**Operator:**

An operator is a symbol that tells the compiler to perform a certain mathematical or logical manipulation. Operators are used in programs to manipulate data and variables.

**Classification if operator on the basis of functionality:**

1. **Arithmetic operator**

The Arithmetic operators are some of the C Programming Operator, which are used to perform arithmetic operations includes operators like Addition, Subtraction, Multiplication, Division and Modulus.

1. **Relational operator**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), a relational operator is a [programming language](https://en.wikipedia.org/wiki/Programming_language) construct or [operator](https://en.wikipedia.org/wiki/Operator_(programming)) that tests or defines some kind of [relation](https://en.wikipedia.org/wiki/Relation_(mathematics)) between [two entities](https://en.wikipedia.org/wiki/Binary_function).

1. **Logical operator**

These operators are used to perform logical operations on the given expressions. There are 3 logical operators in C language. They are, logical AND (&&), logical OR (||) and logical NOT (!).

1. **Assignment operator**

Assignment operators are used to assigning value to a variable. The left side operand of the assignment operator is a variable and right side operand of the assignment operator is a value. The value on the right side must be of the same data-type of the variable on the left side otherwise the compiler will raise an error.

1. **Increment and decrement operator**

Increment Operators are used to increased the value of the variable by one and Decrement Operators are used to decrease the value of the variable by one

1. **Bitwise operator**

These operators are used to perform bit operations. Decimal values are converted into binary values which are the sequence of bits and bit wise operators work on these bits

1. **Conditional operator (ternary operator)**

Conditional operators return one value if condition is true and returns another value is condition is false. This operator is also called as ternary operator.

1. **Special operator**

these are use to assign the value for the variable in c programs. c support comma

operator and size of operator.

**Some example**

**1)write the c program to solve compulation problems using arthmetic operators.**

**Algorithm:**

Step1: start

Step2: input three integer a,b,c assign it value to ‘a’,’b’.

Step3: c=a+b , c=a-b , c=a\*b , c=a/b , c=a%b

Step4:display the result of all arithmetic expression.

Step5:stop

**Program:**

/\*c program to solve compulation problems using arthmetic operators\*/

#include <stdio.h>

int main()

{

int a,b,c;

printf("enter the value of a=");

scanf("%d",&a);

printf("\nenter the value of b=");

scanf("%d",&b);

c=a+b;

printf("a+b=%d\n",c);

c=a-b;

printf("a-b=%d\n",c);

c=a\*b;

printf("a\*b=%d\n",c);

c=a/b;

printf("a/b=%d\n",c);

c=a%b;

printf("modulo division a and b=%d\n",c);

return 0;

