|  |  |  |
| --- | --- | --- |
|  | **INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL** | |
|  | Travaux Pratiques N° 2 : Les structures conditionnelles | |
| **Classes : TI 1** | | **Enseignante : Sana REFAI** |

## 1. Objectifs

Le but de ce TP est d'apprendre à utiliser les différentes structures conditionnelles dans un programme C.

## 2. Définition

Les structures de contrôle définissent la suite dans laquelle les instructions sont effectuées. Dans ce chapitre, nous allons voir comment les instructions de sélection connues fonctionnent en C et nous allons faire connaissance d'un couple d'opérateurs spécial qui nous permet de choisir entre deux valeurs à l'intérieur d'une expression.

Constatons déjà que la particularité la plus importante des instructions de contrôle en C est le fait que les 'conditions' en C peuvent être des *expressions quelconques qui fournissent un résultat numérique.* La valeur zéro correspond à la valeur logique faux et toute valeur différente de zéro est considérée comme vrai.

## 3. if- else

Il s'agit de l'instruction:

**Si (expression conditionnelle vraie) alors**

**<BLOC 1 D'INSTRUCTIONS>**

**Sinon**

**<BLOC 2 D'INSTRUCTIONS>**

**Finsi**



**if (<expression>)**

**<bloc 1 d'instructions>**

**else**

**<bloc 2 d'instructions>**

* **<expression>** :
* Une variable numérique
* Une expression fournissant un résultat numérique
* **<bloc d’instructions>** :
* Un ensemble d’instructions compris entre {}
* Une seule instruction
* **Valeur de l’<expression>** :
* Si l’<expression> fournit une valeur différente de 0 alors le <bloc d’instructions 1> est exécutée.
* Si l ’<expression> fournit la valeur 0 alors le <bloc d’instructions 2> est exécuté

***Remarque:***

* Le <bloc d’instructions> peut contenir d’autres instructions conditionnelles !
* Les {} ne sont pas nécessaires lorsque les blocs ne comportent qu'une seule instruction.

Exemple:

**if (A > B)**

**max = A;**

**else**

**max = B;**

***Attention !***

Comme la partie **else** est optionnelle, les expressions contenant plusieurs structures **if** et **if - else** peuvent mener à des confusions**.**

## 4. if sans else

Le bloc **"sinon"** est optionnel:

**si (expression vraie) Alors**

**<BLOC D'INSTRUCTIONS>**

****

**if (<expression>)**

**<bloc d'instructions>**

Exemple:

**if (A > 0)**

**Valeur\_Absolue = A;**

**else**

**Valeur\_Absolue = -A;**

Peut être remplacée par :

**Valeur\_Absolue = A;**

**if (A < 0)**

**Valeur\_Absolue = -A;**

## 5. if- else if- …- else

Autre utilisation de l’instruction **if** :

**if (<expr1>)**

Syntaxe:

**<bloc1>**

**else if (<expr2>)**

**<bloc2>**

**...**

**else if (<exprN>)**

**<blocN>**

**else**

**<blocN+1>**

* Les expressions <expr1>… <exprN> sont évaluées de haut vers le bas jusqu’à ce que l’une d’elles soit vrai.
* Si une expression est vérifiée, le bloc y lié est exécuté et le traitement s’arrête.
* La dernière **else** traite le cas où aucune des conditions n’a été remplie (elle est optionnelle).

**if (note >= 10)**

Exemple:

**printf (" Succès  \n");**

**else if (note >= 9)**

**printf (" Rachat \n");**

**else if (note >= 8)**

**printf (" Contrôle \n");**

**else**

**printf (” Echec \n” );**

## 6. La structure alternative switch

L'instruction **switch** permet des choix multiples uniquement sur des entiers (int) ou des caractères (char).

Syntaxe:

**switch (<variable>)**

**{**

**case <valeur1> : <bloc1>**

**break;**

**...**

**case <valeurN> : <blocN>**

**break;**

**default : <blocN+1>**

**}**

* On teste si <variable> est égale à <valeur1> ... <valeurN> de haut vers le bas
* Si <variable> est égale à une valeur <valeuri>, le bloc y lié est exécuté et le traitement s’arrête (**break**).
* **Default** traite le cas où aucune des conditions n ’a été remplie (elle est optionnelle et pas de **break**).

