Structured Programming Language Sessional

CSE 1104

Sessional - 03

Decision Making in C

Department of Computer Science and Engineering

Pabna University of Science and Technology

1 Introduction

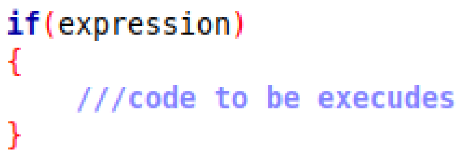
Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

C programming language provides the following types of decision making statements:

- If statement
- Switch statement
- Conditional Operator

1.1 If statement

The if statement is used to check some given condition and perform some operations depending upon the correctness of that condition. It is mostly used in the scenario where we need to perform the different operations for the different conditions. The syntax of the if statement is given below.

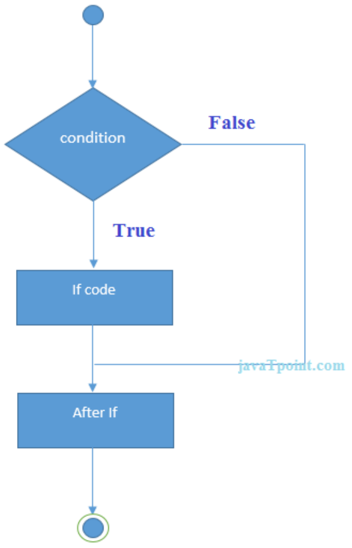


```
if(expression)
{
    ///code to be execudes
}
```

There are the following variants of if statement in C language.

- If statement
- If-else statement
- If else-if ladder
- Nested if

Flowchart of if statement:



Example 1: Write C Program to check If statement.

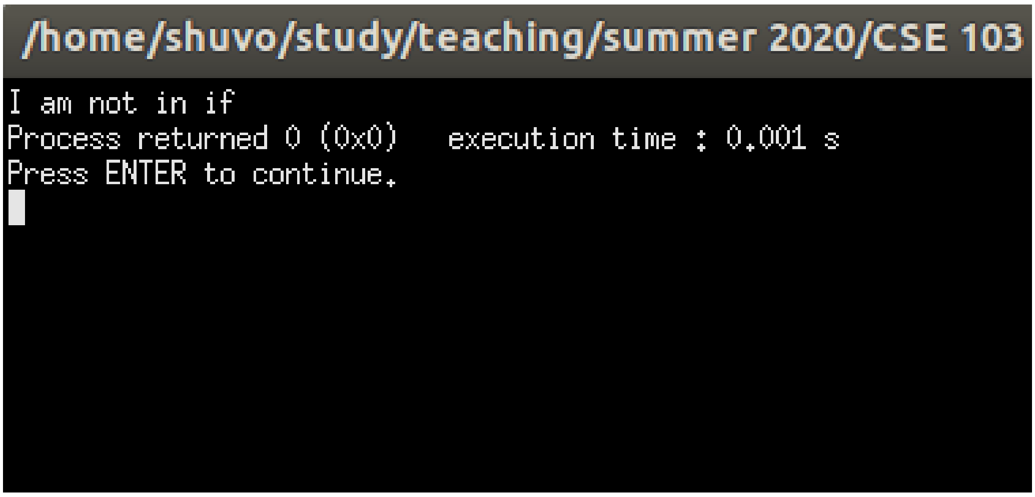
Algorithm 1 Steps in pseudo code:

```
Step1: Start
step2: assign value in variable i
step3: The if statement evaluates the i expression inside the parenthesis ().
step4: If the i expression is evaluated to true, statements inside the body of if are executed. or, If the i expression is evaluated to false, statements inside the body of if are not executed.
step5: The statements after the body of if are executed.
step6: end
```

Code:

```c
#include<stdio.h>

int main()
{
    int i=10;
    if(i>20)
    {
      printf("10 is less than 20");
    }
    printf("I am not in if");
}
```

Output:



