STRUCTURE CONDITIONNELLE IF-ELSE

Structure conditionnelle if

La syntaxe de l'instruction **if** est la suivante:

```
if( condition ){
    // corps
    //instructions à exécuter si la condition est vérifiée (vrai)
}
```

L'instruction **if** évalue la condition à l'intérieur de la parenthèse **()**.

- Si la **condition** est évaluée à vrai, les instructions à l'intérieur du corps de **if** sont exécutées.
- Si la **condition** est évaluée à faux, les déclarations à l'intérieur du corps de if ne sont pas exécutées.

Exemple 1 :

```

#include < stdio.h>
int main(void){
    int a;

    printf("Saisir une valeur : ");
    scanf("%d",&a);

    // structure conditionnelle
    if(a< 0){
        printf("la valeur de a est négative ");
    }

    // fin du programme
    printf("Good bye !");

    return 0;
}
```

```
Saisir une valeur : 5
Good bye !
Saisir une valeur : -3
la valeur de a est négative
Good bye !
```

Structure conditionnelle if-else

L'instruction **if** peut avoir un bloc else optionnel. La syntaxe de l'instruction **if ... else** est la suivante:

```
?
1
2    if( condition ){
3        // corps
4        //instructions à exécuter si la condition est vraie
5    }
6    else{
7        //corps
     // instructions à exécuter si condition est fausse
    }
```

Si la **condition** est évaluée à vrai,

- les instructions à l'intérieur du corps de **if** sont exécutées.
- les instructions à l'intérieur du corps de **else** sont exclues de l'exécution.

Si la **condition** est évaluée à faux,

- les instructions à l'intérieur du corps de **else** sont exécutées.
- les instructions à l'intérieur du corps de **if** sont exclues de l'exécution.

Exemple 2 :

```
1     #include < stdio.h>
2     int main(void){
3         int a;
4
5         printf("Saisir une valeur : ");
6         scanf("%d",&a);
7
8         // structure conditionnelle
9         if(a< 0){
10            printf("la valeur de a est négative ");
11        }
12        else{
13            printf("la valeur de a est positive ou nulle ");
14        }
15
16        // fin du programme
17        printf("Good bye !");

          return 0;
      }
```

```
18
19
```

```
Saisir une valeur : 5
la valeur de a est positive ou nulle
Good bye !
Saisir une valeur : -3
la valeur de a est négative
Good bye !
```

Structure conditionnelle if - else if - else

L'instruction **if ... else** exécute deux codes différents selon que la condition est **vrai** ou **fausse**. Parfois, un choix doit être fait parmi plus de 2 possibilités. **if – else if – else** vous permet de vérifier plusieurs conditions et d'exécuter différentes instructions.

```
1
2     if( condition1 ){
3         // corps
4         //instructions à exécuter si la condition 1 est vraie
5     }
6     else if(condition2){
7         //corps
8         // instructions à exécuter si condition 2 est vraie
9         // condition 1 est false
10    }
11    else if(condition3){
12        //corps
13        // instructions à exécuter si condition 2 est vraie
14        // condition 1, condition 2 est false
15    }
16    ...
17    else{
18        //corps
19        // instructions à exécuter si toutes les conditions est fausse
20    }
```

Exemple 3 :

```
1     #include < stdio.h>
2     int main(void){
3         int a;
4
5         printf("Saisir une valeur : ");
6         scanf("%d",&a);
7
8         // structure conditionnelle
9         if(a< 0){
10            printf("la valeur de a est négative ");
11        }
12        else if( a > 0){
13            printf("la valeur de a est positive ");
14        }
15        else{
```

```
14              printf("la valeur de a est nulle ");
15          }
16
17          // fin du programme
18          printf("Good bye !");
19
20          return 0;
    }
```

```
Saisir une valeur : 5
la valeur de a est positive
Good bye !
Saisir une valeur : -3
la valeur de a est négative
Good bye !
Saisir une valeur : 0
la valeur de a est nulle
Good bye !
```

Structure conditionnelle if imbriquée

Il est possible d'inclure une instruction **if ... else** dans le corps d'une autre instruction **if ... else.**

Exemple 4 :

```
1
2       #include < stdio.h>
3       int main(void){
4           int a;
5
6           printf("Saisir une valeur : ");
7           scanf("%d",&a);
8
9           // structure conditionnelle
10          if(a <= 0){
11              if(a==0){
12                  printf("la valeur de a est nulle ");
13              }
14              else{
15                  printf("la valeur de a est négative ");
16              }
17
18          }
19          else{
20              printf("la valeur de a est positive ");
21          }
22
23          // fin du programme
24          printf("Good bye !");

            return 0;
    }
```

```
Saisir une valeur : 5
la valeur de a est positive
Good bye !
Saisir une valeur : -3
la valeur de a est négative
Good bye !
Saisir une valeur : 0
la valeur de a est nulle
Good bye !
```

