




Leçon 2 : Les Structures conditionnelles





Table des matières

Objectifs	5
I - Pré requis	7
II - INTRODUCTION	9
III - I- LA STRUCTURE CONDITIONNELLE SIMPLE	11
IV - Exercice : EXERCICE N° 1	13
V - II- LA STRUCTURE CONDITIONNELLE ALTERNATIVE	15
VI - Exercice : EXERCICE N° 2	17
VII - III- LA STRUCTURE TERNAIRE	19
VIII - Exercice : EXERCICE N° 3	21
IX - IV- LES STRUCTURES CONDITIONNELLES IMBRIQUÉES	23
X - Exercice : EXERCICE N° 4	25
XI - V- LA STRUCTURE DE CHOIX	27
XII - Exercice : EXERCICE N° 5	29
Solution des exercices	31



Objectifs

Ce cours devra vous permettre de :

- **Utiliser** au mieux les structures conditionnelles en langage C
- **Distinguer** les différentes structures conditionnelles en langage C



Pré requis

I

Avant de suivre cette leçon vous devez :

- Avoir pour acquis la connaissance de l'algorithme ;
- Avoir survie la leçon 1 : Concepts généraux du Langage C





INTRODUCTION

II

Les structures conditionnelles font partie des structures de contrôle en programmation et en langage C en l'occurrence. Elles permettent de restreindre l'exécution d'un bloc d'instructions à partir d'une condition posée. Ainsi, il existe plusieurs structures conditionnelles en langage C que sont : la structure conditionnelle simple, la structure conditionnelle alternative, la structure ternaire, les structures imbriquées et la structure de choix.

I- LA STRUCTURE CONDITIONNELLE SIMPLE



Définition : 1- Définition

Une **structure conditionnelle simple** est une structure qui exécute un bloc d'instructions lorsque la condition posée est vérifiée.

2- Syntaxe

```
If(condition)
{
    Bloc d'instructions ;
}
```



Exemple : 3- Exemple

Énoncé : Écrire un programme qui vérifie si un nombre entré est égal à 2.

#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()

#include<conio.h> // elle permet de manipuler la console windows (par exemple effacer l'écran, changer la couleur de police de console etc.)

main()

{

int val1; //Déclaration de la variable val1

printf("Entrez un nombre entier svp!"); //affiche le message entre les doubles griffes

scanf("%d",&val1); //récupération de la valeur saisie par l'utilisateur

if(val1==2) //vérification si la valeur de la variable val1 est égale 2

{

printf("la valeur saisie est correcte"); //exécutée si condition est vraie

}

}



Exercice : EXERCICE N° 1

[Solution n°1 p 31]

EXO 1

1- Laquelle de ces syntaxes constitue la structure conditionnelle simple en langage C?

- | | |
|-----------------------|---|
| <input type="radio"/> | a. If(condition){ . . . } |
| <input type="radio"/> | b. If{ . . . } (condition) |
| | c. If(condition){ . . . } else{ . . . } |

EXO 2

2- Laquelle de ces syntaxes n'est pas correcte en langage C?

- | | |
|--|---|
| | c. If(condition){ . . . } else{ . . . } |
| | b. If{(condition) . . . } |
| | a. If(condition){ . . . } |

II- LA STRUCTURE CONDITIONNELLE ALTERNATIVE



Exemple : 1- Définition

La **structure conditionnelle alternative** est une structure qui exécute un bloc d'instructions lorsque la condition posée est vérifiée sinon elle exécute un autre bloc d'instructions dans le cas contraire.

2- Syntaxe

```
If(condition)
{
    Bloc d'instructions1 ;
}
else
{
    Bloc d'instructions2;
}
```

Exemple : 3- Exemple

Énoncé : Écrire un programme qui vérifie si un nombre entré est égal à 2 ou pas.

