

## Résumé série TP N°04 :

### Structures de contrôle conditionnel

Ces structures sont utilisées pour déterminer l'exécution d'un bloc d'instructions : soit ce bloc est exécuté, soit il ne l'est pas. Elles servent également à choisir entre l'exécution de deux blocs différents. Nous avons deux types de structures conditionnelles :

#### 1. Test alternatif simple

Un test simple contient un seul bloc d'instructions. Selon une condition (expression logique), on décide si le bloc d'instructions est exécuté ou non. Si la condition est vraie, on exécute le bloc, sinon on ne l'exécute pas.

La syntaxe d'un test alternatif simple est la suivante :

Algorithme	Langage C
<pre><b>si</b> &lt;Condition&gt; <b>alors</b>         &lt;bloc_instructions_si&gt; ; <b>finsi</b> ;</pre>	<pre><b>if</b> (Condition) {   &lt;bloc_instructions_if&gt; ; }</pre>

**Remarque :** Dans le langage C, un bloc est délimité par deux accolades { et }. Si le bloc contient une seule instruction, les accolades { et } sont facultatives (on peut les enlever).

#### 2. Test alternatif double

Un test double contient deux blocs d'instructions : on est amené à choisir entre le premier bloc ou le second. Cette décision est réalisée sur une condition (expression logique ou booléenne) qui peut être vraie ou fausse. Si la condition est vraie on exécute le premier bloc, sinon on exécute le second.

La syntaxe d'un test alternatif simple est la suivante :

Algorithme	Langage C
<pre><b>si</b> &lt;Condition&gt; <b>alors</b>         &lt;bloc_instructions_si&gt; <b>sinon</b>         &lt;bloc_instructions_sinon&gt; ; <b>finsi</b> ;</pre>	<pre><b>if</b> (Condition) {   &lt;bloc_instructions_if&gt; ; } <b>else</b> {   &lt;bloc_instructions_else&gt; ; }</pre>

### Remarques :

- En langage C, il ne faut pas mettre de point-virgule après la condition (erreur logique).
- Dans l'exemple précédent, les accolades { } du **if** et du **else** peuvent être omises puisqu'il y a une seule instruction dans les deux blocs.

Nous avons aussi, les **structures conditionnelles doubles et imbriquées** :

Un test double et imbriqué, tout comme un test double, contient deux blocs instructions avec au moins un des deux blocs (bloc Si et/ou bloc Sinon) est composé d'une instruction de condition simple ou double. Donc un test double et imbriqué contient au moins trois blocs d'instructions avec au moins deux conditions.

La syntaxe d'un test alternatif double imbriqué avec trois blocs d'instructions est :

Algorithme	Langage C
<pre><b>si</b> &lt;Condition&gt; <b>alors</b>   &lt;bloc_instructions_si&gt; <b>sinon</b>   <b>si</b> &lt;Condition&gt; <b>alors</b>     &lt;bloc_instructions_si&gt;   <b>sinon</b>     &lt;bloc_instructions_sinon&gt; ; <b>finsi</b> ; <b>finsi</b> ;</pre>	<pre><b>if</b> (Condition) {   &lt;bloc_instructions_if&gt; ; } <b>else</b> <b>if</b> (Condition) {   &lt;bloc_instructions_if&gt; ; } <b>else</b> {   &lt;bloc_instructions_else&gt; ; }</pre>