# EXERCISE

## Q1. Select the best answer for the following MCQs.

i. **How is a single statement for loop terminated?**

a. With a color
b. With a right brace
c. With a right bracket
d. With a semicolon

ii. **How is a multiple statement for loop terminated?**

a. With a colon
b. With a right brace
c. With a right bracket
d. With a semi colon

iii. **Which of the following can be used to replace ternary operator?**

a. If statement
b. If else statement
c. Else if statement
d. Switch statement

iv. **Which of the following can be used to replace with statement?**

a. If-else statement
b. Break statement
c. Else-if statement
d. While loop

v. **A while loop is more appropriate to use than a for loop when:**

a. When the body of the loop is to be more executed at least once
b. When the loop terminates unexpectedly
c. When the program executes at least once
d. When the number of loop iterations are known In advance

vi. **In which situations a for loop is more appropriate to usE?**

a. When the body of the loop is to be executed at least once
b. When the loop terminates unexpectedly
c. When the program executes at least once
d. When the number of loop iterations are known in advance.

vii. **In which situations a for loop is more appropriate to use?**

a. When the body of the loop is to be executed at least once
b. When the loop terminates unexpectedly
c. When the program executes at least once
d. When the number of loop iterations are known in advance

viii. **Which of the following transfers control to the beginning of the loop, skipping the remaining statements?**

a. Exit function
b. Continue statement
c. Break statement
d. Nested loop

**Answers**

| | | | |
|---|---|---|---|
| i. | With a right bracket | ii. | With a right brace |

| | | | |
|---|---|---|---|
| iii. | Else if statement | iv. | Else if statement |
| v. | The body of the loop is to be executed at least once | vi. | When the body of the loop is to be executed at least once |
| vii. | When the number of loop iterations are known in advance | viii. | Continue statement |

# SHORT QUESTIONS

Q2. Give short answers of the following questions

i. Why control statements are used in C programs?

Answer

Control statements are used in C programs because these statements allow programmers to control the flow of program execution.

ii. Differentiate between if else and else if statements

Answer

Difference

| If else statements | Else if statement |
|---|---|
| If else statement is used when one block of statements is to be executed if the condition is true. If the condition is false then the block | Else if selects a block of statements from multiple options. When it is executed a block of statements is selected for execution based on a |

| | |
|---|---|
| of statements following if are skipped and the other block of statements following else is executed. It selects one block of statements from two options | condition stating from the first condition. If all the conditions are false then it executes the block of statements after else if it exists otherwise program execution continues from the statement after if block |

iii. Differentiate between for and while loop

Answer

Difference

| for loop | While loop |
|---|---|
| A for loop is used to execute one or more statements for a specific number of times. It is also known as counter loop | A while loop is a sentinel loop statement. In this loop the condition is checked at the beginning of the loop. The body of the loop executes as long as the condition remains true. The control can exit a loop in two ways, when the condition becomes false or using break statement |
| **General form**<br>The following is the general form of for loop<br>For (initialization; condition;<br>Increment/decrement)<br>{ | **General form**<br>The following is the general form of while loop<br>While (condition)<br>{<br>Block of statements |

| | |
|---|---|
| Block of statements<br><br>} | } |

iv. What is the usage of break and continue statements in C programs

Answer

Usage of Break Statement

The break statement has two usages

i. It is used to terminate a case in switch statement and program execution continues from the next statement following the switch statement

ii. The break statement is also used to terminate a loop when it is encountered inside a loop and program execution continues from the next statement following the loop

**Usage of Continue Statement**

The continue statement is used inside a loop. When it is encountered, it transfers control to the beginning of the loop skipping the remaining statements.

v. What is the purpose of exit function?