}

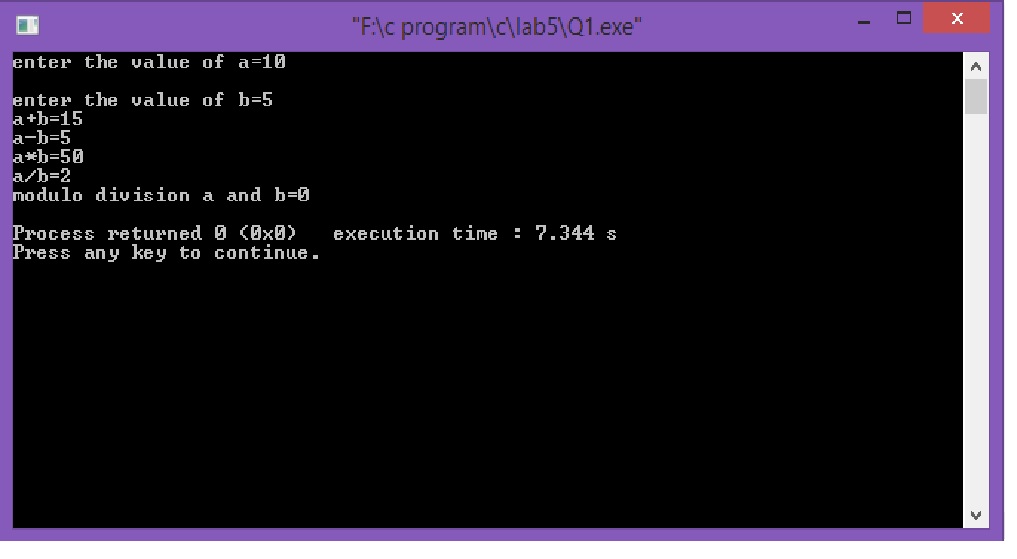
**Flowchart:**

input three integer a,b,c

c=a+b , c=a-b , c=a\*b , c=a/b , c=a%b

display the result of all arithmetic expression

**Output:**

****

**2)write a c program to illaustrate the use of unary prefix increment and decrement operator.**

**Algorithm:**

Step1: start

Step2: input integer a,b,c and assign the value to ‘a’.

Step3: b=++a, b=a++,b=--a, b=a--

Step4: display the result of all postfix and prefix of increment and decrement.

Step5: stop

**Program:**

/\*c program to illaustrate the use of unary prefix increment and decrement operator\*/

#include <stdio.h>

int main()

{

int a,b;

printf("enter a number");

scanf("%d",&a);

b=++a;

printf("the value of b using prefix increment is %d and the value of a is %d",b,a);

b=a++;

printf("\nthe value of b using postfix increment is %d and the value of a is %d",b,a);

b=--a;

printf("\nthe value of b using prefix decrement is %d and the value of a is %d",b,a);

b=a--;

printf("\nthe value of b using postfix decrement is %d and the value of a is %d",b,a);

return 0;

}

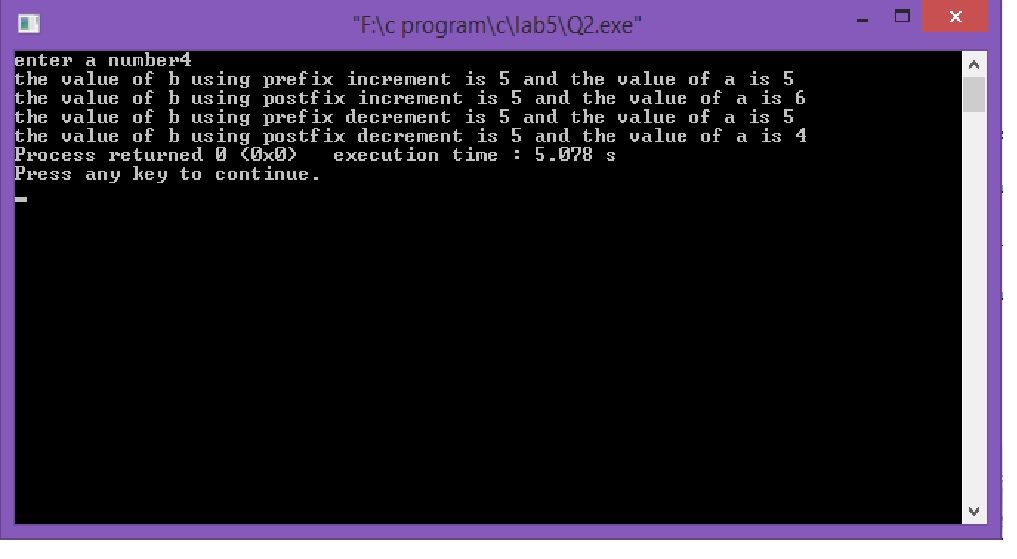
**Flowchart:**

input integer a,b,c

b=++a, b=a++,b=--a, b=a--

display the result of all postfix and prefix of increment and decrement

**Output:**

****

**3)write a c program to find the size of evey data using of oprator.**

**Algorithm:**

Step1: start

Step2: input integer a.

Step3: a=sizeof(“data type”)

Step4: display the result of size of all data type.

