

Structures conditionnelles

Mohamed MESSABIHI

mohamed.messabihi@gmail.com

Université de Tlemcen
Département d'informatique

23 octobre 2014

Sans conditions, nos programmes informatiques feraient toujours la même chose !

On appelle structure conditionnelle les instructions qui permettent de tester si une condition est vraie ou non.



L'instruction « if... »

L'instruction **if** est la structure de test la plus basique, on la retrouve dans tous les langages (avec une syntaxe différente...). Elle permet d'exécuter une série d'instructions si jamais une condition est vérifiée.

Syntaxe :

```
if (condition)
{
    //liste d'instructions;
}
```

```
if (note >= 10)
{
    printf ("vous etes admis ! ");
}
```

L'instruction « if ... else »

L'expression **if ... else** permet d'exécuter une autre série d'instructions en cas de non-réalisation de la condition.

Syntaxe :

```
if (condition)
{
    //liste d'instructions
}

else
{
    //autre bloc d'instructions
}
```

Les opérateurs de comparaison

Ils servent à comparer deux expressions entre elles

Symbole	Signification
==	est égal à
>	est supérieur à
<	est inférieur à
>=	est supérieur ou égal à
<=	est inférieur ou égal à
!=	est différent de

Exemple :

```
int main () {
    int  valeur1;
    int  valeur2;
    printf ("Entrez une 1ere valeur : ");
    scanf ("%d",&valeur1);
    printf ("Entrez 2eme valeur : ");
    scanf ("%d",&valeur2);
    if (valeur1<valeur2)
        printf ("%d > %d\n",valeur2, valeur1);
    else
        printf ("%d >= %d\n",valeur1, valeur2);
    return 0;
}
```

Les opérateurs logiques

Les opérateurs logiques permettent de combiner des expressions logiques (conditions).

Symbole	Signification
&&	ET
	OU
!	NON

Exemple :

```
int i1=1;
int i2=0;
printf("i1 || i2 = %d", i1||i2);
printf("i1 && i2 = %d", i1&&i2);
printf("contraire(1)=%d", !(1));
}
```



Tests imbriqués

Exemple :

```
int main () {
    int a = 0;
    printf("Saisie de a : ");
    scanf("%d",&a);
    if (a < 0)                /* Strictement negative ? */
        printf("la variable a est negative.\n");
    else
    {
        if (a > 0)            /* Strictement positive ? */
            printf("la variable a est positive\n");
        else                  /* Sinon a est nulle */
            printf("la variable a est nulle\n");
    }
    return 0;
}
```

L'instruction « else if »

Exemple :

```
int main () {
    int a = 0;
    printf("Saisie de a : ");
    scanf("%d",&a);

    if (a < 0)                /* Strictement negative ? */
        printf("la variable a est negative.\n");
    else if (a > 0)           /* Strictement positive ? */
        printf("la variable a est positive\n");
    else                      /* Sinon a est nulle */
        printf("la variable a est nulle\n");

    return 0;
}
```


L'instruction « switch » 1/3

L'instruction **switch** permet de faire plusieurs tests de valeurs sur le contenu d'une même variable.

Syntaxe :

```
switch (Variable) {  
  case Valeur1 :  
    //Liste d'instructions;  
    break;  
  case Valeur2 :  
    //Liste d'instructions;  
    break;  
  case Valeurs... :  
    //Liste d'instructions;  
    break;  
  default:  
    //Liste d'instructions;  
}
```



L'instruction « switch » 2/3

Exemple :

```
int main()
{ int choixMenu;
  printf("==== Menu =====\n\n");
  printf("1. Royal Cheese \n");
  printf("2. Big Burger \n");
  printf("3. Complet Poulet \n");
  printf("4. Panini Thon \n");
  printf("\nVotre choix ? ");
  scanf("%d", &choixMenu); /* Saisie du choix de l'utilisateur */
  printf("\n");
  switch (choixMenu)      /* Tester le choix de l'utilisateur */
  {
    case 1: printf("Vous avez choisi un Royal Cheese !");
            break;
    case 2: printf("Vous avez choisi un Big Burger !");
            break;
    case 3: printf("Vous avez choisi un Complet Poulet !");
            break;
    case 4: printf("Vous avez choisi un Panini Thon !");
            break;
    default: printf("Choix incorrect. Vous ne mangerez rien !");
            break;
  } return 0;
}
```



L'instruction « switch » 3/3

- N'oubliez pas d'insérer des instructions break entre chaque test, ce genre d'oubli est difficile à détecter car aucune erreur n'est signalée...
- Ceci peut être utilisé judicieusement afin de faire exécuter les mêmes instructions pour différentes valeurs consécutives

Syntaxe :

```
switch(variable){  
  case 1:  
  case 2:  
    //instructions exécutées pour les valeur 1 et 2  
  break;  
  case 3:  
    //instructions exécutées pour la valeur 3  
  break;  
  default:  
    //instructions exécutées pour toute autre  
    valeur  
}
```



Une façon plus courte de faire un test

Il est possible de faire un test avec une structure beaucoup moins lourde grâce à la structure suivante :

`(condition) ? instruction si vrai : instruction si faux`

Remarques :

- la condition doit être entre des parenthèses
- Lorsque la condition est vraie, l'instruction de gauche est exécutée
- Lorsque la condition est fausse, l'instruction de droite est exécutée
- En plus d'être exécutée, la structure `? :` renvoie la valeur résultant de

Exemple :

```
(moyenne >= 10) ? printf ("Admis") : printf ("Ajourne");  
admis = ((moyenne >= 10) ? 1 : 0);
```

À retenir ...

- La valeur **Vrai** peut être assimilée à la valeur numérique **1** ou à toute valeur **non nulle**.
- La valeur **Faux** peut être assimilée à la valeur numérique **0**.
- Ne pas oublier les **parenthèses** lorsqu'il y a un **if**.

Exemple 1 :

```
if (1)
{
    printf("C'est vrai");
}
else
{
    printf("C'est faux");
}
```

Un exemple pour conclure ...

Exemple :

```
int main()
{
    int heure = 0;
    printf("Saisie de l'heure : ");
    scanf("%d",&heure);

    if (heure >= 6 && heure <12)
        printf(" C'est le matin \n");
    else if (heure >= 12 && heure < 17)
        printf(" C'est l'apres midi \n");
    else if (heure >= 17 && heure < 20)
        printf(" C'est le soir \n");
    else if ((heure >= 20 && heure <= 23) || (heure
        >=0 && heure <6) )
        printf(" C'est la nuit \n");
    return 0;
}
```