Answer

**Purpose of EXIT() Function**

The purpose of exit() function is to terminate a C++ program before its normal termination and exit to the operating system. It requires the standard library header file stdlib.h.

vi. What is nested loop? Give one example

Answer

### Nested loop

A loop inside another loop is known as nested loop. The C++ language allows to nest a for, while or do while loop inside another for, while or do while loop

### Program

The following program demonstrates the use of nested for loop inside another for loop

```
#include<iostream>
#include<conio.h>
Void main()
{
Int I,j;
For (i=1; i<%; i++)                    // outer loop
{
Cout<<"\n";                            // transfers printing to next line
        For (j=1; j<11; j++)     // inner loop
                Cout<,"*";
}
Getch);
}
```

### Explanation

In this program, I is the outer loop variable and j is the nested loop variable. When this program is executed, the outer loop will execute till the value of i is less than 5 which means it will execute 4 times. The variable i will be initialized to 1 and the statement cout<<"\n"; will transfer the control to next line. The

nested loop will execute and print 10 asterisks (*) on a single line as shown below

**********

The variable i will be incremented by 1 and the nested loop will again print another 10 asterisks. This process will continue till i is less than 5. Hence, this program will print 4 lines of 10 asterisks.

**Output of the program**

The following is the output of the program

**********

**********

**********

**********

**vii. Write the following code using while loop**

```
Sum=0;
For(k=20;k<100);k=k+2)
Sum=sum+K;
Printf("\nSum=%d",sum):
```

**Answer**

```
Int sum=0, k=20:
While (k<100)
{
Sum=sum+k;
Cout<<"\nSum="<<sum;
K=k+2
```

}

viii. Write the following code using switch statement, to produce the same output.

```
If(choice = =1)
Printf("\nSum=%2d",x+y);
Else if (choice = = 2)
Printf("\nProduct = = 2)
Printf("\nProduct=%2d",x*y);
Else
Printf("\nAverage=%2",(x+y)/2);
```

**Answer**

```
Switch(choice)
{
Case 1:
        Printf("/nSum=%2d",x+y);
        Break;
Case 2:
        Printf("\nProduct=%2d",s*y);
        Break;
Default:
        Printf("\nAverage=%2d",(x+y)/2);
}
```

ix. What will be the output of the following code?

```
Int k, sum; k=1; sum=0;
While(k<10)
```

```
{
Sum=sum+k;
Printf("\n%2d\t%2d",k,sum_"
K=k+2;
}
```

**Answer**

| | |
|---|---|
| 1 | **1** |
| **2** | **4** |
| 5 | 9 |
| 7 | 16 |
| 9 | 25 |

**x. What will be the output of the following code?**

```
Int a, b, c:
A=0; b=1; c=2;
A+b+c;
B=+aa;
C=b++;
Printf("\na=%d, b=%d, c=%d",a, b, c);
```

**Answer**

A=4, b=5, c=4

**xi. Write the nested loops that will print the following patterns:**

a. 1 2 3 4 5
1 2 3 4
1 2 3
1 2

b. 1 2 3 4 5
2 3 4 5
3 4 5
4 5

c. *
* *
* * *
* * * *

1 5 * * * * *

**Answer**

a. 1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

**Code**

```
Int main()
{
Int I, j, rows;
Cout<<"Enter the number of rows: ";
Cin>>rows:
For(i=rows; i>=1; - - i)
{
    For(j=1;k<=I;++j)
    {
        Cout<<j<<"";
    }
        Cout<<"\n";
    }
Return 0
}
```

b. Code

```
Int main()
{
    Int I, j:
For(i=1; i<=5; i++)
{
    For (j=i; j<=5; j++
    {
        Cout<<j;
    }
        Cout<<end1;
    }
    Return 0;
}
```

c. Code

```
Int main()
{
    For( int i=1; i<=5; i+=)
      {
    For (int j=1; j<=i; j++)
          Cout<<"*"
}
    Return 0;
}
```

# EXTENSVE QUESTIONS

Q3. Write long answers of the following questions

i. What is decision control structure? Explain all types of if statements with syntax and examples

**Answer**

**Decision making Statements**

Decision making statements are control structures that are used in programming to make decisions. They allow programs to execute a specific statement or a set of statements based on one or more conditions

