

| **TITLE:** Write a program to accept 3 numbers from the user and find the largest of the 3 numbers using                    If - else if-else                    Ternary operator |
| --- |

**AIM:** Write a program to accept 3 numbers from the user and find the largest of the 3 numbers using

                  If - else if-else

                  Ternary operator

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

| CO2: Apply basic concepts of C programming for problem solving |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

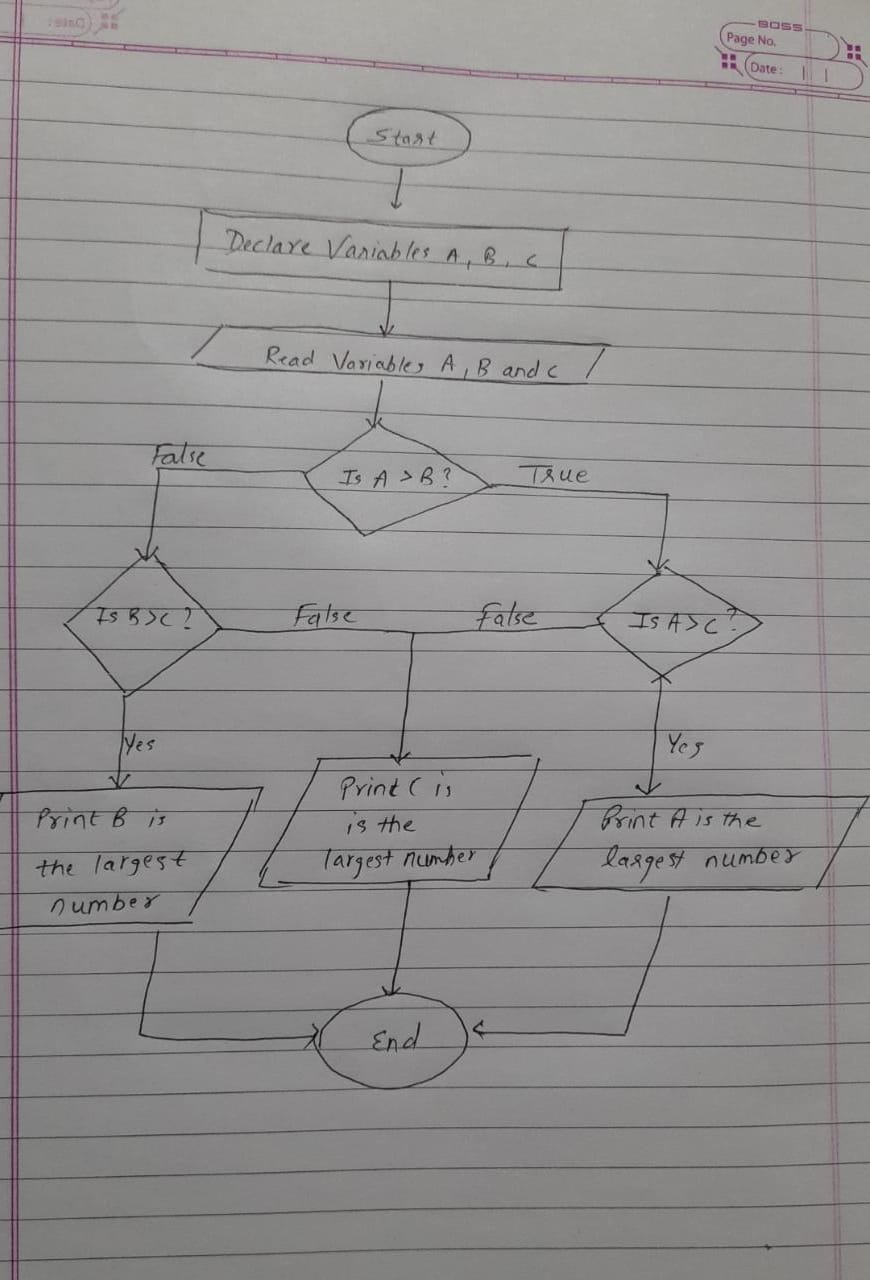
1. Programming in ANSI C, E. Balagurusamy, 7 th Edition, 2016, McGraw-Hill Education, India.
2. Structured Programming Approach, Pradeep Dey and Manas Ghosh, 1 st Edition, 2016, Oxford University Press, India.
3. Let Us C, Yashwant Kanetkar, 15th Edition, 2016, BPB Publications, India.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

