

## Chapitre 2 : LES STRUCTURES CONDITIONNELLES

### Problématique

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

- Jusqu'ici, dans le corps d'un programme algorithmique, toutes les lignes de codes 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 Algorithme on utilise trois(3) structures conditionnelles :

- ⇒ La structure conditionnelle **simple**
- ⇒ La structure conditionnelle **alternative**
- ⇒ La structure à **choix multiple**

### **I. La structure conditionnelle simple**

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

<b>SYNTAXE</b>	<b><u>Si ( condition(s) ) alors</u></b> <action 1> <action 2> ... <action N> <b><u>Fin Si</u></b>
----------------	--

### **Remarque :**

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

### **Exemple**

Ecrire un programme qui permet de vérifier si un nombre 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 sont intimement liés. Ainsi, l'exécution de l'un exclut celle de l'autre.

SYNTAXE	<b><u>Si ( condition(s) ) alors</u></b>
	<action 1>
	<action 2>
	...
	<action N>
	<b><u>Sinon</u></b>
	<action 1>
	<action 2>
	...
	<action N>
	<b><u>Fin Sin</u></b>

### 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.

### 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.

SYNTAXE	<b><u>Suivant ( expression ) faire</u></b>
	<b>Valeur1 :</b>
	<action 1>
	...
	<action N>
	<b>Valeur 2 :</b>
	<action 1>
	...
	<action N>
	<b>Valeur N :</b>
	...
	<b><u>Sinon</u></b>
	<action 1>
	<action 2>
	...
	<action N>
	<b><u>Fin Suivant</u></b>

#### 1- Exemple

Ecrire un programme qui permet à un utilisateur de saisir un chiffre et le programme affiche sa valeur en lettre.

**Exercice d'application :**

Ecrire un programme intitulé CALCULATRICE qui permet de saisir deux nombres 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
- Ecrire un programme qui permet de saisir trois nombres entiers 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.