#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()

#include<conio.h> // elle permet de manipuler la console windows (par exemple effacer l'écran, changer la couleur de police de console etc.)

main()

{

Int val1; //Déclaration de la variable val1

printf("Entrez un nombre entier svp!"); // affiche le message entre les doubles griffes

scanf("%d",&val1); //récupération de la valeur saisie par l'utilisateur

if(val1==2) //vérification si la valeur de la variable val1 est égale 2

{

printf("la valeur saisie est correcte"); //exécutée si condition est vraie

}

else //vérification si la valeur de la variable val1 est différente de 2

{


```
printf("la saisie est incorrecte"); //exécutée dans le cas contraire (a valeur  
de la variable val1 est différente de 2) ;  
}  
}
```





Exercice : EXERCICE N° 2

[Solution n°2 p 31]

EXO 1

```
1- main()
{
  int v1, v2, v3;
  v1=2;
  v2=6;
  v3=5;
  if(v1 % 2 == 1 )
  {
    v3+=v2;
  }
  else
  {
    v3-=v2;
  }
}
```

Que valent v2 et v3 à la fin du programme?

a.v2 vaut 11 et v3 vaut 11
b. v2 vaut 6 et v3 vaut 11
c. v2 vaut 6 et v3 vaut -1

EXO 2

```
2- main()
{
  int v1,v2,v3;
  v1=1;
  v2=6;
  v3=5;
  if(v1==(v2-v3))
  {
    v1=6;
```

Exercice : EXERCICE N° 2

```
v2=--v1;  
}  
if((v2%2)==0)  
{  
v3=--v2;  
v1+=v3;  
}  
else  
{  
v3=v2++;  
v1-=v3;  
}  
}
```

Que valent v1, v2 et v3 à la fin du programme ?

	a. v1 vaut 0, v2 vaut 6 et v3 vaut 5
	b. v1 vaut 6, v2 vaut 5 et v3 vaut 5
	c. v1 vaut 12, v2 vaut 5 et v3 vaut 5



III- LA STRUCTURE TERNAIRE



Définition : 1- Définition

La **structure ternaire** est une forme de structure alternative qui se définit comme une instruction. Elle peut, par moment disposer de variable pour récupérer le contenu de l'instruction (par exemple lorsque l'instruction ne porte pas sur un affichage).

2- Syntaxe

(Condition) ? action_si_vrai : action_si_faux;

Action_si_vrai correspond à l'instruction à exécuter si la condition posée est vérifiée

Action_si_faux est l'instruction à réaliser dans le cas contraire

Exemple : 3- Exemple

Énoncé : Écrire un programme qui affiche la valeur absolue d'un nombre.

#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()

#include<conio.h> // elle permet de manipuler la console windows (par exemple effacer l'écran, changer la couleur de police de console etc.)

main()

{

int val1, val2; //Déclaration de variables val1 et val2

printf("Entrez un nombre entier svp!"); // affiche le message entre les doubles griffes

scanf("%d",&val1); //récupération de la valeur saisie par l'utilisateur

val2=val1; //affectation de la valeur de val1 à val2

val2=(val2>=0) ? val2 : -1 *val2; // si la valeur de val2 est supérieure ou égale à 0 alors val2 reçoit la valeur de val2 sinon val2 reçoit (-1*val2)

printf("la valeur absolue de %d est %d", val1, val2); // affichera la valeur absolue de la valeur de val1 est la valeur de val2.

}

Exercice :

EXERCICE N° 3

[Solution n°3 p 32]

EXO 1

```
main()
{
  int val1, val2;
  val1 = 2;
  val2=4;
  val2=(val2 - val1 >0)?val1*3:val2++;
  val1 =(val1 - val2>=0)?val1++ : val2-1;
}
```

Que valent $v1$, $v2$ et $v3$ à la fin du programme ?

a-	val1 vaut 6 et val2 vaut 5
b-	val1 vaut 5 et val2 vaut 6
c-	val1 vaut 6 et val2 vaut 4

EXO 2

```
int main()
{
  int val1, val2;
  val1 = 0;
  val2=1;
  val2=(val2 && val1)?val1*3:val2-1;
  val1 =(val1 || val2)?val1-1 : val2-1;
}
```

Que valent $v1$, $v2$ et $v3$ à la fin du programme ?

a-	val1 vaut 1 et val2 vaut 0
b-	val1 vaut 0 et val2 vaut 1
c-	val1 vaut -1 et val2 vaut 0

IV- LES STRUCTURES CONDITIONNELLES IMBRIQUÉES

Définition : 1- Définition

Les **structures conditionnelles imbriquées** sont généralement utilisées lorsque vous avez plus de deux conditions à vérifier. Les imbrications consistent à définir une série de structures conditionnelles.