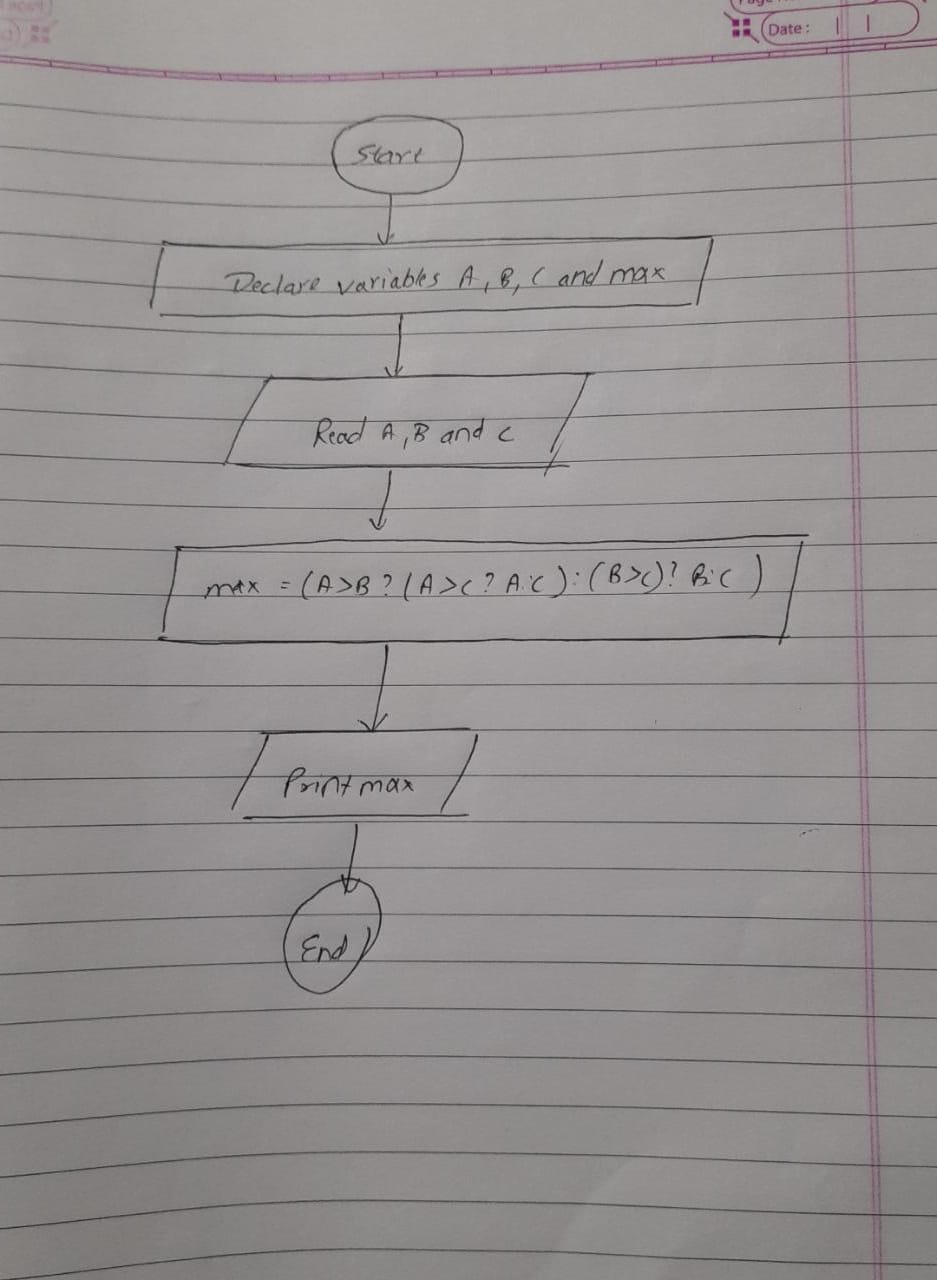
**Problem Definition:**

Ask user to input three numbers. Compare three numbers to find the largest of them using

