ', Tarif)

**FIN**

# Instructions à structure alternative en C

## Traduction la structure alternative : SI

Syntaxe en algorithme	Syntaxe en C
<b>Si</b> condition <b>Alors</b>  instructions  <b>FinSi</b>	<b>if</b> ( condition ) { instructions }
<b>Si</b> condition <b>Alors</b>  instructions  <b>Sinon</b>  instructions  <b>FinSi</b>	<b>if</b> ( condition ) { instructions } <b>else</b> { instructions }

# Instructions à structure alternative en C

## Traduction la structure alternative : SI

### Structures alternatives imbriquées

Syntaxe en algorithmme ( <i>si –sinon si...</i> )	Syntaxe en C ( <i>if - else - ... – else</i> )
<b>Si</b> condition 1 <b>alors</b> instructions A <b>Sinon si</b> condition 2 <b>alors</b> instructions B <b>sinon si</b> condition 3 <b>alors</b> instructions C <b>sinon</b> instructions D <b>finsi</b> <b>finsi</b> <b>FinSi</b>	<b>if</b> (condition 1 ) <b>instructions A</b> <b>else if</b> (condition 2) <b>instructions B</b> <b>else if</b> (condition 3) <b>instructions C</b> <b>else</b> <b>instructions D</b>

# Les opérateurs conditionnels en C

Le langage C possède une paire d'opérateurs qui peut être utilisée comme alternative à **if - else** et qui a l'avantage de pouvoir être intégrée dans une expression:

## *Les opérateurs conditionnels*

**<expr1> ? <expr2> : <expr3>**

- Si **<expr1>** fournit une valeur différente de zéro, alors la valeur de **<expr2>** est fournie comme résultat
- Si **<expr1>** fournit la valeur zéro, alors la valeur de **<expr3>** est fournie comme résultat

## *Exemple*

La suite d'instructions

```
if (A>B)
    MAX=A;
else
    MAX=B;
```

Peut être remplacée par: **MAX = (A > B) ? A : B;**



# Instructions à structure alternative en C

## Traduction la structure alternative : Selon

Syntaxe en algorithme	Syntaxe en C
<p><b>Selon</b> variable <b>Faire</b></p> <p><b>Cas</b> Valeur 1: <i>Actions 1</i></p> <p><b>Cas</b> Valeur 2, Valeur3, Valeur 4: <i>Actions 2</i></p> <p><b>Cas</b> Valeur 5 à Valeur 7: <i>Actions 3</i></p> <p>..</p> <p><b>Cas</b> valeur n: <i>Actions n</i></p> <p><b>Sinon</b> <i>Actions n+1</i></p> <p><b>FinSelon</b></p>	<pre>switch (Variable) {   case Valeur1 :     Liste d'instructions;     break;   case Valeur2 :     Liste d'instructions;     break;   case Valeurs... :     Liste d'instructions;     break;   default:     Liste d'instructions; }</pre>

**Valeur 1, . . . , Valeur n** sont des constantes de type scalaire (**entier ou caractère**)