



Chapitre 2: Les structures conditionnelles

Babacar DIOP

Adresse Mail: diopbabacar4888@gmail.com

Téléphone: 777812040

Problématique

- ❖ Un programme C comporte une suite d'instructions qui sont exécutées les unes après les autres.
- ❖ Jusqu'ici, dans un programme C, 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 qui permettent d'exécuter des actions spécifiques en fonction de la véracité ou de la fausseté d'une ou plusieurs conditions.**
- ❖ En langage C 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 la retrouve dans tous les langages de programmation.
- ❖ Elle permet d'exécuter une série (un bloc) d'instructions si la condition est réalisée.

- ❖ **Syntaxe:**

```
if ( condition(s) ){  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
}
```

- ❖ Remarque: Il est possible de combiner plusieurs conditions. Dans ce cas, il faut les séparer soit par **&& (ET logique)** ou **|| (OU logique)**.
- ❖ Exercices d'application:
 - Ecrire un programme qui permet de vérifier si un nombre est pair ou impair.
 - Ecrire un programme qui permet de contrôler L'âge d'une personne. Une personne ne doit être âgée de -1 an, 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:**

```
if ( condition(s) ){  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
} else {  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
}
```

- ❖ Exercices d'application:

- 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.
- 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 possibles.
- ❖ Les actions à exécuter dépendent de l'option choisie
- ❖ **Syntaxe:**

```
Switch ( expression ) {  
  case Valeur 1:  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
  break;  
  case Valeur 2:  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
  break;  
  case Valeur N:  
    ...  
  default:  
    <action 1>  
    <action 2>  
    ...  
    <action N>  
}
```

- ❖ Exercices d'application:
 - Ecrire un programme qui permet à un utilisateur de saisir un chiffre et le programme affiche sa valeur en lettre.
 - 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 une date (jour, mois et année), puis indique si la date est correcte ou pas.
Conditions de validité:
 - Le jour doit être dans la plage valide pour le mois donné.
 - Le mois doit être compris entre 1 et 12.
 - L'année doit être positive.