1. Nested if else statement
2. Using ternary operator

**1.nested if else statement**

**2.**Using ternary operator



**Implementation details:**

1.//nested if else statement Meet Gala 16010121051 A3

#include <stdio.h>

int main() {

int n1, n2, n3;

printf("Enter three numbers: ");

scanf("%d %d %d", &n1, &n2, &n3);

if (n1 >= n2) {

if (n1 >= n3)

printf("%d is the largest number.", n1);

else

printf("%dis the largest number.", n3);

} else {

if (n2 >= n3)

printf("%d is the largest number.", n2);

else

printf("%d is the largest number.", n3);

}

2.

//ternary operator Meet Gala 16010121051 A3

#include<stdio.h>

int main()

{

int a,b,c,max;

printf("Enter three numbers\n");

scanf("%d %d %d" ,&a, &b, &c);

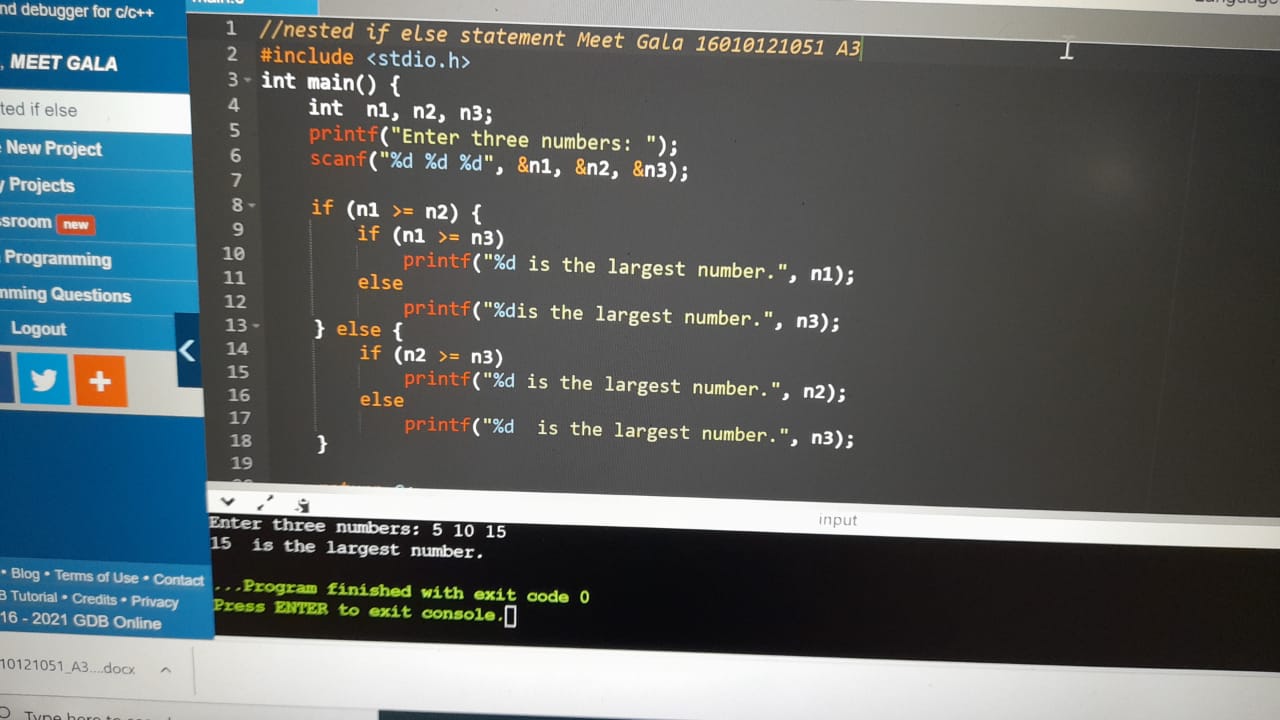
max= a>b ? (a>c?a:c):(b>c?b:c);