2- Syntaxe

Il n'existe pas de syntaxe figée pour les structures conditionnelles imbriquées, cependant nous pouvons présenter quelques-unes.

```
If(condition)  
{  
  Bloc d'instructions1 ;  
}  
else if(condition2)  
{  
  Bloc d'instructions2;  
}  
else  
{  
  Bloc d'instructions n;  
}
```

Autre syntaxe :

```
If(condition)  
{  
  If(condition)  
  {  
    Bloc d'instructions;  
  }  
  else  
  {  
    Bloc d'instructions;
```

```

}
}

```

Exemple : 3- Exemple

Énoncé : Écrire un programme qui permet de dire si un nombre est pair ou impair. Si le nombre est pair il faut aussi spécifier s'il est positif ou pas.

#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()
#include<conio.h> // elle permet de manipuler la console windows (par exemple effacer l'écran, changer la couleur de police de console etc.)

main()

```
{
```

int nb; // déclaration de la variable nb

printf("saisissez un nombre svp!"); // affiche le message entre les griffes

scanf("%d",&nb); // récupère la valeur saisie dans la variable nb

if(nb % 2==0) // vérifie si le reste de la division de nb par 2 égal à 0 (% = modulo).

```
{
```

// la condition est vérifiée (nb % 2 ==0)

If(nb>=0) // vérifie encore si la valeur de nb est supérieure ou égale à 0

```
{
```

Printf("%d est un nombre pair positif", nb); // affiche la valeur de la variable nb est un nombre pair positif.

```
}
```

else

```
{
```

Printf("%d est un nombre pair négatif", nb); // affiche la valeur de la variable nb est un nombre pair négatif.

```
}
```

```
}
```

else

```
{
```

// la condition n'est pas vérifiée (nb % 2 <> 0)

Printf("%d est un nombre impair", nb); // affiche la valeur de la variable nb est un nombre impair.

```
}
```

```
}
```



Exercice : EXERCICE N° 4

[Solution n°4 p 32]

EXO 1

```
1- main()
{
int v1,v2,v3,v4,v5,v6;
v1=1;
v2=3;
v2*=3;
v3=10;
v4=10;
v5=v3--;
v6=--v4;
if(v6==v4)
{
if(v2!=v5)
{
printf("RSI");
}
else
{
printf("DAS")
}
}
if(v5==v6)
{
print("BD");
}
if(v2!=v4)
{
printf("MMX");
}
}
}

Que retourne ce programme ?
```


	a- MMX
	b- BD
	c- RSI
	d- DAS

EXO 2

```

4- main()
{
  int v1,v2,v3;
  v1=2;
  v2=4;
  v3=3;
  if(v1%2==1)
  {
    if((v2+v3)%2==0)
    {
      v3=++v1;
    }
    else
    {
      v3=v1++;
    }
  }
  else
  {
    if((v2+v1)%2==0)
    {
      v3-=v1;
    }
    else
    {
      v3+=v1;
    }
  }
}

```

Que valent v1, v2 et v3 à la fin du programme ?

	a. v1 vaut 4, v2 vaut 2 et v3 vaut 1
	b. v1 vaut 2, v2 vaut 4 et v3 vaut 1
	c. v1 vaut 1, v2 vaut 4 et v3 vaut 2

V- LA STRUCTURE DE CHOIX

Définition : 1- Définition

Elle permet d'exécuter une instruction lorsque l'une des conditions posées est vérifiée. Lorsqu'une des conditions posées est vérifiée avant que l'on atteigne la fin du programme, le mot réservé **break** permet de sortir de la structure après exécution de l'instruction. Si aucune des conditions n'est vérifiée, une instruction par défaut est exécutée grâce au mot réservé **default**.

2- Syntaxe

```
Switch(variable)
{
    case valeur1 : instruction1;
    break;
    case valeur2 : instruction 2;
    break;
    .
    .
    .
    default: instruction_par_défaut;
    break;
}
```

Exemple : 3- Exemple

Énoncé :

Écrire un programme qui dans un premier temps un menu de saisie pour afficher la salutation dans l'une des langues de choix.