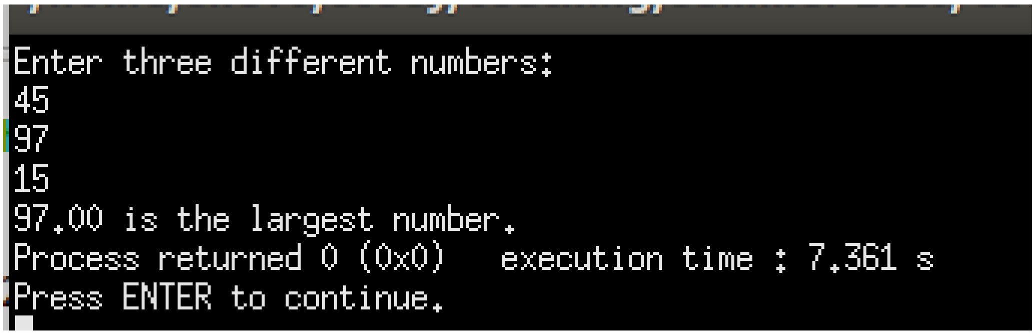
```
I am not in if
Process returned 0 (0x0)   execution time : 0.001 s
Press ENTER to continue.
```

Example 2: Write a C program to check the largest number(take three number from user).

Code:

```c
#include <stdio.h>
int main() {
    double n1, n2, n3;
    printf("Enter three different numbers: ");
    scanf("%lf %lf %lf", &n1, &n2, &n3);

    // if n1 is greater than both n2 and n3, n1 is the largest
    if (n1 >= n2 && n1 >= n3)
        printf("%.2f is the largest number.", n1);

    // if n2 is greater than both n1 and n3, n2 is the largest
    if (n2 >= n1 && n2 >= n3)
        printf("%.2f is the largest number.", n2);

    // if n3 is greater than both n1 and n2, n3 is the largest
    if (n3 >= n1 && n3 >= n2)
        printf("%.2f is the largest number.", n3);

    return 0;
}
```

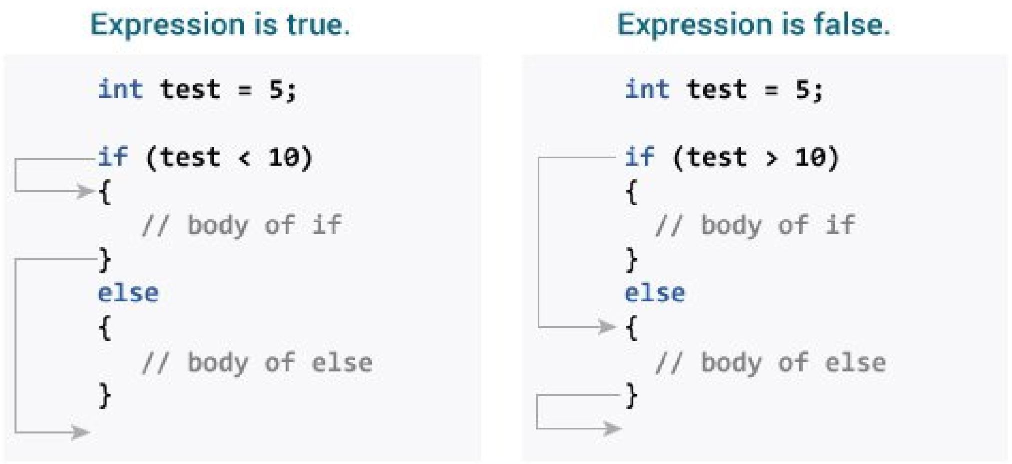
Output:



```
Enter three different numbers:
45
97
15
97.00 is the largest number.
Process returned 0 (0x0)   execution time : 7.361 s
Press ENTER to continue.
```

1.1.1 If-else Statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.



Expression is true.

```
int test = 5;

if (test < 10)
{
   // body of if
}
else
{
   // body of else
}
```

Expression is false.

```
int test = 5;

if (test > 10)
{
   // body of if
}
else
{
   // body of else
}
```