**Types of Decision making statements**

The decision making statements available in C++ language are as follows

i. If statement
ii. If else statement
iii. Else if statement
iv. Switch statement

1. **If statement**

   If statement is used to execute a block of statement based on a condition

   **General form**

   The following is general form of if statement

   If (condition)

   {

   Block of statements

   }

**Explanation**

1. The condition is evaluated
2. If the condition is true the block of statements within the braces, following the condition is executed
3. If the condition is false, the block of statement is skipped and control is transferred to the next statement after the closing brace
4. If there is a single statement to be executed then braces is not required

**Program**

The following program will read two numbers. If the first number is a positive number then it will print the sum and product of two number

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int x, y, sum, prod:

Cout<<"\nEnter first number:"

Cin>>x;

Cout<<\nEnter second number:";

Cin>>y;

If (x>0)

{

        Sum=x+Y;

        Prod=x*Y;

Cout<<"\nSum="<<sum<<end1;

Cout<<"\nProduct"<<prod;
```

}

Getch():

}

**Output of the program**

The following is the execution of the program

Enter first number :3

Enter second number: 4

Sum=7

Product= 12

**Explanation**

This program prompts the user to enter two numbers which are stored in variables x and y. if the first number (x) is greater than zero then the sum and product of the numbers are calculated and stored in variables sum and prod and their values are printed.

2. **If else statement**

   The if else statement allows making decision between two courses of action based on a condition

   **General Form**

   The following is the general form of if else statement

   If (condition)

   {

   Block of statements-1 // if the condition is true

   }

   Else

   {

   Block of statements=2 // if the condition is false

```
}
```

**Explanation**

1. The condition is evaluated
2. If the result of evaluation is true the first Block of statements 1 is executed the second block of statements-1 is skipped and then control is transferred to the next statement
3. If the condition evaluates to false, the first Block of statements-1 is skipped and the second block of statements-2 following the keyword else is executed
4. If there is a single statements to be executed whether the condition is true or false then braces are no required

**Program**

The following program reads a number and prints whether it is even or odd number

```
#incIde<iostream.h>
#include<conio.h>
Void main()
{
Int n,r:
Cout<<"\nEnter a number: ";
Cin>>n;
R = n % 2;
If(r==0)
    Coul<<n<<"is even number";
Else
```

```
        Cout<<n<<"is odd number";
Getch();
}
```

**Output of the Program**

The following is the execution of the program

Enter a number : 15

15 is odd number

**Explanation**

When this program is executed it prompts the user to enter a number. The number is stored in variable r. the modulo operator(%) gives the remainder after division of the entered number by 2 and it is stored in variable r. if the remainder is equal to 0 then the program will print n is even otherwise it is odd

Braces are not used in if-else statements in this program because a single statement is to be executed whether the condition is true or false

3. **Else-if Statement**

   The else if statement is used in situation where a decision is to be made from several alternatives based on various conditions

   **General form**

   The following is the general form of else if statement

```
If (condition-1)
{
        Block of statements-1
}
Else if (condition-2)
{
        Block of statements-2
}
```

```
Else
{
        Block of statements-n
}
```

**Explanation**

1. The condition-1 is evaluated
2. If it is true the block of statements-1 is executed and control is transferred to the next statements
3. If the condition-1 is false then condition-2 is evaluated if it is true then the block of statements-2 following condition-2 is executed
4. In this manner conditions are evaluated one by one. When any condition is true, the block of statements following that condition is executed, rest of the code is skipped and control is transferred to the next statement
5. If a single statement is to be executed instead of a block of statements the braces are not required

**Program**

The following program reads a number and prints the message whether it is positive number, negative number or it is equal to zero

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int n;
Cout<<"\nEnter a number:";
```

```
Cin>>n;
If (n>0)
        Cout<<n<< "is positive number";
Else if (n < 0)
        Cout<<N<<"is negative number";
Else
        Cout<<n<<"is equal to zero";
Getch();
}
```