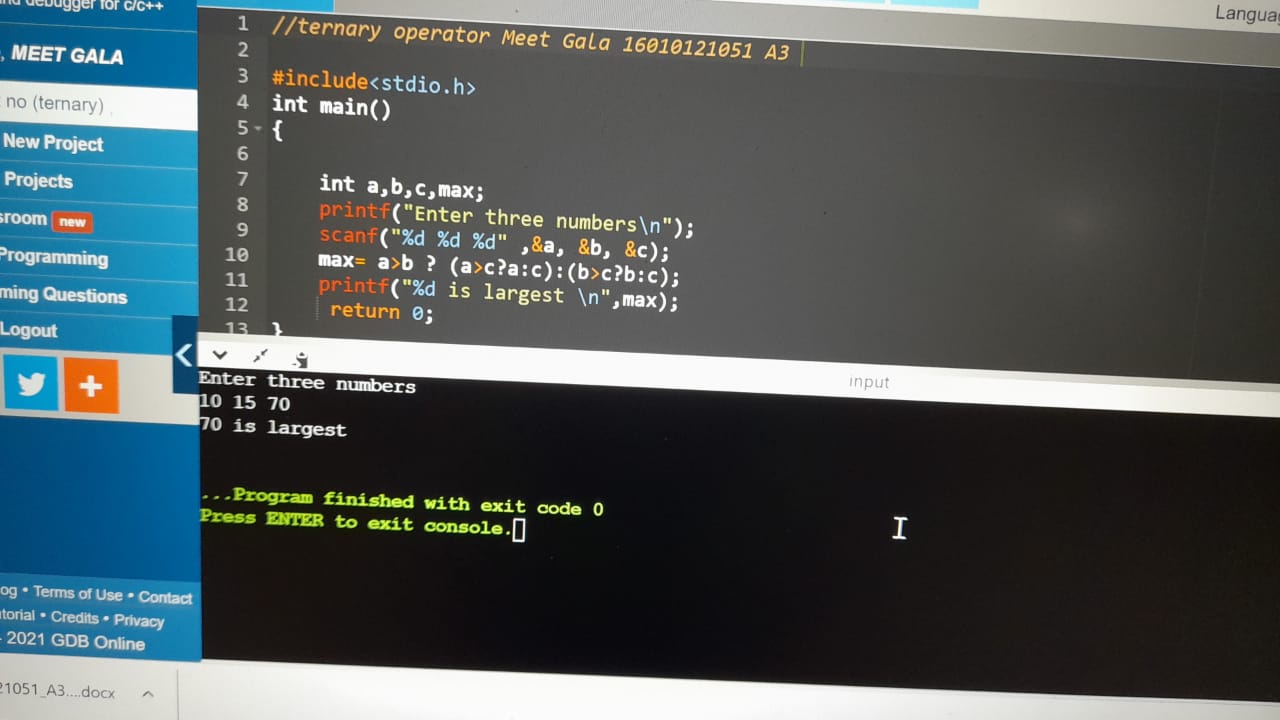
printf("%d is largest \n",max);

return 0;

}

**Output(s):**

1.

2.

