

Travaux Pratiques N°3

Les structures conditionnelles

Objectif :

- Rappeler les structures conditionnelles
- Présenter les syntaxes offertes par le langage C permettant la manipulation des structures conditionnelles en C.

I. La structure alternative en C

I.1 La structure if else

Syntaxe

```

if ( expression ){
    ..... ;      /*bloc d'instructions*
}
else{
    ..... ;      /*bloc d'instructions*/
}

```

Dans le cas où aucun traitement n'est évoqué si l'expression logique est fausse, la structure conditionnelle devient :

```

if ( expression ){
    ..... ;      /*bloc d'instructions*
}

```

Remarques

- ✓ les {} ne sont pas nécessaires lorsque les blocs ne comportent qu'une seule instruction.
- ✓ Le langage C possède une paire d'opérateurs un peu exotiques qui peut être utilisée comme alternative à **if - else** et qui a l'avantage de pouvoir être intégrée dans une expression:

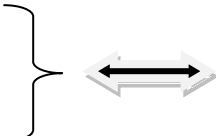
L'opérateur conditionnel (? :)

Syntaxe :

expr1 ? expr2 : expr3

Si **expr1** fournit une valeur non nulle, alors la valeur de **expr2** est fournie comme résultat si non la valeur de **expr3** est fournie comme résultat

Exemple :

<pre> if (A>B) MAX=A; else MAX=B; </pre>		<pre> MAX = (A > B) ? A : B; </pre>
---	---	--

I.2 La structure **Switch**: Instruction des élections multiple

Syntaxe :

```
switch(<var>){
    case Const1: instr1; break;
    case Const2: instr1; break;
    .....
    case Constn: instrn; break;
    default: instr1 ;
}
```

- <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 cecas.
- 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**).

Exemples:

On suppose que choix est une variable de type caractère, une instruction **switch** typique est la suivante:

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

On suppose que jour est une variable de type entier, une instruction **switch** typique est la suivante:

```
switch (jour)
{ case 0 : case 1: case 2: case 3 : case 4: printf("Atravail!"); break;
  case 5: printf("Aujourd'hui est samedi");
  case 6: printf("Reposez-vous"); }
```

Dans cette instruction **switch**, les valeurs 0 à 4 conduisent toutes à exécuter la même instruction **printf** puis l'instruction **break** fait sortir de **switch**. Mais si jour vaut 5 l'instruction **printf** est exécutée suivie de l'instruction **printf**("Reposez-vous") ; si jour vaut 6 seule la dernière instruction **printf** est exécutée.

II. Travail demandé

Exercice 1 :

Ecrire un programme qui lit trois valeurs entières (A, B et C) au clavier et qui affiche la plus grande des trois valeurs, en utilisant:

- a) **if - else** et une variable d'aide MAX
- b) **if - else if - ... - else** sans variable d'aide
- c) les opérateurs conditionnels et une variable d'aide MAX
- d) les opérateurs conditionnels sans variable d'aide

Exercice 2 :

Ecrivez un programme qui lit trois valeurs entières (A, B et C) au clavier. Triez les valeurs A, B et C par échanges successifs de manière à obtenir :

$$\text{val}(A) \leq \text{val}(B) \leq \text{val}(C)$$

Affichez les trois valeurs.

Exercice 3 :

Ecrivez un programme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe du produit de A et B sans faire la multiplication.

Exercice 4 :

Ecrivez un programme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de la somme de A et B sans faire l'addition. Utilisez la fonction **fabs** de la bibliothèque `<math.h>`.

Exercice 5 :

Ecrivez un programme qui calcule les solutions réelles d'une équation du second degré $ax^2+bx+c=0$ en discutant la formule:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Utilisez une variable d'aide **D** pour la valeur du discriminant b^2-4ac et décidez à l'aide de **D**, si l'équation a une, deux ou aucune solution réelle. Utilisez des variables du type **int** pour A, B et C.

Considérez aussi les cas où l'utilisateur entre des valeurs nulles pour A; pour A et B; pour A, B et C. Affichez les résultats et les messages nécessaires sur l'écran.