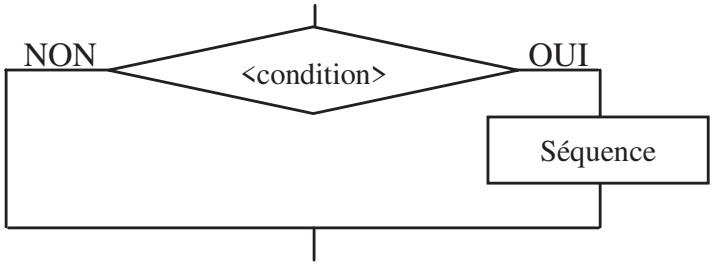
# Chapitre 3 : Les structures conditionnelles

## 1. Introduction

Les algorithmes vus précédemment sont exécutés séquentiellement. Les ruptures des séquences peuvent être effectuées par des structures de contrôle classées en deux catégories : les structures conditionnelles et les boucles. Les structures conditionnelles (simples, composées et multiples) sont aussi appelées structures alternatives, structures de choix ou les tests.

## 2. Structure conditionnelle simple

Format général : Si <condition> Alors <Séquence>.
Cette opération est lue comme suit : Si la condition est vérifiée (VRAI) alors la séquence d'opérations s'exécute.
La condition est une expression logique qui retourne la valeur VRAI ou FAUX. L'expression peut être simple (condition simple) ou composée (plusieurs conditions composées avec des opérateurs logiques ET, OU et NON).
La séquence peut contenir une ou un ensemble d'opérations. Si la séquence contient plusieurs opérations alors elles sont séparées par des points virgules, et mises entre début et fin. Si la séquence contient une seule opération alors les mots début et fin ne sont pas obligatoires.
La structure conditionnelle simple peut être représentée dans un organigramme comme suit :




Voyons les exemples suivants :

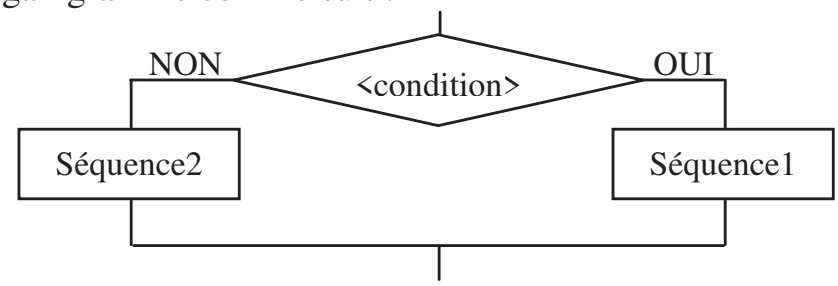
– Si (X > 10) Alors Ecrire(X) ; On affiche la valeur de X si elle est supérieure à 10.

- Si (X > 10) Alors début Ecrire(X) ; Y ← X ; fin ; On affiche la valeur de X et on affecte la valeur de X à Y, si X est supérieure à 10.
- Si (X > 10) ET (X < 15) Alors Ecrire(X) ; On affiche la valeur de X si elle est prise entre 10 et 15.
- Si (X > 10) Alors Si (X < 15) Alors Ecrire(X) ; Cet exemple est équivalent à l'exemple précédent, sauf que cette fois-ci on utilise des tests imbriqués.
- Si (X < 10) Alors Si (X > 15) Alors Ecrire(x) ; Dans cet exemple, la valeur de X n'est jamais affichée car il n'y a aucun cas qui satisfait la condition.

En Pascal, la structure conditionnelle simple s'écrit sous la forme:
if <condition> then <Séquence> ;
**Exemple :** if (X > 10) then write(X);
Si la séquence contient plus d'une instruction alors le begin et end sont obligatoires pour délimiter la suite d'instructions.

## 3. Structure conditionnelle composée

Format général : Si <condition> Alors <Séquence1> Sinon <Séquence2>.
Cette opération est lue comme suit : Si la condition est vérifiée (VRAI) alors les opérations de la séquence1 sont exécutées. Dans le cas contraire, ce sont les opérations de la séquence2 qui vont être exécutées.
Les mots début et fin sont utilisés pour délimiter une séquence de plusieurs opérations.
La structure conditionnelle composée peut être représentée dans un organigramme comme suit :




Soit l'exemple suivant : Si (X > 10) Alors Ecrire(X) Sinon Ecrire('valeur non acceptée') ; Dans cet exemple, la valeur de X

est affichée si elle est supérieure à 10, sinon on affiche un message d'erreur.
En Pascal, on aura : if (x > 10) then writeln(x) else writeln ('valeur non acceptée') ;
Si nous avions une suite d'instructions dans une séquence alors on aurait dû utiliser les mots begin et end.
**Remarques :**

- En Pascal, le else n'est jamais précédé par un point virgule (;).
- Le else se rapporte toujours au if…then le plus proche. Pour casser ce rapport, il est possible d'utiliser le begin et end.
  Par exemple, dans le cas de :

```
if (x > 10) then
  begin if (x < 20) then writeln(x) ; end
else writeln ('valeur non acceptée') ;
```

  le else suit le premier if et non pas le deuxième.

