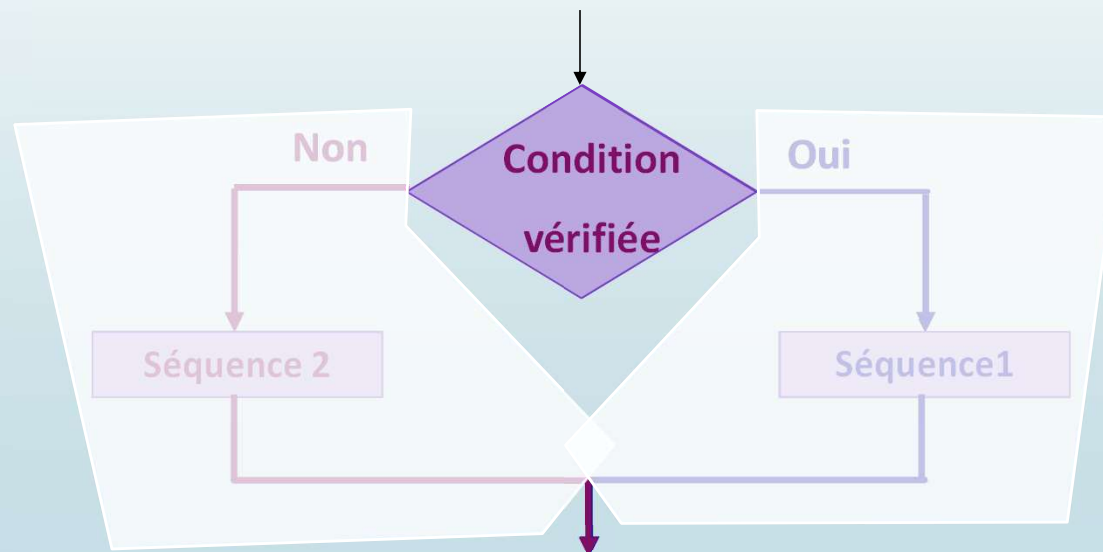


# Chapitre 2 : Structures de contrôle conditionnelles

1

## Les structures alternatives

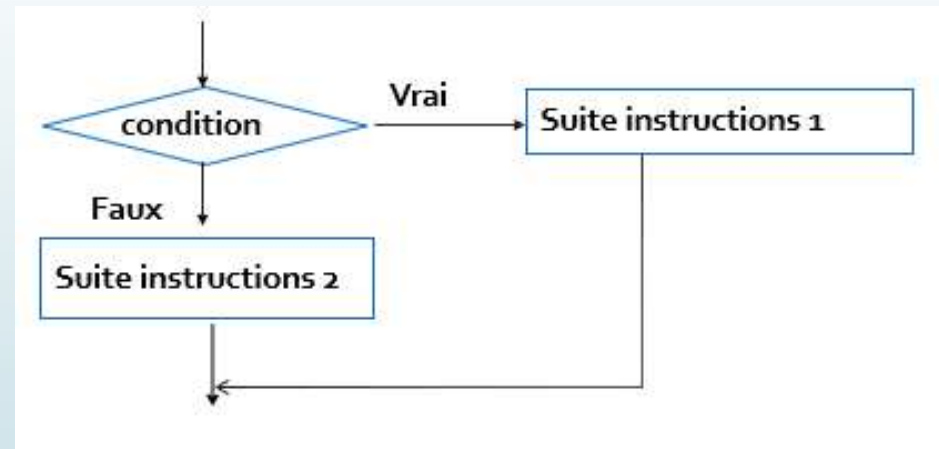
- **Problème:** afficher un message précisant que la valeur d'une variable est positive ou négative
- **Solution:** besoin d'une structure pour effectuer des choix dans le traitement réalisé
- Cette structure s'appelle la structure alternative



## Les structures alternatives

- Les structures alternatives (conditionnelles) permettent de choisir une instruction ou un bloc d'instructions selon qu'une condition est vérifiée ou non
- Syntaxe:

**SI** (*condition*) **ALORS**  
    *bloc 1 d'instructions*  
**SINON**  
    *bloc 2 d'instructions*  
**FINSI**



- Si la condition est **vraie**, le *bloc1 d'instructions* sera exécuté et le bloc2 sera ignoré
- Si la condition est **fausse**, seul le *bloc2 d'instructions* sera exécuté