- Il demande par la suite que l'utilisateur fasse son choix.
- Selon que la valeur saisie par ce dernier vaille 1, 2, 3 ou 4, une salutation est retournée.
- Si aucune de ces 4 valeurs n'est saisie, un message par défaut est retourné.

#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()

#include<conio.h> // elle permet de manipuler la console windows (par exemple effacer l'écran, changer la couleur de police de console etc.).

main()

{

int choix; // déclaration de la variable choix

```

printf("***** MENU SALUTATION *****\n\n"); // affichera ***** MENU
SALUTATION ***** et ira deux fois à la ligne à cause de deux \n
printf(" 1 - FRANCAIS \n 2 - ANGLAIS \n 3 - ESPAGNOL \n 4 -
ALLEMAND\n\n"); // affichera
1 – FRANÇAIS
2 – ANGLAIS
3 – ESPAGNOL
4 – ALLEMAND
A cause des \n et ira deux fois à la ligne à cause de deux \n après 4 – ALLEMAND
printf("Faites votre choix svp!\n"); // affichera Faites votre choix svp! Et ira à la
ligne à cause du \n
scanf("%d",&choix); //récupère la valeur saisie dans la variable choix
switch(choix) // selon la valeur de choix il affichera un message de salutation si la
valeur est comprise en 1 et 4 sinon il affichera La valeur saisie est incorrecte
{
case 1: printf("SALUT!");
break;
case 2: printf("HELLO!");
break;
case 3: printf("HOLA!");
break;
case 4: printf("TAG!");
break;
default: printf("La valeur saisie est incorrecte");
}
}
}
}

```



Exercice : EXERCICE N° 5

[Solution n°5 p 32]

EXO 1

```
10- main()
{
  int n1,n2;
  n1=3;
  switch(n1)
  {
    case 2: n2=10;
    break;
    case 3: n2=12;
    case 4: n2=14;
    default: n2=16;
    break;
  }
  Printf("%d",n2);
}
Que retourne n2 ?
```

a. 12

b. 14

c. 16



Solution des exercices

> Solution n°1 (exercice p. 13)

EXO 1

- | | |
|----------------------------------|---|
| <input checked="" type="radio"/> | a. If(condition){ . . . } |
| <input type="radio"/> | b. If{ . . . } (condition) |
| <input type="radio"/> | c. If(condition){ . . . } else{ . . . } |

EXO 2

- | | |
|----------------------------------|---|
| | c. If(condition){ . . . } else{ . . . } |
| <input checked="" type="radio"/> | b. If{(condition) . . . } |
| | a. If(condition){ . . . } |

> Solution n°2 (exercice p. 17)

EXO 1

- | | |
|--|----------------------------|
| | a.v2 vaut 11 et v3 vaut 11 |
| | b. v2 vaut 6 et v3 vaut 11 |
| | c. v2 vaut 6 et v3 vaut -1 |

EXO 2

a.	v1 vaut 0, v2 vaut 6 et v3 vaut 5
b.	v1 vaut 6, v2 vaut 5 et v3 vaut 5
c.	v1 vaut 12, v2 vaut 5 et v3 vaut 5

> Solution n°3 (exercice p. 21)

EXO 1

a-	val1 vaut 6 et val2 vaut 5
b-	val1 vaut 5 et val2 vaut 6
c-	val1 vaut 6 et val2 vaut 4

EXO 2

a-	val1 vaut 1 et val2 vaut 0
b-	val1 vaut 0 et val2 vaut 1
c-	val1 vaut -1 et val2 vaut 0

> Solution n°4 (exercice p. 25)

EXO 1

a-	MMX
b-	BD
c-	RSI
d-	DAS

EXO 2

a.	v1 vaut 4, v2 vaut 2 et v3 vaut 1
b.	v1 vaut 2, v2 vaut 4 et v3 vaut 1
c.	v1 vaut 1, v2 vaut 4 et v3 vaut 2

> Solution n°5 (exercice p. 29)

EXO 1

Solution des exercices

a. 12

b. 14

c. 16