## 4. Structure conditionnelle multiple

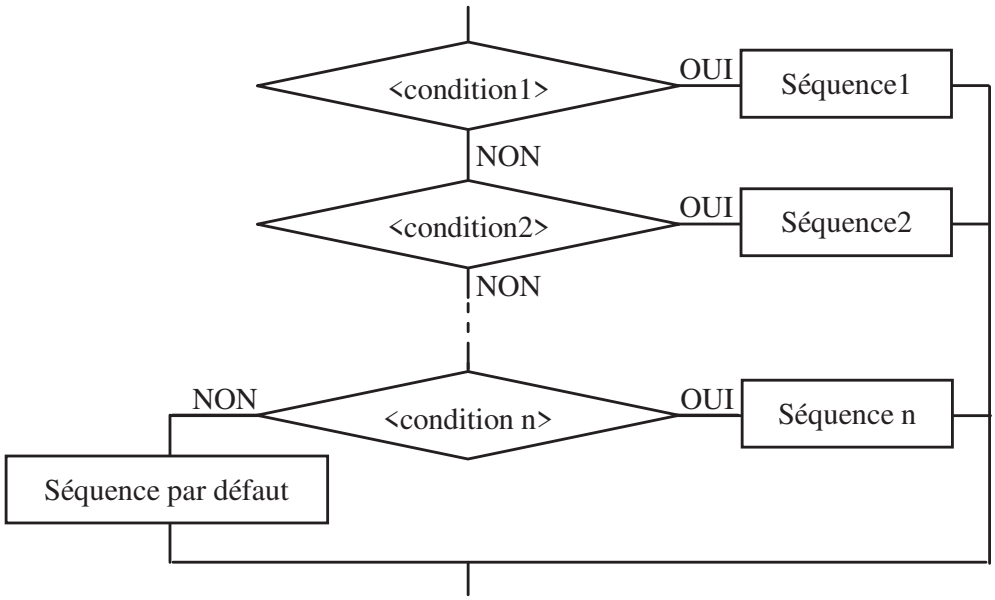
La structure conditionnelle multiple, appelée aussi l'alternative classifiée ou le choix multiple, permet de comparer un objet (variable ou expression) à toute une série de valeurs, et d'exécuter une séquence d'opérations parmi plusieurs, en fonction de la valeur effective de l'objet. Une séquence par défaut peut être prévue dans le cas où l'objet n'est égal à aucune des valeurs énumérées.
Chaque séquence est étiquetée par une valeur. Pour que cette séquence soit choisie, il faut que sa valeur soit équivalente à l'expression. La structure conditionnelle multiple se présente comme suit :

```
Cas <variable ou expression> de
  Valeur 1 : <Séquence1>
  Valeur 2 : <Séquence2>
      …
  Valeur n : <Séquence n>
  Sinon <Séquence par défaut>
fin ;
```

Ceci est équivalent à :

Si (variable ou expression = Valeur1) alors <Séquence1>
Sinon Si (variable ou expression = Valeur2) alors <Séquence2>
…
Sinon Si (variable ou expression = Valeur n) alors <Séquence n>
Sinon <Séquence par défaut>

La structure conditionnelle multiple peut être représentée dans un organigramme comme suit :




Dans l'exemple suivant, la valeur de X est affichée en lettre, si elle est égale à 1 ou 2, sinon afficher un message d'erreur :

```
Cas (X) de
  1 : Ecrire('Un') ;
  2 : Ecrire('Deux')
  Sinon Ecrire('Valeur sup à deux ou inf à un') ;
fin ;
```

En Pascal ça s'écrit :

```
case (X) of
  1 : writeln('Un');
  2 : writeln('Deux')
  else writeln('Valeur sup à deux ou inf à un') ;
end;
```

**Remarques :**

- Dans l'instruction de choix multiple, l'ordre de présentation ne change rien.
- L'expression et les valeurs à choisir doivent être de même type.
- Si une séquence est composée d'un ensemble d'instructions, elles doivent être prises entre begin et end.

## 5. Le branchement

Il est possible d'effectuer un saut direct vers une opération en utilisant une opération de branchement de la forme aller à <étiquette>. Les étiquettes sont des adresses qui doivent être déclarées dans la partie déclaration de l'algorithme.

**Exemple :**

```
Algorithme afficher_nbr_positif ;
  Etiquettes 10 ;
  Variables
    I : entier ;
Début
  Lire (I) ;
  Si (I<0) Alors aller à 10 ;
  Ecrire (I) ;
  10 : Ecrire('Merci') ;
Fin.
```

En Pascal ça s'écrit :

```
program afficher_nbr_positif ;
  label 10 ;
  var    I : integer ;
begin
  readln (I);
  if (I<0) then goto 10 ;
  writeln (I) ;
  10 : writeln('Merci') ;
end.
```

On note qu'il est déconseillé d'utiliser l'instruction de branchement, et cela pour réduire la complexité des programmes en termes de temps.