## Les structures alternatives

- **Exemple1:** Ecrire un algorithme permettant d'afficher le signe d'un nombre saisi par l'utilisateur

**Algorithme SigneNombre**

**Variable x : réel**

**Début**

**Ecrire** (" Entrez un nombre : ")

**Lire** (x)

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

**Ecrire** ("le nombres est positif ")

**Sinon**

**Ecrire** ("le nombre est négatif")

**FinSi**

**Fin**

## Les structures alternatives

- La **structure alternative** peut prendre une autre forme possible où la partie ***Sinon*** est absente
- Elle s'écrit dans ce cas :

```
SI (condition) ALORS  
    bloc d'instructions  
FINSI
```

- **Exemple:** Dans un programme de calcul du montant d'une facture, on applique une remise de 10% si le montant dépasse 5000 Dhs. Nous écrivons :

```
SI (montant > 5000) ALORS  
    montant ← montant * 0.9  
FINSI
```

## Les structures alternatives

- **Exemple2:** Ecrire un algorithme permettant d'afficher la valeur absolue d'un nombre

### Algorithme ValeurAbsolue\_1

**Variable** x : réel

**Début**

Ecrire(" Entrez un nombre : ")

Lire(x)

**Si** ( $x < 0$ ) **alors**

Ecrire("la valeur absolue de ", x, " est: ", -x)

**Sinon**

Ecrire("la valeur absolue de ", x, " est: ", x)

**FinSi**

**Fin**

## Les structures alternatives

- **Exemple2:** Ecrire un algorithme permettant d'afficher la valeur absolue d'un nombre

```
Algorithme ValeurAbsolue_2  
Variables x, absx : réel  
Début  
    Ecrire(" Entrez un nombre : ")  
    Lire(x)  
    absx ← x  
    Si (x < 0) alors  
        absx ← (-1)* x  
    FinSi  
    Ecrire("la valeur absolue de ", x, " est: ", absx)  
Fin
```

## Les structures alternatives

- **Exercice:** Ecrire un algorithme permettant d'afficher si un nombre saisi par l'utilisateur est pair ou impair

**Algorithme PairOuImpair**

**Variable** x : Entier

**Début**

Ecrire(" Entrez un nombre : ")

Lire(x)

**Si** ( $x \bmod 2 = 0$ ) **alors**

Ecrire(x , "est pair ")

**Sinon**

Ecrire(x , "est impair ")

**FinSi**

**Fin**



## Les conditions simples

- Une condition simple consiste en une comparaison entre deux expressions de même type
- Cette comparaison s'effectue avec des opérateurs de comparaison dont voici la liste:

Opérateur	Signification
=	Egal
< >	Différent
<	Inférieur
<=	Inférieur ou égal
>	Supérieur
>=	Supérieur ou égal

## Les conditions simples

- Une condition complexe est formée de plusieurs conditions simples reliées entre elles par des opérateurs logiques:

**ET, OU, OU exclusif (XOR) et NON**

- Exemples :

✓ A compris entre -4 et 1 :

✓ n divisible par 5 ou par 7 :

✓ Au moins deux valeurs sont identiques parmi a, b et c :

- L'évaluation d'une condition composée se fait selon des règles présentées généralement dans les tables de vérité