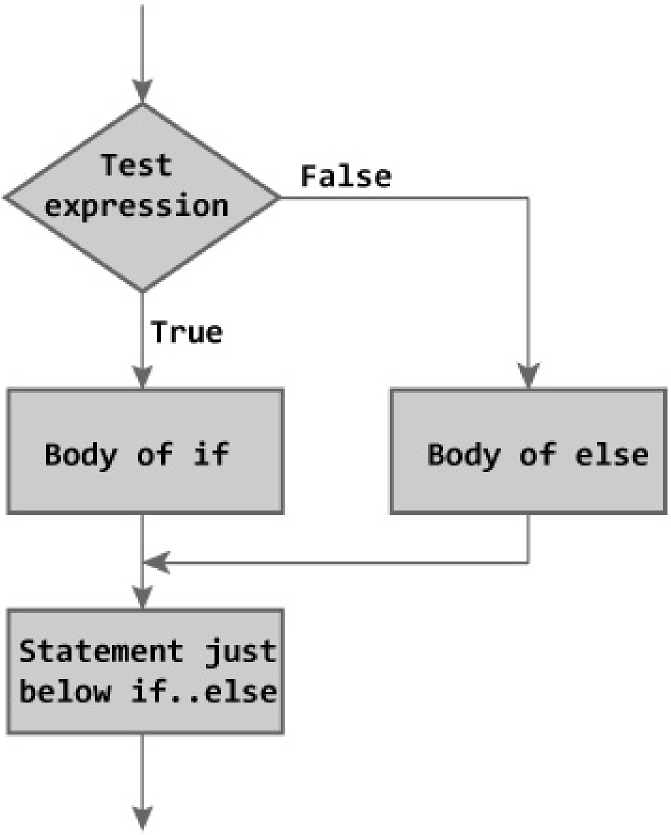


Figure: Flowchart of if...else Statement

Example 1: Write a C program to Check whether an integer is odd or even(take input from user).

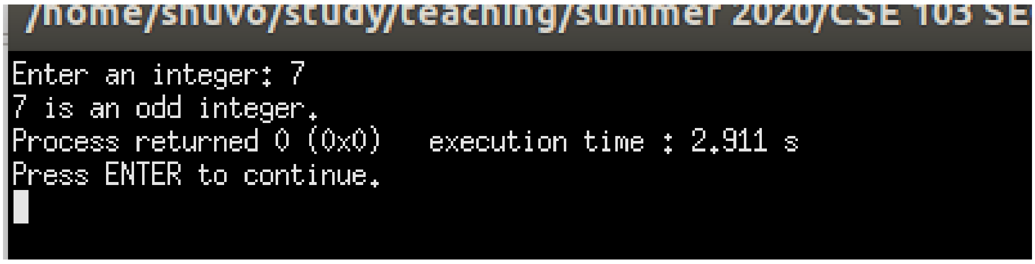
Algorithm 2 Steps in pseudo code:

```
Step1: Start
Step2: Declare variable number
Step3: read number
Step4: If the remainder of that number is 0, it is evaluated to true. The statements inside the body of if are executed. & The statements inside the body of else are skipped from execution.
Step5: If the remainder of that number is not 0, it is evaluated to false. The statements inside the body of else are executed. & The statements inside the body of if are skipped from execution.
Step6: End
```

Code:

```c
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);

    // true if num is perfectly divisible by 2
    if(num % 2 == 0)
        printf("%d is even.", num);
    else
        printf("%d is odd.", num);

    return 0;
}
```

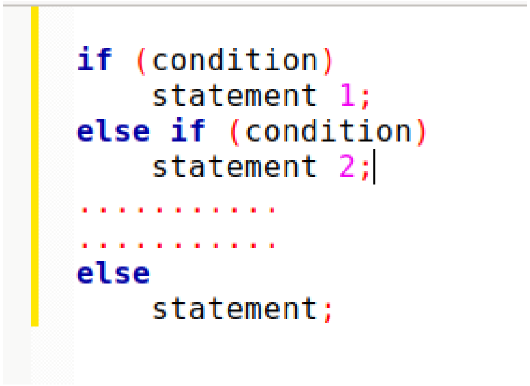
Output:



```
/home/shuvo/study/teaching/summer 2020/CSE 103 SE
Enter an integer: 7
7 is an odd integer.
Process returned 0 (0x0)   execution time : 2.911 s
Press ENTER to continue.
```

