

# Chapitre 4 – Les structures alternatives

## I. Introduction

Contrairement au traitement séquentiel, La structure alternative ou conditionnelle permet d'exécuter ou non une série d'instruction selon la valeur d'une condition.

## II. Structure d'un test

Il y'a deux formes possibles pour un test :

- La forme la plus simple :

**Si Booléen Alors**

Instruction 1

Instruction 2

...

**Finsi**

- La forme la plus complexe :

**Si Booléen Alors**

Instruction 1

Instruction 2

...

**Sinon**

Instruction 3

Instruction 4

...

**Finsi**

Un **booléen** est une **expression** dont la valeur est VRAI ou FAUX. Cela peut donc être (il n'y a que deux possibilités) :

- Une **variable** (ou une expression) de type booléen
- Une **condition**

**Exemple :**

Allez tout droit jusqu'au prochain magasin

**Si** la rue est bloqué **Alors**

Tournez à droite

Avancez

Prenez la 2<sup>ème</sup> rue à gauche

**Sinon**

Continuez tout droit

Prenez la 3<sup>ème</sup> rue à droite

**Finsi**

### III. Les conditions

Une condition est une comparaison. Elle est composée de trois éléments :

- Valeur
- **Opérateur de comparaison**
- Une autre valeur

Les valeurs peuvent être de n'importe quel type (numérique, caractères...).

Les opérateurs de comparaison possibles sont :

- Egal à : =
- Différent de : <>
- Strictement plus petit : =<
- Strictement plus grand : >=
- Plus petit ou égal à : <=
- Plus grand ou égal à : >=

**Exemple :**

**Algorithme** exemple

**Variables**

A, B : entier

**Début**

Ecrire("Donner la valeur de A : ")

Lire(A)

Ecrire("Donner la valeur de B : ")

Lire(B)

Si A > B Alors

Ecrire("La valeur de A est plus grand que B ")

Sinon

Ecrire("La valeur de A est plus petit que B ")

Finsi

**Fin**

## **IV. Exercices**

### **1.1. Exercice 1**

Écrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif.

### **1.2. Exercice 2**

Ecrire un algorithme qui demande un nombre entier à l'utilisateur, puis qui teste et affiche s'il est divisible par 3

### **1.3. Exercice 3**

Ecrire un algorithme qui permet de calculer la division entre deux nombres saisis par l'utilisateur. Afficher le message « impossible de faire la division » si le dénominateur égale à 0.

### **1.4. Exercice 4**

Écrire un algorithme qui permet la résolution d'une équation du premier degré (une équation sous la forme  $ax+b=0$ )

### **1.5. Exercice 5**

Ecrire un algorithme permettant de saisir deux nombres ainsi que la lettre représentant l'opération (s pour la somme, p pour produit).

## **V. Conditions composés**

Certains problèmes exigent de formuler des conditions qui ne peuvent pas être exprimée sous la forme simple.

Dans le cas d'une condition composée, on utilise un opérateur logique. Les opérateurs logiques que nous avons à notre disposition 3 opérateurs logiques :

- **ET** : Pour qu'une "condition1" **ET** "condition2" soit vraie. Il faut que la condition1 soit vraie et même la condition2 soit vraie.

Condition 1	Condition2	Condition1 ET Condition2
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Faux	Faux	Faux

- **OU** : Pour qu'une "condition1" **OU** "condition2" soit vraie. Il faut que l'un des deux conditions soit vraie.

Condition 1	Condition2	Condition1 OU Condition2
Vrai	Vrai	Vrai
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Faux	Faux	Faux

- **NON** : cet opérateur inverse une condition :
  - NON(condition1) est vraie si la condition1 est faux.
  - NON(condition1) est faux si la condition 1 est vraie.

## Exemple :

**Algorithme** exemple

**Variables**

A : entier

**Début**

Ecrire("Donner la valeur de A : ")

Lire(A)

Si  $A > 0$  ET  $A < 10$  Alors

Ecrire("La valeur de A est entre 0 et 10 ")

Sinon

Ecrire("La valeur de A n'est pas entre 0 et 10 ")

Finsi

**Fin**

## VI. Exercices

### 1.1. Exercice 1

Écrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif. Attention toutefois, on ne doit pas calculer le produit !

### 1.2. Exercice 2

Ecrire un algorithme qui demande à l'utilisateur de saisir un caractère et affiche le message « **Entre A et M** » si le caractère saisi est entre A et M sinon il va afficher « **Le caractère saisi n'est pas entre A et M** »

### 1.3. Exercice 3

Écrire un algorithme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de la somme de A et B sans faire l'addition.

## VII. Les tests imbriqués

Les tests peuvent avoir un degré quelconque d'imbrications

**Si condition1 alors**

**Si condition2 alors**

instructionsA

**Sinon**

instructionsB

**Finsi**

**Sinon**

**Si condition3 alors**

instructionsC

**Finsi**

**Finsi**

Exemple :

**Algorithme** Test\_signe

**Variables** n : entier

**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**

## VIII. Exercices

### 1.4. Exercice 1

Donner les valeurs des a, b, c et d après chaque instruction de l'algorithme suivant :

**Algorithme** Penible

**Variables**

a, b, c, d : entier

**Début**

Ecrire("Saisir trois valeurs :")

Lire(a)

Lire(b)

Lire(c)

$d \leftarrow 2$

Si a = 3 Alors

$a \leftarrow 2$



```

        b ← (a + c) * d
    Sinon
        Si a = 0 Alors
            a ← 2
            c ← d * a
        Sinon
            c ← 2 + b
            d ← b - a
        Finsi
    Finsi
    Ecrire("A = ", a, "B = ", b, "C = ", c)

```

**Fin**

Cas 1 : a = 3, b = 2, c = 1

Cas 2 : a = 0, b = 4, c = 5

Cas 3 : a = 1, b = 5, c = 3

Après instruction	Valeurs des variables			
	a	b	c	d
i1				
i2				
i3				
i4				
i5				
i6				
i7				
i8				
i9				

### 1.5. Exercice 2 :

Écrire un algorithme qui permet la résolution d'une équation du second degré (une équation sous la forme  $ax^2+bx+c=0$ )

### 1.6. Exercice 3 :

Calculer le lendemain d'une journée donnée (jour, mois, année)" On ne tiendra pas compte ici des années bissextiles, le mois de février aura toujours 28 jours.

### 1.7. Exercice 4 :

Calculer la durée d'un trajet connaissant l'heure de départ et d'arrivée". On se contente des heures et des minutes, la durée totale ne dépassera jamais 24 heures.

### 1.8. Exercice 5 :

Le prix de photocopies dans une reprographie varie selon le nombre demandé: 0,5 DH la copie pour un nombre de copies inférieur à 10, 0,4DH pour un nombre compris entre 10 et 20 et 0,3DH au-delà.

Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées, qui calcule et affiche le prix à payer

## IX. Les structures alternatives Multiples

Pour éviter un lourd algorithme contenant plusieurs briques de **Si..Finsi** pour plus de lisibilité on a adopté une structure plus simple pour traiter plusieurs conditions à la fois cette structure est : **Selon .. FinSelon**

**Exemple :**

```
Selon (variable_a_tester /condition)
  cas cas1 : BlocInstruction1
  cas cas2 : BlocInstruction2
```

.....  
**Autrement/sinon** : BlocInstruction  
**Finselon**

**Exemple : Exercice jours de la semaine**