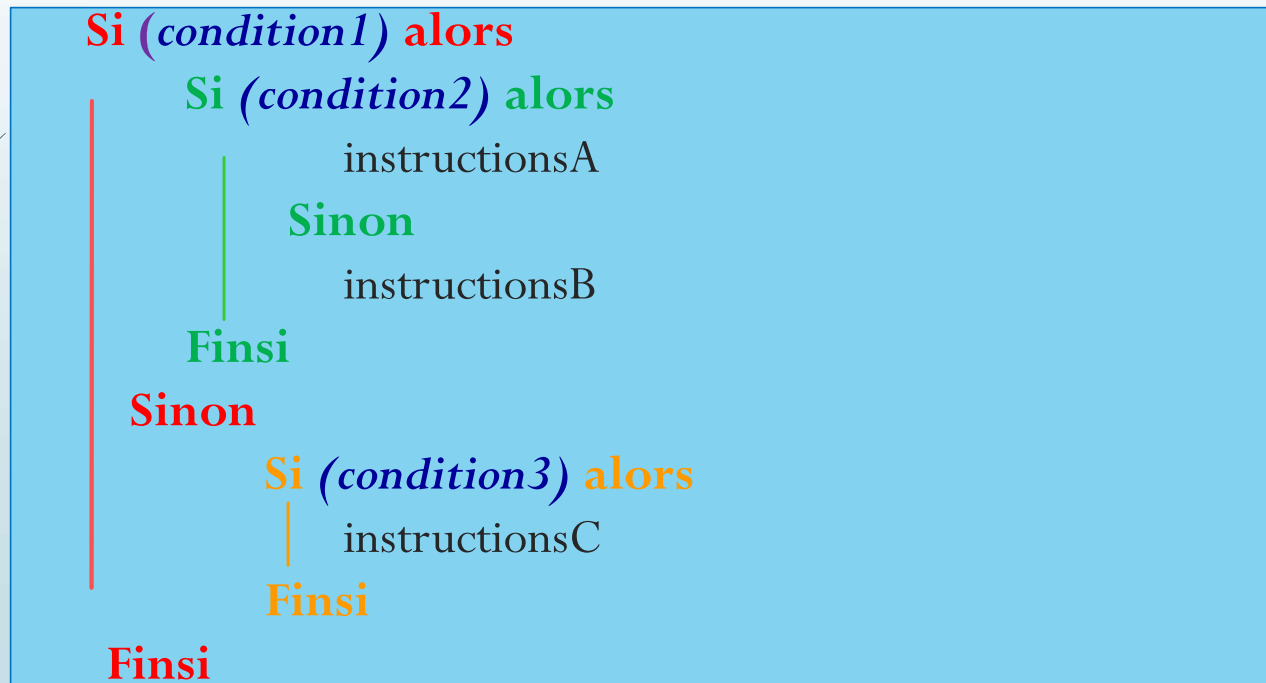
## Les structures alternatives imbriquées

► **Exemple** : Ecrire un algorithme qui donne l'état de l'eau selon sa température

```
Algorithme EtatEau
VARIABLE Temp : ENTIER
Debut
    ECRIRE ("Entrez la température de l'eau :")
    LIRE (Temp)
    Si (Temp <= 0) Alors
        Ecrire ("C'est de la glace")
    Sinon
        Si (Temp < 100) Alors
            Ecrire ("C'est du liquide")
        Sinon
            Ecrire ("C'est de la vapeur")
        Finsi
    FinSi
Fin
```

## Les structures alternatives imbriquées

- Une structure alternative peut contenir à son tour une autre structure alternative. Dans ce cas, on parle de structures alternatives imbriquées



## Les structures alternatives imbriquées

- **Exercice:** Ecrire un algorithme qui détermine si un entier saisi par l'utilisateur est positif, négatif ou nul

```
Algorithme SigneEntier (V1)
Variable a : ENTIER
Debut
    ECRIRE("Saisir un entier:")
    LIRE(a)
    Si (a > 0) Alors
        Ecrire(" Nombre positif ")
    Sinon
        Si (a < 0) Alors
            Ecrire(" Nombre négatif ")
        Sinon
            Ecrire(" Valeur nulle ")
        Finsi
    FinSi
Fin
```

## Les structures alternatives imbriquées

- **Exercice:** Ecrire un algorithme qui détermine si un entier saisi par l'utilisateur est positif, négatif ou nul

```
Algorithme SigneEntier (V2)
Variable a : ENTIER
Debut
    ECRIRE("Saisir un entier:")
    LIRE(a)
    Si (a > 0) Alors
        Ecrire(" Nombre positif ")
    FinSi
    Si (a < 0) Alors
        Ecrire(" Nombre négatif ")
    FinSi
    Si (a = 0) Alors
        Ecrire(" Valeur nulle ")
    Finsi
Fin
```

## Exercice

- Q1. Simuler l'exécution de l'algorithme ci-dessous :

**Algorithme maxdeux**

**Variables** nb1, nb2 : entier maxi :  
entier

**Début**

écrire('donner deux entiers ')

lire(nb1, nb2)

**Si** nb1 > nb2 **alors**

maxi ← nb1

**sinon**

maxi ← nb2

**finsi**

écrire('Le plus grand des deux nombres ', nb1, ' et ', nb2, 'est : ', maxi)

**Fin**

## Exercice

- Q2. Les deux algorithmes suivants sont-ils équivalents ?