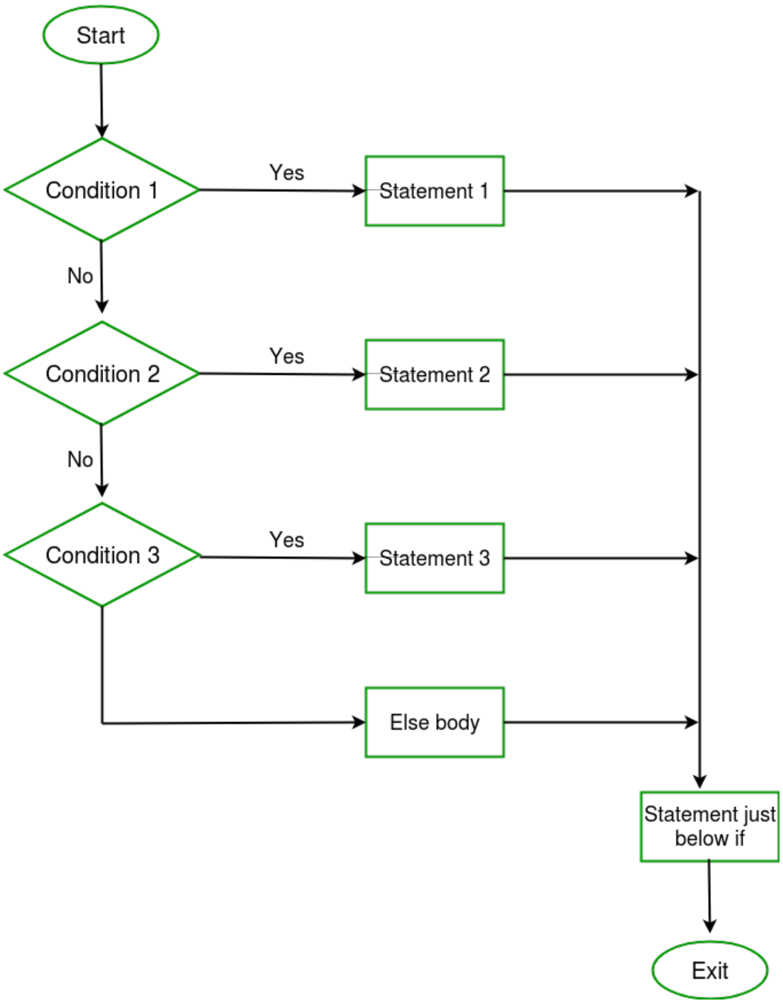
1.1.2 If else-if ladder

In C if-else-if ladder helps user decide from among multiple options. The C if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. The syntax of the else-if ladder statement is given below.



```
if (condition)
    statement 1;
else if (condition)
    statement 2;
...........
...........
else
    statement;
```

Flowchart of if-else statement:

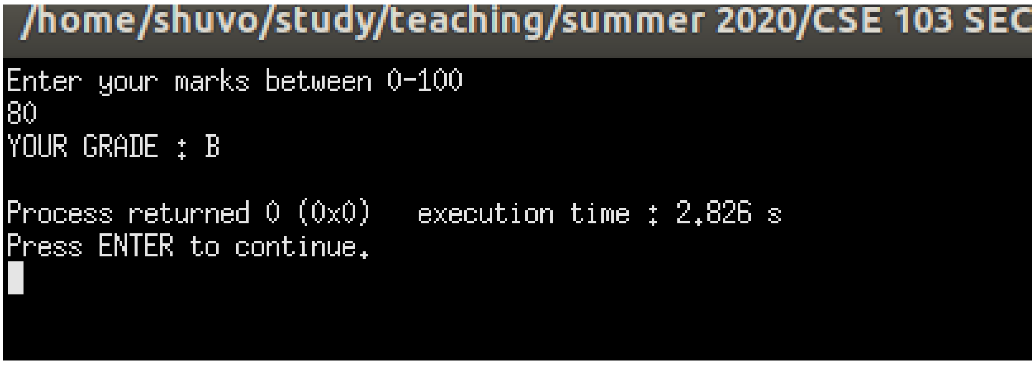


Example 1: Write a C Program to print grade of a student using If Else Ladder Statement.(take input from user).

