

## Chapitre 2 : LES STRUCTURES CONDITIONNELLES

### Problématique

Un programme en C se comporte d'une suite d'instructions qui sont exécutées les unes après les autres.

- Jusqu'ici, dans le corps d'un programme en C, toutes les lignes de code sont exécutées du début à la fin (**A**).
- Lorsqu'on écrit un programme, il arrive que l'on ait besoin de contrôler le déclenchement de l'exécution d'une ou de plusieurs instructions (**B**).

Pour résoudre ces problèmes (**A**) et (**B**), on fait alors appel aux *structures conditionnelles*. On appelle structures conditionnelles, les instructions nous permettant de tester si une ou plusieurs conditions sont vérifiées ou pas et éventuellement de donner des actions à exécuter. En C, on utilise trois (3) structures conditionnelles :

- ⇒ La structure conditionnelle **simple** : `if(){ ... }`
- ⇒ La structure conditionnelle **alternative** : `if(){ ... } ... else{ ... }`
- ⇒ La structure à **choix multiple** : `switch(){ ... }`

### I. La structure conditionnelle simple

C'est la structure de test la plus basique, on l'a retrouvé dans tous les langages de programmation. Elle permet d'exécuter une série (un bloc) d'instructions si jamais une condition est réalisée.

SYNTAXE	<pre>if ( condition(s) ) {     &lt;action 1&gt;     &lt;action 2&gt;     ...     &lt;action N&gt; }</pre>
---------	---

### Remarque :

Il est possible de définir plusieurs conditions à remplir. Dans ce cas, il faut les séparer soit par le ET logique ou le OU logique.

### Exemple

Ecrire un programme qui permet de saisir un nombre entier et de vérifier s'il est pair ou impair.

### Exercice d'application

Ecrire un programme qui permet de contrôler l'âge d'une personne. Une personne ne doit être âgée de -1an ni de +145 ans.

## II. La structure conditionnelle alternative

Elle permet de tester toutes les alternatives possibles et de donner pour chacune d'elle les actions à faire. Elle se compose de deux (2) blocs intimement liés. Ainsi, l'exécution de l'un exclut celle de l'autre.

<b>SYNTAXE</b>	<pre> if ( condition(s) ){     &lt;action 1&gt;     &lt;action 2&gt;     ...     &lt;action N&gt; } else {     &lt;action 1&gt;     &lt;action 2&gt;     ...     &lt;action N&gt; } </pre>
----------------	--

### Exemple

Pour entrer à la gendarmerie, il faut avoir un poids d'au moins 75kg. Ecrire un programme qui permet de vérifier si un candidat est accepté ou pas.

### Exercice d'application :

Ecrire un programme qui permet de saisir deux entiers, le programme détermine et affiche le minimum et le maximum.

**Remarque :** Operateur ternaire → (condition(s)) ? action1 : action2 ;

## III. La structure à choix multiple

C'est une structure qui propose une liste d'options possible. Les actions à exécuter dépendent de l'option choisie.

<b>SYNTAXE</b>	<pre> switch ( expression ) { case valeur1 :     &lt;action 1&gt;     ...     &lt;action N&gt;     break ; case valeur2 :     &lt;action 1&gt;     ...     &lt;action N&gt;     break ; ..... default :     &lt;action 1&gt;     &lt;action 2&gt;     ...     &lt;action N&gt; } </pre>
----------------	---

#### Exercice d'application :

Ecrire un programme intitulé CALCULATRICE qui permet de saisir deux nombres entiers puis un opérateur (+, -, \*, /, %). Le programme calcule et affiche le résultat de l'opération.

#### **IV. Imbrication de structure :**

C'est le fait de définir une structure conditionnelle à l'intérieur d'une autre structure conditionnelle. Dans ce cas, la structure conditionnelle la plus interne est fermée avant celle qui est la plus externe.

#### **Remarque :**

Toute structure alternative ou à choix multiples peut être transformée en une suite de structures conditionnelles simples.

#### Exercices d'application :

- Ecrire un programme qui permet de comparer deux nombres entiers.
- Ecrire un programme qui permet de saisir trois nombres puis détermine et affiche le maximum et le minimum
- Ecrire un programme qui permet de saisir une date (jour, mois et année), puis indique si la date est correcte ou pas.