**Output of the Program**

The following is the execution of the program

Enter a number – 6

-6 is negative number

**ii. Explain the purpose of switch statement with syntax and one example**

**Answer**

**Switch Statement**

The switch statement is a control statement that is used in programming when a single block of statements is to be selected among many choices. It is very similar to else if statement

**General form**

The following is the general form of switch statement

```
Switch (expression/variable)
```

```
{
Case constant-1:    block of statements
                                Break;
Case constant-2:    block of statements
                                Break;
Case constant-3     block of statements
                                Break;

.
Default:            block of statements
                                Break;
}
```

**Explanation**

1. The expression within the brackets is evaluated
2. The result of the expression or the value of variable is compared in sequence with the constant values given after the keyword case
3. If none of the constant values after the case keyword match with the result of expression or the value of variable then the block of statements following the keyword default is executed. Its use is optional
4. Is switch statement, it is allowed to use a variable within the parenthesis instead of an expression based on which block of statements following a case can be executed
5. The purpose of break statement is to exit the body of the switch statement

**Program**

The following program reads an integer between 1 to 7 that represents a day of week starting from Monday, it prints the name of the day based on the value of day

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int a:

Cout<"\nEnter an integer (1-7):";

Switch (a)

{

Case 1:      cout<<"\nMonday":

                    Break;

Case 2:      cout<<"\nTuesday";

                    Break;

Case 3:      cout<<"\nWednesday":

                    Break;

Case 4:      cout<<"\nThursday";

                    Break;

Case 5:      cout<<"\nFriday";

                    Break;

Case 6:      cout<<"\nSaturday";

                    Break;
```

```
Case 7      cout<<"\nSunday":
                Break;
Default:    cout<<"\nNot a valid day":
}
Getch():
{
```

**Output of the Program**

The following is the execution of the program

Enter an integer : 6

Saturday

iii. **What is looping control structure? Explain all types of looping statements with syntax and examples**

**Answer**

**Looping Control Structure**

A loop is a control structure that repeatedly executes a sequence of statements until condition is true. Loops are also called repetition control structures in C++

**Types of loops in C++**

There are three types of loops in C++ which are as follows

i. For loop
ii. While loop
iii. Do while loop

1. **For loop**

   A for loop is used to execute one or more statements for a specific number of times. It is also known as counter loop.

### General Form

The following is the general form of for loop

For (initialization: condition; increment/decrement)

{

Block of statements

}

### Explanation

1. A variable known as loop counter or loop variable, is assigned an initial value in the initialization part of the loop. For example a=1 or b=50
2. The condition which is a relational expression such as a< 5 is evaluated. If the condition is true then the block of statements within the braces is executed
3. After the execution of the statements control is transferred to the increment/decrement part of the loop. This part consists of an assignment statement such as a= a+7 or a =a-1 that increments or decrements the loop variable
4. If only a single statement is to be executed in for loop then braces are not required

### Program

The following program prints four times the output of statements that are in the for loop

```
#include<iostream.h>

#include<Conio.h>

Void main()

{

Int k;
```

```
For (k=1: k<5: k++) // loop executes four times till k is less than 5
{
Cout<<""\nI am a student "<<end1:
Cout<<"I was born in 2001"<<end1;
}
Getch():
}
```

**Explanation**

When this program is executed the loop counter (k) is initialized to 1. The loop condition k<% is checked. Since, it is true, the loop executes and displays the output of two statements that are within the curly brackets. The loop counter is incremented by 1. The condition is checked and the loop statements are again executed. Each time the loop is executed, the counter k is incremented by 1. The loop continues to execute till k is less than 5. When k becomes 5 It terminates and control is transferred to the next line. In this program k++ is used which is same as k=k+1

**Output of the program**

The following is the execution of the program

I am a student

I was born in 2001

I am a student

I was born in 2001

I am a student

I was born in 2001

I am a student

I was born in 2001

2. **While Loop**

A while loop is a sentinel loop statement. In this loop, the condition is checked at the beginning of the loop. The body of the loop executes as long as the condition remains true. The control cane exit a loop in two ways when the condition becomes false or using breaks statement

**General Form**

The following is the general form of while loop