Step5: stop

**Program:**

/\*c program to find the size of evey data using of oprator\*/

#include <stdio.h>

int main()

{

int a;

a=sizeof(int);

printf("the size of the int is %d",a);

a=sizeof(float);

printf("\nthe size of the float is %d",a);

a=sizeof(char);

printf("\nthe size of the char is %d",a);

a=sizeof(double);

printf("\nthe size of the double is %d",a);

a=sizeof(long double);

printf("\nthe size of the long double is %d",a);

a=sizeof(short int);

printf("\nthe size of the short int is %d",a);

return 0;

}

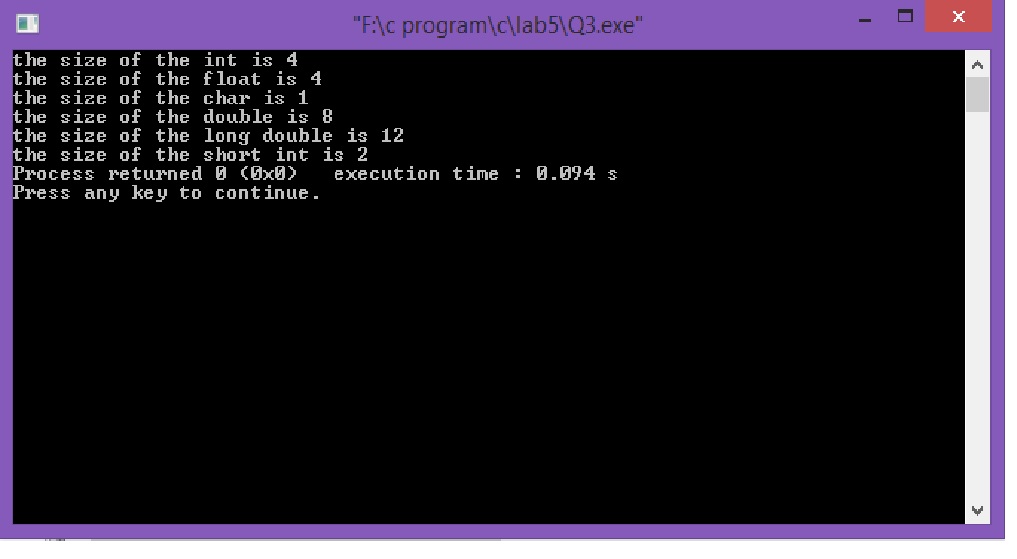
**Flowchart:**

input integer a

a=sizeof(“data type”)

display the result of size of all data type

**Output:**

****

**4)write a c program to solve compulational problems using bitwise logical operator.**

**Algorithm:**

Step1: start

Step2: input integer a,b,c and assign the value of ‘a’,’b’.

Step3: c=a&b, c=a^b, c=~a, c=a<<b , c=a>>b

Step4: display the result of all bitwise operator.

Step5: stop

**Program:**

/\*c program to solve compulational problems using bitwise logical

operator\*/

#include <stdio.h>

int main()

{

int a,b,c;

printf("enter the value of a=");

scanf("%d",&a);

printf("\nenter the value of b=");

scanf("%d",&b);

c=a&b;

printf("\na&b=%d",c);

c=a^b;

printf("\na^b=%d",c);

c=~a;

printf("\na~b=%d",c);

c=a<<b;

printf("\na<<b=%d",c);

c=a>>b;

printf("\na>>b=%d",c);

return 0;