Code:

```c
#include<stdio.h>
int main(){
    int marks;

    printf("Enter your marks between 0-100\n");
    scanf("%d", &marks);
    /* Using if else ladder statement to print
       Grade of a Student */
    if(marks >= 90){
        /* Marks between 90-100 */
        printf("YOUR GRADE : A\n");
    } else if (marks >= 70 && marks < 90){
        /* Marks between 70-89 */
        printf("YOUR GRADE : B\n");
    } else if (marks >= 50 && marks < 70){
        /* Marks between 50-69 */
        printf("YOUR GRADE : C\n");
    } else {
        /* Marks less than 50 */
        printf("YOUR GRADE : Failed\n");
    }
    return(0);
}
```

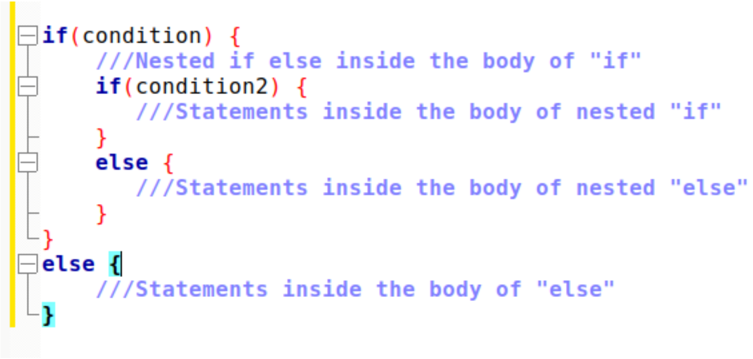
Output:



```
Enter your marks between 0-100
80
YOUR GRADE : B

Process returned 0 (0x0)   execution time : 2.826 s
Press ENTER to continue.
```

1.1.3 Nested if

In C programming, when an if else statement is present inside the body of another "if" or "else" then this is called nested if else.The syntax of the nested if statement is given below



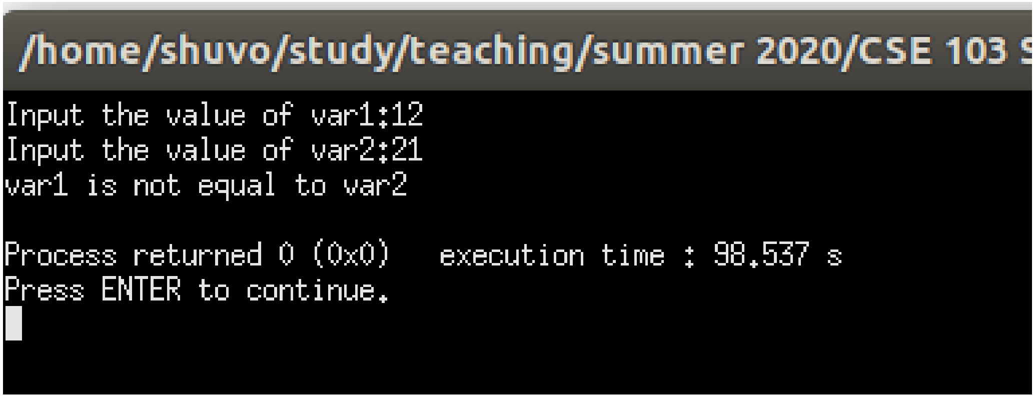
```
if(condition) {
    ///Nested if else inside the body of "if"
    if(condition2) {
        ///Statements inside the body of nested "if"
    }
    else {
        ///Statements inside the body of nested "else"
    }
}
else {
    ///Statements inside the body of "else"
}
```

Example 1: C program to illustrate the concept of nested if-else.

Code:

```
#include <stdio.h>
int main()
{
   int var1, var2;
   printf("Input the value of var1:");
   scanf("%d", &var1);
   printf("Input the value of var2:");
   scanf("%d",&var2);
   if (var1 != var2)
   {
  printf("var1 is not equal to var2\n");
  if (var1 > var2)
  {
    printf("var1 is greater than var2\n");
  }
  else
  {
    printf("var2 is greater than var1\n");
  }
   }
   else
   {
  printf("var1 is equal to var2\n");
   }
   return 0;
}
```

