



# CH2

# Les Structures de contrôle conditionnelles

1<sup>ère</sup> Année Licence GL

Kais ben Salah  
&  
Taoufik Sakka Rouis

# Les structures de contrôle conditionnelles

## 1- Introduction

En programmation, on est souvent confronté à des situations où a besoin de choisir entre 2 ou plusieurs traitements selon la réalisation ou non d'une certaine condition ; d'où la notion de traitement conditionnel.

# Les structures de contrôle conditionnelles

## 2- if-else

### Syntaxe :

```
if (<expression>
    < Bloc d'instructions 1 > ;
else
    < Bloc d'instructions 2 > ;
```

- ❑ Si l'<expression> fournit une valeur différente de 0 alors le <bloc d'instructions1> est exécuté.
- ❑ Si l'<expression> fournit la valeur 0 alors le <bloc d'instructions 2> est exécuté.
- ❑ La partie <expression> peut désigner :
  - ✓ Une variable numérique.
  - ✓ Une expression fournissant un résultat numérique.
- ❑ La partie <bloc d'instructions > peut désigner :
  - ✓ Un bloc d'instructions compris entre accolades.
  - ✓ Une seule instruction terminée par un point-virgule.

# Les structures de contrôle conditionnelles

## Exemple 1

```
if ( a>b)
    Max = a ;
else
    Max = b ;
```

## Exemple 2

```
if (A>B)
{
    aide = A ;
    A = B ;
    B = aide ;
}
else
{
    aide = B ;
    B = A ;
    A = aide ;
}
```

# Les structures de contrôle conditionnelles

## 3. if sans else

Syntaxe :

```
if (<expression>)  
{  
    < Bloc d'instructions > ;  
}
```

Exemple

```
if (nb==0)  
    printf ("%d ne peut pas être un diviseur\n", nb) ;
```

# Les structures de contrôle conditionnelles

## 4. Test Imbriqués

Syntaxe :

```
if (<expr 1>)
    <Bloc 1> ;
else
    if (<expr 2>)
        <Bloc 2> ;
    else
        .
        .
        .
    else
        if (<expr N>)
            <Bloc N> ;
        else
            <Bloc N+1> ;
```

Les expressions <expr 1 >...<expr N> sont évaluées du haut vers le bas jusqu'à ce que l'une d'elles soit différente de 0 . Le bloc d'instructions y lié est alors exécuté et le traitement de la commande est terminé.

# Les structures de contrôle conditionnelles

## Exemple

```
#include<stdio.h>
void main()
{
    int A, B ;
    printf(" Donner 2 entiers : ") ;
    scanf("%d %d",&A, &B) ;
    if(A>B)
        printf("%d est plus grand que %d\n",A,B) ;
    else
        if(A<B)
            printf("%d est plus petit que %d\n",A,B) ;
        else
            printf("%d est égal a %d\n",A,B) ;
}
```

# Les structures de contrôle conditionnelles

**Remarque** : En C une partie else est toujours liée au dernier if.

**Exemple** :

```
if(a<=b)
    if(b<=c)
        printf("Ordonné") ;
    else
        printf(" Non ordonné") ;
```



# Les structures de contrôle conditionnelles

## 5. Les opérateurs conditionnels

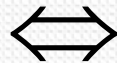
**Syntaxe :**

**<expr 1> ? <expr 2> : <expr 3> ;**

- ❑ Si <expr 1> fournit une valeur différente de 0, alors la valeur de <expr 2> est fournit comme résultat.
- ❑ Si <expr 1> fournit la valeur 0, alors la valeur de <expr 3> est fournit comme résultat.

**Exemple :**

```
if (a>b)
    Max = a ;
else
    Max = b ;
```



```
max = a>b ? a : b
```

# Les structures de contrôle conditionnelles

## 6. Test multiples (switch)

Syntaxe :

```
switch (var)
{
    case constante_1 : instruction_1; break;
    case constante_2 : instruction _2; break;
    .
    .
    case constante_N : instruction _N; break;
    default : intruccion_N+1 ;
}
```

# Les structures de contrôle conditionnelles

- `<var>` est une variable de type `int` ou `char`.
- "default " (la traduction de " autrement ") est facultatif.
- `Const_i` est une référence de cas qui représente une des valeurs autorisées de la variable `<var>` (si `<var>` est de type entier alors les références de cas doivent être des valeurs entières).
- `Instr_i` peut être simple ou composée.
- Les instructions vides sont aussi permises pour indiquer qu'aucun traitement n'est à effectuer dans ce cas.
- L'instruction `switch` prend la valeur de `<var>` et compare à chacune des étiquettes `case`, dès qu'elle trouve celle qui correspond, les instructions qui suivent sont exécutées soit jusqu'à la rencontre d'une instruction `break`, soit jusqu'à la fin du corps de l'instruction `switch`.
- Il ne faut pas oublier d'utiliser l'instruction `break` chaque fois qu'un cas donné est terminé, sinon, les instructions suivantes seraient exécutées (jusqu'au prochain `break`).

# Les structures de contrôle conditionnelles

## Exemple 1 :

```
switch(choix)
{
    case 'R' : printf("Rouge"); break;
    case 'B' : printf("Bleu");  break;
    case 'J' : printf("Jaune");  break;
}
```

# Les structures de contrôle conditionnelles

**Exemple 2 :** Écrire un programme C qui permet de saisir un nombre entre 0 et 9 et d'afficher la nature de ce nombre (zéro, pair, impair).

```
#include<stdio.h>
void main()
{   int NB ;
    printf(" Donner un entier : ") ; scanf("%d", &NB) ;
    switch(NB)
    {
        case 0 :printf("\n Nombre Zéro");break ;/*On saute les instructions suivantes*/
        case 1 :
        case 3 :
        case 5 :
        case 7 :
        case 9 : printf(" \n Nombre impair ");break ;
        case 2 :
        case 4 :
        case 6 :
        case 8 : printf(" \n Nombre pair "); break ;
        default : printf(" \n Nombre erroné ");
    }
}
```