}

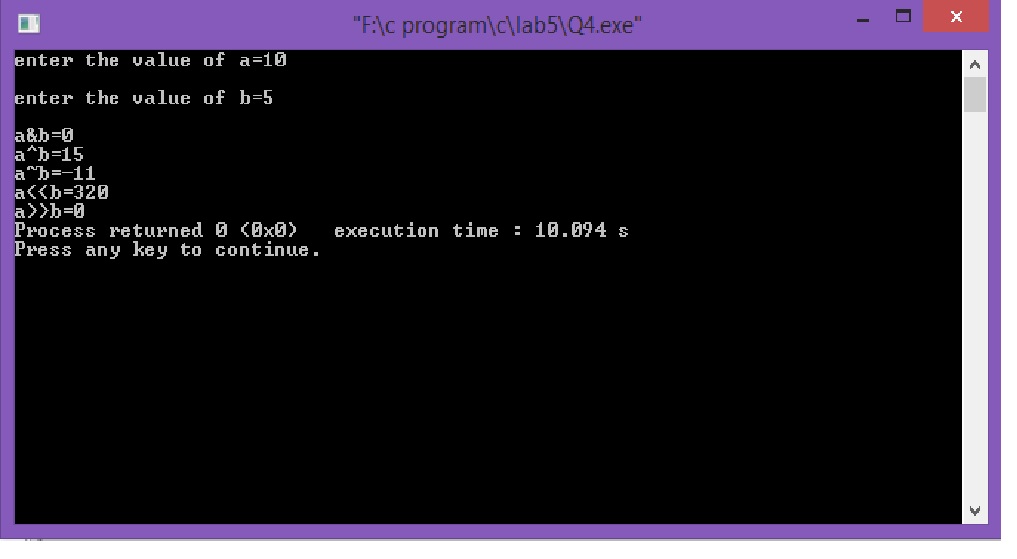
**Flowchart:**

input integer a,b,c

c=a&b, c=a^b, c=~a, c=a<<b , c=a>>b

display the result of all bitwise operator

**Output:**

****

**5)write a c program to find the largest of three number using operator.**

**Algorithm:**

Step1: start

Step2: input integer a,b,c,max and assign the value of ‘a’, ‘b’, ‘c’.

Step3: max=a>b?(a>c?a:b):(b>c?b:c)

Step4: display the value of max.

Step5: stop

**Program:**

/\*c program to find the largest of three number using operator.\*/

#include <stdio.h>

int main()

{

int a,b,c,max;

printf("enter 1st number=");

scanf("%d",&a);

printf("\nenter 2nd number=");

scanf("%d",&b);

printf("\nenter 3rd number=");

scanf("%d",&c);

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

printf("the largest value is=%d",max);

return 0;

}

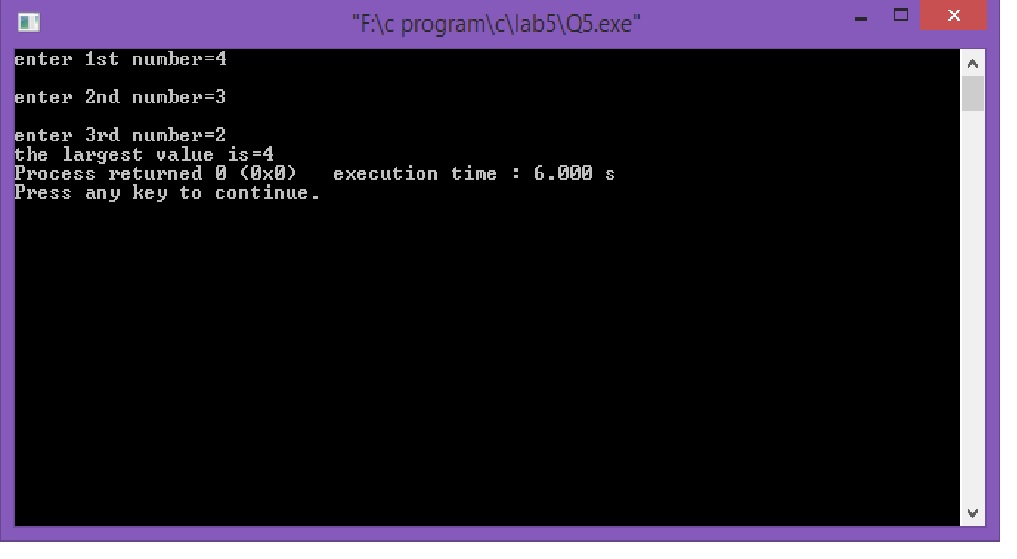
**Flowchart:**

input integer a,b,c,max

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

display the value of max.

**Output:**

****

**6)write a c program to find the root of quadratic rquation (ax2=bx+c=0).**

**Algorithm:**

Step1: start

Step2: input float a,b,c,d,x1,x2,img,real assign the value of ‘a’, ‘b’, ‘c’.