### Algorithme maxdeux

**Variables** nb1, nb2 : entier

maxi : entier

#### Début

écrire('donner deux entiers ')

lire(nb1, nb2)

**Si** nb1 > nb2 **alors**

maxi ← nb1

**sinon**

maxi ← nb2

**finsi**

écrire('Le plus grand des deux nombres  
, nb1, ' et ', nb2, 'est : ', maxi)

**Fin**

### Algorithme maxdeux-2

**Variables** nb1, nb2 : entier

maxi : entier

#### Début

écrire('donner deux entiers ')

lire(nb1, nb2)

maxi ← nb1

**Si** nb2 > maxi **alors**

maxi ← nb2

**finsi**

écrire('Le plus grand des deux nombres ',  
nb1, ' et ', nb2, 'est : ', maxi)

**Fin**



## Exercice

Q3. Les deux algorithmes suivants sont-ils équivalents ? Pour vérifier la réponse réaliser trois simulations de chaque algorithme lorsque l'utilisateur donne la valeur 5, puis la valeur 0, puis la valeur -5.

### Algorithme test\_1

**Variables** nb : entier

**Début**

écrire("Donner un nombre entier")

lire (nb)

**si** nb <= 0 **alors**

nb ← nb + 5

**sinon**

nb ← nb - 5

**finsi**

écrire("maintenant le nombre vaut : ", nb)

**Fin**

### Algorithme test\_2

**Variables** nb : entier

**Début**

écrire("Donner un nombre entier")

lire (nb)

**si** nb <= 0 **alors**

nb ← nb + 5

**finsi**

**si** nb > 0 **alors**

nb ← nb - 5

**finsi**

écrire ("maintenant le nombre vaut : ", nb)

**Fin**

## La structure Selon

► Utilisée lorsqu'on doit comparer une même variable avec plusieurs valeurs

► Exemple

```
Si (a=1) Alors   instruction1  
  
Sinon Si (a=2) Alors   instruction2  
  
Sinon Si (a=3) Alors   instruction3  
  
Sinon Si (a=4) Alors   instruction4  
  
....      .....  
  
Sinon   Instruction N  
  
FinSi
```

## La structure Selon

- Syntaxe

```
SELON   Variable  
    Cas valeur1 : action1  
    Cas valeur2 : action2  
    ...  
    Cas valeurN : actionN  
    SINON : AutreAction  
FIN SELON
```

- Si *Variable* est égale à *valeuri*, on exécute *actioni* et on passe à la suite de l'algorithme. Sinon on exécute *AutreAction* et on passe à la suite de l'algorithme
- Cette structure conditionnelle est appelée aussi *à choix multiple* ou *sélective* car elle sélectionne entre plusieurs choix à la fois, et non entre deux choix alternatifs

## La structure Selon

**Exercice:** Ecrire un algorithme qui demande à l'utilisateur de saisir un entier entre 1 et 7 et affiche le jour de la semaine correspondant

**Algorithme JourSemaine**

**Variable a : ENTIER**

**Début**

ECRIRE(" Saisir un entier: ")

LIRE(a)

**SELON (a)**

**Cas 1** : Ecrire(" Lundi ")

**Cas 2** : Ecrire(" Mardi ")

**Cas 3** : Ecrire(" Mercredi")

**Cas 4** : Ecrire(" Jeudi ")

**Cas 5** : Ecrire(" Vendredi ")

**Cas 6** : Ecrire(" Samedi")

**Cas 7** : Ecrire(" Dimanche ")

**SINON** : Ecrire(" valeur erronée ")

**FIN SELON**

**Fin**

## Exercices

1. Étant donnés 2 nombres  $a$  et  $b$ , écrire l'algorithme qui résout l'équation du premier degré:  $ax + b = 0$ .
2. Écrire l'algorithme qui lit 3 nombres et qui teste si l'un de ces derniers est égal à la somme des deux autres. Si un tel nombre existe on l'affiche, sinon on affiche un message.
3. Ecrire un algorithme qui lit trois entiers  $A$ ,  $B$  et  $C$ , et affiche le plus grand.

## Exercices

4. Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !
5. A partir d'un montant lu, on détermine un montant net par application d'une remise de :
  - 1% si le montant est compris entre 2000 et 5000 Dhs (valeurs comprises)
  - 2 % si le montant est supérieur à 5000 Dhs.