```
While (condition)
{
    Block of statements
}
```

**Explanation**

1. The condition which is a relational expression such as k<10 is evaluated
2. If the condition evaluates to true, the block of statements within the braces is executed
3. After the execution of statements, control is transferred back to the beginning of the loop and the condition is again evaluated. If it is true then the body of the loop is evaluated again
4. If the body of the loop consists of a single statement then braces are not required

**Program 1**

The following program prints the sum of all the positive numbers up to 15 using a while loop

(sum=1+ 2+ 3+ 4+.....+ 15)

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int k, sum:
Sum=0:
K=1:
While (k<=15)                    // initialization of loop variable
{
Sum=sum+k
K=k+1                            // initialization for loop continuation
}
Cout<<"Sum = "<<Sum:<<end1:
Getch()
}
```

**Explanation**

The variable sum is initialized to 0 and the loop variable k to 1. Each time the loop is executed, the value of k is added to sum and it is incremented by 1. This process continues till the condition k<=15 is true. When k becomes 1 the loop terminates

**Output of the Program**

The following is the output of the program

Sum = 120

### 3. Do while Loop

A do while loop is very similar to while loop except that the condition is checked at the end of the loop. Therefore, the body of the loop is executed at least once

**General Form**

The following is the general form of do while loop

Do

{

Block of statements

}

While (condition);

**Explanation**

1. The block of statements following the keyword do is executed
2. The condition at the end of the loop is evaluated. If the condition is true, control is transferred to the beginning of the loop
3. When the condition becomes false control is transferred to the next statement
4. If the body of the loop consists of a single statement then braces are not required

**Program**

The following program prompts the user to enter two numbers and prints their product. After printing the product, it asks the user whether he/she wants to continue printing product of another set of two numbers. If the user wants to continue, he/she enters the character 'y' otherwise the character 'n'.

```
#include<iostream.h>
void main ()
int a, b, prod:
char ch;
do
{
cout<<"\nEnter two numbers : ":
cin>>a>>b:
cout<<"Product = "<<prod<<endl;
count<<"Do you want to continue? (y/n): ":
cin>>ch;
}
While (ch!='n');
}
```

**Output of the Program**

The following is the execution of the program

Enter two number: 3, 4

Product = 12

Do you want to continue? (y/n) : y

Enter two numbers: 7,8

Product=56

Do you want to continue? (y/n) : y

```
Enter two numbers: 10,12

Product=120

Do you want to continue? (y/n) : n
```

## Lab Activities

i. Practice all the programs given in the chapter.

**Answer**

Practical work.

ii. Write a program that reads a number and prints its square if the number is greater than 10 otherwise prints its cube.

**Program**

```
#include <iostream>

int main ()

{

int number;

std : : cout <<  "Enter the number " ;

std : : cin >> number ;

if (number>10)

        std   cout << "Square of "<< number << "is: "<< number*number ;

else

        std   cout << "Cube of "<< number << "is: "<< number*number*number ;
```

iii. Write program that reads an integer and prints whether it is odd or even number.

Program

```
#include <iostream>

int main ()

{

  int num;

std : : cout << "Enter a number : " ;

std : : cin >> num;

if ( num % 2 ==0)

std : : cout <<"The number " << num << "is even. "<< std end1;

std : : cout <<"The number " << num << "is odd. "<< std end1; return 0,

}
```

iv. Write a program that reads three numbers and prints the largest one.

