

Les structures de contrôles (les tests)

Les structures de contrôle sont des actions composées qui permettent de spécifier des traitements complexes.

Si nous voulons conditionner l'exécution d'un bloc d'actions par la vérification d'une condition nous utilisons l'alternative (ou structure de contrôle conditionnelle)

Il existe deux formes pour l'alternative:

L'alternative simple:

Syntaxe :

```
SI Condition ALORS  
    action1  
    action2  
    .  
    .  
FINSI;
```

Une **condition** est une expression logique

Ce qui signifie : si la condition est vraie le bloc d'actions est exécuté, sinon il est ignoré.

En fortran

```
If (condition) then  
    action1  
    action2  
endif
```

Exemple :

```
    Si il fait trop chaud ET il ne pleut pas Alors  
        Ouvrir la fenêtre  
    Fin Si
```

L'alternative complète:

```
SI condition ALORS  
    action1  
    .  
Sinon  
    action2
```

```
.  
.br/>FINSI
```

En fortran

```
If (condition) then  
    action1  
else  
    action2  
endif
```

Ce qui signifie: si la condition est vraie le premier bloc d'actions est exécuté, et si elle est fausse le deuxième bloc d'actions est exécuté

Exemple :

```
Si il fait trop chaud ET il ne pleut pas Alors  
    Ouvrir la fenêtre  
Sinon  
    Fermer la fenêtre  
Fin Si
```

Exemples:

<pre>Si X > 0 alors Y ← X B ← 0 Finsi</pre>	<pre>Si (N >= 0) et (T = X) alors Val ← T + N écrire (Val) Sinon Val ← T Finsi</pre>
--------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------

En Fortran

<pre>If(X > 0) then Y = X B = 0 endif</pre>	<pre>if ((N >= 0) .and. (T = X)) then Val = T + N Print*,Val endif Val = T endif</pre>
---------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------

Remarques:

- ✓ La condition est une expression du type booléen (logique).
- ✓ Le bloc d'actions écrit après le Si et le Sinon peut contenir une alternative, on dit alors que les alternatives sont imbriquées.

Tests imbriqués

L'instruction SI ouvre deux voies, correspondant à deux traitements différents. Mais il y a des tas de situations où deux voies ne suffisent pas.

Par exemple, un programme devant donner l'état de l'eau selon sa température doit pouvoir choisir entre trois réponses possibles (solide, liquide ou gazeuse).

```
Var      Temp : Entier ;
Début
  Ecrire ("Entrez la température de l'eau :") ;
  Lire (Temp) ;
  Si Temp =< 0 Alors
    Ecrire ("C'est de la glace") ;
  FinSi ;
  Si (Temp > 0 Et Temp < 100) Alors
    Ecrire ("C'est du liquide") ;
  Finsi ;
  Si Temp > 100 Alors
    Ecrire ("C'est de la vapeur") ;
  Finsi ;
Fin
```

On constate que c'est un peu laborieux. Les conditions se ressemblent plus ou moins, et surtout on oblige la machine à examiner trois tests successifs alors que tous portent sur une même chose, la température de l'eau (la valeur de la variable Temp). Il serait ainsi bien plus rationnel d'imbriquer les tests de cette manière :

```
Var      Temp : Entier ;
Début
  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
```

Nous avons fait des économies : au lieu de devoir taper trois conditions, dont une composée, nous n'avons plus que deux conditions simples. Mais aussi, et surtout, nous avons fait des économies sur le temps d'exécution de l'ordinateur. Si la température est inférieure à zéro, celui-ci écrit dorénavant « C'est de la glace » et passe directement à la fin, sans être ralenti par l'examen d'autres possibilités (qui sont forcément fausses).

Exercice:

Donnez un algorithme qui permet de faire la division de deux nombres réels.

```
Algorithme division
Variable A, B : réel
Début
Lire (A, B)
Si B = 0 alors
    écrire ('division impossible')
Sinon
    écrire (A / B)
Finsi
Fin
```

Choix multiple :

Le choix multiple est une autre forme de l'action sélective. Il sert à choisir une action entre plusieurs actions en fonction d'une condition.

Syntaxe

```
                SELON variable faire
début
                val 1 : action 1;
                val 2 : action 2;
                ...
                val N : action N
                Sinon : action
fin;
```

Exemple :

```
Cas choix De
    '+' : resultat ←  $A+B$  ;
    '-' : resultat ←  $A-B$  ;
    '*' : resultat ←  $A*B$  ;
    '/' : resultat ←  $A/B$  ;
Sinon
    Ecrire('erreur'),
Fin
```