Remarque ! Si le corps d'une instruction **if ... else** comporte une seule instruction, vous n'avez pas besoin d'utiliser des crochets **{}.**

Exemple 5 :

```
1    if(a <= 0){
2        printf("la valeur de a est négative ou nulle");
3    }
4    else{
5        printf("la valeur de a est positive ");
6    }
```

est équivalent à

Exemple 6 :

```
1    if(a <= 0)
2        printf("la valeur de a est négative ou nulle");
3    else
4        printf("la valeur de a est positive ");
```

INSTRUCTION SWITCH EN C

L'instruction **switch** est une instruction multidirectionnelle utilisée pour gérer les décisions. Cela fonctionne presque exactement comme la déclaration **if-else**. La différence est que l'instruction **switch** génère un code plus lisible par rapport à l'instruction **if-else**.

Syntaxe

```
1     switch(expression)
2     {
3         case val1:
4             instruction1;
5             instruction2;
6             ...
7             break ;
8         case val2:
9             instruction3;
10            instruction4;
```

```
            ...
            break ;
        case val3:
            instruction5;
            instruction6;
            ...
            break ;
        default:
            instruction7;
            ...
    }
```

"**expression**" dans l'instruction switch peut être toute expression valide qui donne une **valeur entière**. L'expression peut également être un caractère (car tous les caractères sont finalement convertis en un entier avant toute opération), mais il ne peut s'agir ni de virgule flottante (float, double) ni de chaîne de caractères.

val1, **val2** et ainsi de suite après le mot clé "**case**" doit être de type entier (comme int, long int, etc.) ou de type caractère. Elle peut aussi être une expression qui donne une valeur entière. Chaque cas doit avoir une seule valeur. Les valeurs multiples dans la déclaration **case** ne sont pas autorisées (contrairement à ce que vous avez en algorithmique), de plus, toutes les valeurs doivent être uniques.

Après chaque cas, nous pouvons avoir n'importe quel nombre d'instructions ou aucune. S'il existe plusieurs instructions, vous n'avez pas besoin de les inclure entre accolades ({}).

Voici quelques expressions de **switch** valides et des valeurs de cas.

Exemple 1

```
int addition(int x, int y){
    return x+y;
}
int a;
float b;
char c;

switch(a){ // valide

}
switch(c){ // valide

}

switch(b){ // invalide (valeur réelle)

}

switch(2+3){ // valide
```

```
}

switch(a+c){ // valide

}

switch(a+4){ // valide

}
switch("nom"){ // invalide (chaine de caractères)

}

switch(addition(a,2)){ // valide

}
```

Exemple 2

```
case 2:// valide
case 2+3:// valide
case 'a': // valide
case 'a' < 'c':// valide

case 1,2,3: // invalide (valeurs multiples)
case 3.5: // invalide (valeur réelle)
case a: // invalide (les variables ne sont pas autorisées)
```

Comment ça marche

Tout d'abord, l'expression de switch est évaluée, puis la valeur de cette expression est comparée à chaque cas, un par un. Si la valeur de l'expression correspond à une valeur de **case**, les instructions sous ce cas sont exécutées. Si on n'indique pas « break » toutes les instructions de tous les cas suivants seront exécutées. **Break** veut dire arrêter et sortir du bloc switch. Si la valeur de l'expression ne correspond à aucune valeur de case, les instructions de **default** sont exécutées.
L'instruction **default** est facultative si elle est omise et qu'aucun cas ne correspond, aucune action n'a lieu.

Exemple 3

voici un programme qui à partir d'un nombre compris entre 1 et 7 affiche le jour correspondant

```
#include< stdio.h>

```

```
int main(void){
    int jour;

    printf("saisir le numéro du jour : ");
    scanf("%d",&jour);

    switch(jour){
        case 1 : printf("Lundi"); break ;
        case 2 : printf("Mardi"); break ;
        case 3 : printf("Mercredi"); break ;
        case 4 : printf("Jeudi"); break ;
        case 5 : printf("Vendredi"); break ;
        case 6 : printf("Samedi"); break ;
        case 7 : printf("Dimanche"); break ;
        default: printf("jour invalide");
    }

    return 0;
}
```