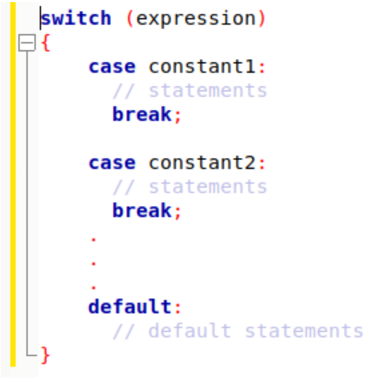
Output:



```
Input the value of var1:12
Input the value of var2:21
var1 is not equal to var2

Process returned 0 (0x0)   execution time : 98.537 s
Press ENTER to continue.
```

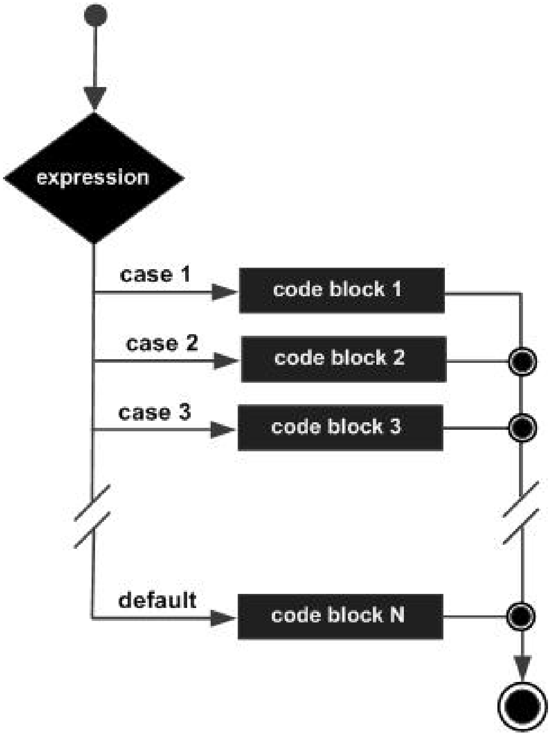
1.2 Switch Statement

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case. The syntax for switch case statement is given below.



```
switch (expression)
{
    case constant1:
      // statements
      break;

    case constant2:
      // statements
      break;
    .
    .
    .
    default:
      // default statements
}
```

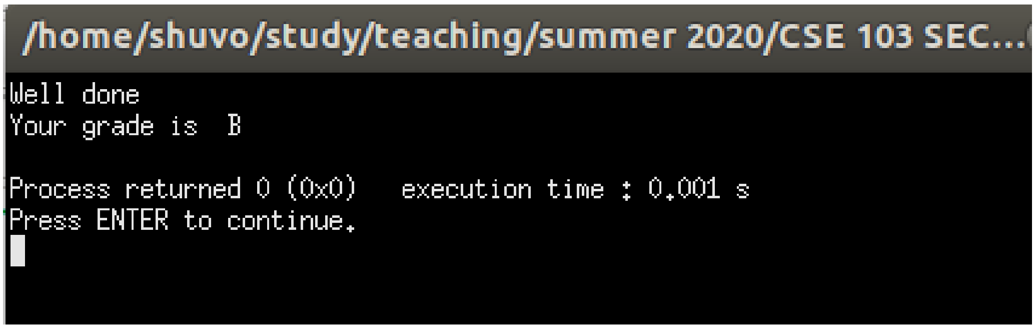
Flowchart of switch-case statement:



Example 1: Write a C Program to illustrate the concept of switch case in C. Code:

```c
#include <stdio.h>

int main () {

   /* local variable definition */
   char grade = 'B';

   switch(grade) {
      case 'A' :
         printf("Excellent!\n" );
         break;
      case 'B' :
      case 'C' :
         printf("Well done\n" );
         break;
      case 'D' :
         printf("You passed\n" );
         break;
      case 'F' :
         printf("Better try again\n" );
         break;
      default :
         printf("Invalid grade\n" );
   }

   printf("Your grade is  %c\n", grade );

   return 0;
}
```