**REMARQUE** **:**

* L’instruction **switch** correspond à une cascade d’instructions **if ...else**

Exemple1:

**switch (trouve)**

**{ case 0 : printf (" Pas de solution\n");**

**break;**

**case 1: printf (" j’ai trouvé une solution \n");**

**break;**

**case 2: printf (" j’ai trouvé deux solutions \n");**

**break;**

**case -1: printf (" Impossible de résoudre \n");**

**break;**

**}**

Exemple2:

**switch (operateur)**

**{**  **case ‘+’ :** **printf (" c’est l’addition ");**

**break;**

**case ‘-’ : printf (" c’est la soustraction ");**

**break;**

**case ‘\*’ : printf (" c’est la multiplication ");**

**break;**

**case ‘/’ : printf (" c’est la division ");**

**break;**

**default : printf (" opérateur non connu ");**

## 7. Les opérateurs conditionnels

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 :

**Les opérateurs conditionnels:**

**<expr1> ? <expr2> : <expr3>**

* Si <expr1> fournit une valeur différente de zéro, alors la valeur de <expr2> est fournie comme résultat.
* Si <expr1> fournit la valeur zéro, alors la valeur de <expr3> est fournie comme résultat.

La suite d’instructions :

**if (A>B)**

Exemple:

**MAX=A;**

**else**

**MAX=B;**

peut être remplacée par:

**MAX = (A > B) ? A : B;**

Employés de façon irréfléchis, les opérateurs conditionnels peuvent nuire à la lisibilité d'un programme, mais si on les utilise avec précaution, ils fournissent des solutions très élégantes :

**printf("Vous avez %i carte%c \n", N, (N==1) ? ' ' : 's');**

Les ***règles de conversion de types*** s'appliquent aussi aux opérateurs conditionnels **? :** Ainsi, pour un entier N du type **int** et un rationnel F du type **float**, l'expression

**(N>0) ? N : F**

va ***toujours*** fournir un résultat du type **float**, n'importe si N est plus grand ou plus petit que zéro

## 8. Les Exercices

### 8.1 Exercice 1

Ecrire un programme C qui permet de saisir et de comparer deux entiers naturels a et b et affiche sur l’écran le résultat suivant : a égale b si a=b ; a est inférieur à b si a < **b ;**  a est supérieur à b si a > **b ;**

### 8.2 Exercice 2

Ecrire un programme qui lit un caractère au clavier puis affiche s'il s'agit d'une lettre minuscule, d'une lettre majuscule, d'un chiffre ou d'un caractère spécial.

### 8.3 Exercice 3

Une année est bissextile (contient 366 jours) si :

* si l'année est divisible par 4 et non divisible par 100 ;
* si l'année est divisible par 400

Exemples

• 1980 et 1996 sont bissextiles car elles sont divisibles par 4

• 2000 est une année bissextile car elle est divisible par 400

• 2100 et 3000 ne sont pas bissextiles car elles ne sont pas divisibles par 400. Ecrire un programme C qui permet de déterminer si un entier positif donné correspond à une année bissextile ou non.

### 8.4 Exercice 4

Ecrire un programme C qui lit au clavier 2 entiers et un opérateur ( +, -, /,\*) et simule le fonctionnement d’une calculatrice en affichant le résultat.

### 8.5 Exercice 5

Ecrire un programme C qui lit au clavier un entier correspondant à la position d’une couleur dans l’arc-en-ciel et affiche le nom correspondant à l’écran.

* Violet (1)
* Indigo (2)
* Bleu (3)
* Vert (4)
* Jaune (5)
* Orangé (6)
* Rouge (7)

### 8.6 Exercice 6

Exécuter le code suivant pour an = 1986 , puis 1988, 1900 et 2000 :

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int main()  {  int bissextile, an;  printf("Entrer une année \n");  scanf("%d",&an);  bissextile=(an % 4 == 0 && !(an % 100 == 0)) || (an % 400 == 0);  printf("la valeur de bissextil est %d\n",bissextile);  return 0;  } |

Que retourne ce code comme résultat ? Que permet-il de faire ?

### 8.7 Exercice7

|  |
| --- |
| #include <stdio.h>  int main()  {  int mm,an;  int rs;  printf("Entrer une anneée \n");  scanf("%d",&an );  printf("Entrer un mois \n");  scanf("%d",&mm);  if (mm == 2)  { /\* à complètera utiliser le code de ‘l’exercice 1.3.1 \*/ }  else if (mm <= 7)  { rs = (mm % 2 == 1 ? 31 : 30); }  else  { rs = (mm % 2 == 0 ? 31 : 30); }  printf("le dernier jour du mois est %d",rs);  return 0;  } |

Compléter e code suivant afin qu’il calcule le dernier jour du mois

### 8.8 Exercice8

* Ecrire un programme C qui saisit 3 entiers : jour, mois et année et qui calcule la date du lendemain.

Vous devez utiliser le code de l’exercice 2.3.1 et 2.3.