**Conclusion:**

**Program 1 built using “If - else If – else” shows the largest number among three input numbers without any error.**

**Program 2 built using “Ternary Operator” shows the largest number among three input numbers without any error.**

**Post Lab Descriptive Questions**

**1. Explain bitwise operators with examples**

**Bitwise Operators are used for manipulating data at the bit level, also called bit level programming. Bitwise operates on one or more-bit patterns or binary numerals at the level of their individual bits. It consists of two digits, either 0 or 1. They are used in numerical computations to make the calculation process faster.**

**Operator Meaning**

**& Bitwise AND operator**

**| Bitwise OR operator**

**^ Bitwise exclusive OR operator**

**~ One’s Complement Operator**

**<< Left shift operator**

**>> Right shift operator**

**Bitwise AND Operator:**

**It is denoted by the single ampersand sign (&). Two integer operands are written on both sides of the (&) operator. If the corresponding bits of both the operands are 1, then the output of the bitwise AND operation is 1, otherwise the output would be 0.**

**Example:**

**Let the two variable A and B be:**

**A= 5**

**B=9**

**The binary representation of A and B will be:**

**A= 0101**

**B= 1001**

**On applying Bitwise AND Operator i.e. A&B the output will be:**

**Result= 0001**

**Bitwise OR operator:**

**It is represented by a single vertical sign (|). Two integer operands are written on both sides of the (|) symbol. If the bit value of any of the operand is 1, then the output would be 1, otherwise 0.**

**Example:**

**Let the two variable A and B be:**

**A= 5**

**B=9**

**The binary representation of A and B will be:**

**A= 0101**

**B= 1001**

**On applying Bitwise OR Operator i.e. A|B the output will be:**

**Result: 1101**

**Bitwise exclusive OR operator:**

**It is denoted by (^) symbol. Two operands are written on both sides of the exclusive OR operator. If the corresponding bit of any of the operand is 1 then the output would be 1, otherwise 0.**

**Example:**

**Let the two variable A and B be:**

**A= 5**

**B=9**

**The binary representation of A and B will be:**

**A= 0101**

**B= 1001**

**On applying Bitwise EXCLUSIVE OR Operator i.e. A^B the output will be:**

**Result: 1100**

**One’s Complement Operator:**

**The bitwise complement operator is also known as one's complement operator. It is represented by the symbol tilde (~). It takes only one operand or variable and performs complement operation on an operand. When we apply the complement operation on any bits, then 0 becomes 1 and 1 becomes 0.**

**Example:**

**Let A=5**

**The binary representation of A will be A=0101**

**On applying One’s Complement Operator i.e., ~A the output will be:**

**Result= 1010**

**Left shift operator:**

**It is an operator that shifts the number of bits to the left-side. In the case of Left-shift operator, 'n' bits will be shifted on the left-side. The 'n' bits on the left side will be popped out, and 'n' bits on the right-side are filled with 0.**

**Example:**

**Let variable A be:**

**A=5**

**The binary representation of A will be A= 00000101**

**If we want to left-shift the above representation by 2, then the statement would be:**

**A<<2**

**Result: A<<2 = 00010100**

**Right shift operator:**

**It is an operator that shifts the number of bits to the right side. In the case of the right-shift operator, 'n' bits will be shifted on the right-side. The 'n' bits on the right-side will be popped out, and 'n' bits on the left-side are filled with 0.**

**Example:**

**Let variable B be:**

**B=9**

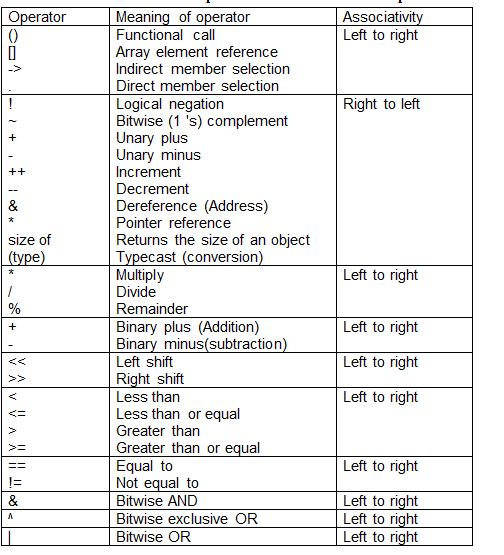
**The binary representation of B will be B= 00001001**

**If we want to right-shift the above representation by 2, then the statement would be:**

**B>>2**

**Result: B>>2=00100100**

**2. Write associative rules and precedence table of various operators.**

****

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**