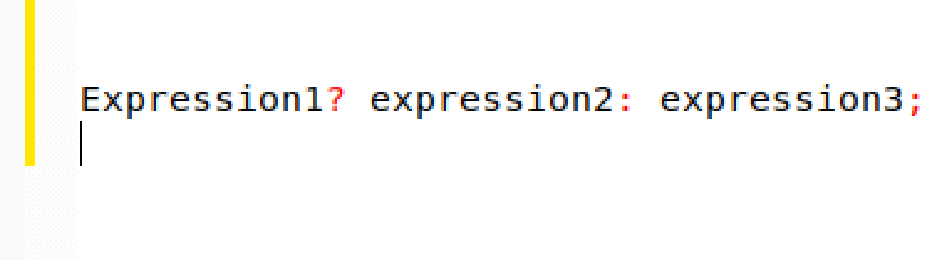
Output:



```
Well done
Your grade is  B

Process returned 0 (0x0)   execution time : 0.001 s
Press ENTER to continue.
```

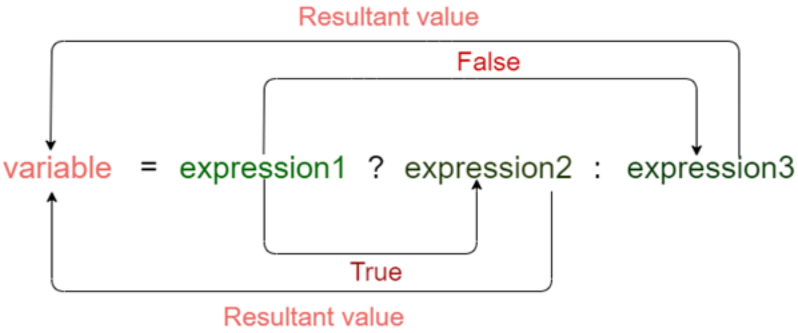
1.3 Conditional Operator Statement

The conditional operator is also known as a ternary operator. The conditional statements are the decision-making statements which depends upon the output of the expression. As conditional operator works on three operands, so it is also known as the ternary operator. The syntax for conditional operator is given below.



```
Expression1? expression2: expression3;
```

Flowchart of conditional statement:

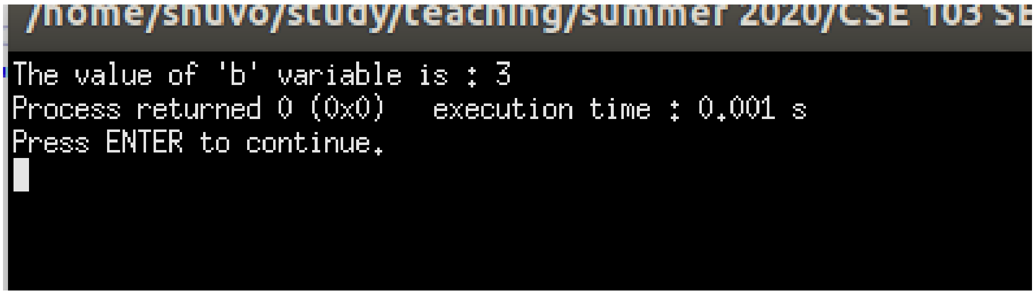


Example 1: C Program to illustrate the concept of conditional operator in C.

Code:

```c
#include <stdio.h>
int main()
{
   int a=5,b;  // variable declaration
   b=((a==5)?(3):(2)); // conditional operator
   printf("The value of 'b' variable is : %d",b);
    return 0;
}
```

Output:



```
/home/shuvo/study/teaching/summer 2020/CSE 103 SE
The value of 'b' variable is : 3
Process returned 0 (0x0)   execution time : 0.001 s
Press ENTER to continue.
```

2 Discussion & Conclusion

Based on the focused objective(s) to understand about the stack operations, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

3 Lab Task (Please implement yourself and show the output to the instructor)

1. Write a C program to find maximum between two numbers.

2. Write a C program to check whether a number is negative, positive or zero.

3. Write a C program to check whether a given number is odd or even.

4 Lab Exercise (Submit as a report)

- Write a C program to check whether a number is divisible by 5 and 11 or not.
- Write a C program to find maximum between three numbers.
- Write a Program to take the value from the user as input any alphabet and check whether it is vowel or consonant (Using the switch statement).
- Write a C program to check whether a year is leap year or not.

5 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.