Step3: d=(b\*b)-(4\*a\*c)

Step4: if d>0

x1=(-b+sqrt(d))/(2\*a)

x2=(-b-sqrt(d))/(2\*a)

display the value of first root and second root

else if d=0

x1=(-b/(2\*a))

display the equal root value

else

real=-b/(2\*a)

img=sqrt(-d)/(2\*a)

display imaginary first root or second root

Step5: stop

**Program:**

/\*c program to find the root of quadratic rquation (ax^2=bx+c=0)\*/

#include<stdio.h>

#include<math.h>

int main()

{

float a,b,c,d,x1,x2,img,real;

printf("enter the value of a=");

scanf("%d",&a);

printf("enter the value of b=");

scanf("%d",&b);

printf("enter the value of c=");

scanf("%d",&c);

d=(b\*b)-(4\*a\*c);

if (d>0)

{

x1=(-b+sqrt(d))/(2\*a);

x2=(-b-sqrt(d))/(2\*a);

printf("real and equal first root is=%f second root is=%f",x1,x2);

}

else if (d==0)

{

x1=(-b/(2\*a));

printf("equal the roots is=%f",x1);

}

else

{

real=-b/(2\*a);

img=sqrt(-d)/(2\*a);

printf("imaginary first root is=%f+%fi\n",real,img);

printf("imaginary second root is=%f+%fi",real,img);

}

return 0;

}

**Flowchart:**

input float a,b,c,d,x1,x2,img,real

d=(b\*b)-(4\*a\*c)

If d>0

display the value of first root and second root

x1=(-b+sqrt(d))/(2\*a)

x2=(-b-sqrt(d))/(2\*a)

x1=(-b/(2\*a))

Else if d==0

display the equal root value

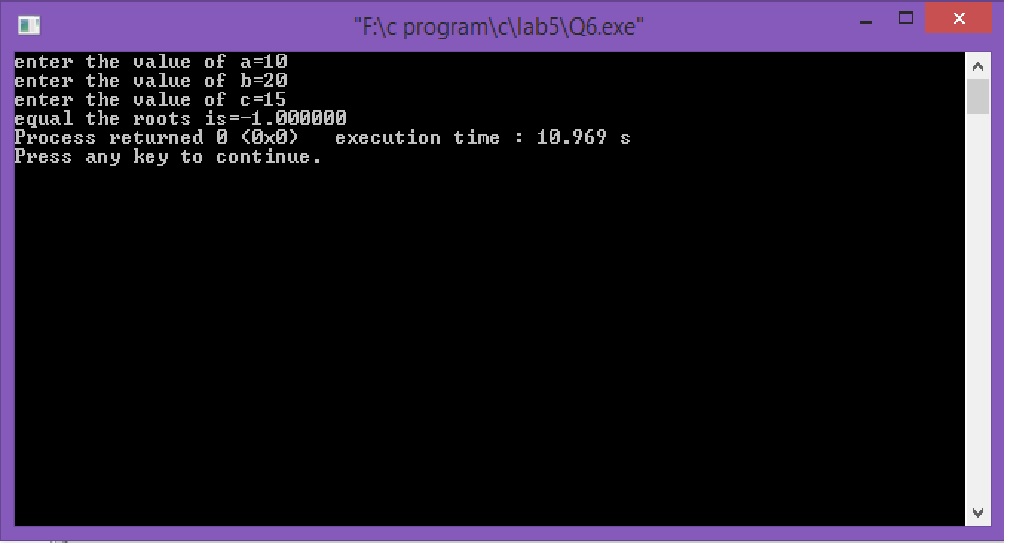
display imaginary first root or second root

real=-b/(2\*a)

img=sqrt(-d)/(2\*a)

If d<0

**Output:**

****

**Conclusion:**

Now we have learned about the usage of operator and learn how to implement the expression in c program and also we learn how to find the size of all data type and also we learn how to find the increment and decrement of the given variable.