Program

```
#include <iostream>

int main ()

{

int n1, n2, n3;

std : : cout << "Enter three numbers : " ;

std : : cin >> n1 >> n2 >> n3 ;

if (n1>= n2 && n1 >= n3)
```

```
{
    std cout << "The largest number is : " << n1 ;
}
If (n2 >= n1 && n2 >= n3)
    std cout << "The largest number is : " << n2 ;
}
If (n2 >= n1 && n2 >= n3)
    std cout << "The largest number is : " << n3 ;
        }
return 0
}
```

v. **Write a program that read a letter and prints whether it is a lowercase or uppercase letter**

**Program:**

```
#include<iostream.h>
Int main()
{
Char ch;
Std :: cout<<"Enter any character:";
Std : : cin>>ch;
If (ch>=65 && ch<=90)
    Std : : cout<<"Entered character is in uppercase letter";
```

```
Else if (ch>=97 && ch<=122)
        Std : : cout<< "Entered character is in lowercase letter";
Return 0;
}
```

vi. Write a program that reads an integer and prints its multiplicative table up to 20

Program

```
#include<iostream>
Int main()
{
    Int n;
Std : : cout <<"Enter a positive integer";
Std :: cin>> n;
        For (int i=1: i<=20: ++i)
{
        Std cout << n << "^" << i << "=" << n << std end1;
}
Return 0;
}
```

vii. Sameer works in a firm as a programmer and is getting a monthly pay. Write a program to take his basic pay as input through the keyboard a calculate his net pay. His net pay is to be calculated as given below

Net pay = basic pay + house rent

**Program**

```
#include<iostream>

Int main(0

{

Float basicPay, houserent, netPay;

Std :: cout<<"Sameer's basic pay:";

Std :: cin>> basicPay;

Std :: cout<<"Sameer's house rent";

netPay = basicPay + houserent;

std :: cout<<"Sameer's net pay is: "<<netPay;

return 0;

}
```

viii. Sameer get his house rent on his basic pay according to the scheme given below

| Basic pay | House rent |
|---|---|
| Basic pay < 30000 | 30% of Basic Pay |
| **Basic pay ≥ 30000 and ≤ 50000** | 35% of Basic Pay |
| Basic Pay > 50000 | 40% of Basic Pay |

Write a program to calculate the Net Pay of Sameer

**Program**

```
#include<iostream>

Int main()
```

```
{
Float BasicPay, houseRent, netPay;
Std :: cout<<"Sameer's basic pay";
Std :: cin>>basicpay;
        If (basicPay < 30000)
        {
        netPay= basicPay + ((basicPay*30)/100);
        }
        Else if (BasicPay >= 30000 && basicPay <= 500000)
        {
        netPay = basicPay + ((basicPay*35)/100);
        }
        Else if (basicPay > 50000)
        {
        netPay= basicPay + ((basicPay*40)/100);
        }
Std cout<<"Sameer's net pay is: "<<netPay;
Return 0;
}
```

ix. Write a program that will produce a table of equivalent temperatures in both Fahrenheit and Celsius with an increment of 5 from 50 to 100 as given below

C=5/9 (F-32)

| Fahrenheit | Celsius |
|---|---|
| 50 | 9.90 |
| 55 | 12.65 |

100 37.40

**Program**

```
#include <iostream>
Int main ( )
{
Double fah, cel;
Std : : cout<<"\tFahrenhiet"<<"\t"<<"Celsius"<<std : :end1;
Std : : cout<<"\t------------------------------"<<std: :end1;
For (fah=50; fah<=100; fah+=5)
{
Cel=(fah-32.0)*5.0/9.0'
Std cout<<"\t"<<fah<<"\t\| \t"<<cel<<std   end1;
}
}
```

x. **Write a program that prints the sum of the following sequences using for loop**

**Sum= 30 + 33 + 36 + 39 +.....+ 60**

**Program**

```
#include<iostream>
Int main()
{
Int sum = 0;
      For (int i=30; i<=60; i+=3)
// here we increment of 3 because we sum up the sequences
{
```

```
        Sum=sum + I;
}
Std     cout<< "Sum of the sequences is: << sum << std ent1;
}
```

**xi. Write the above program using while loop**

**Program**

```
#include<iostream>
Int main()
{
Int sum= 0, i=30;
   While (i <=60)
    // here we increment of 3 because we sum up the sequences
    {
     Sum = sum + i;
     I += 3;
     }
Std     cout<<"Sum of the sequences is :  << sum << std   end1;
}
```

**xii. Write a program that prints all the positive odd numbers up to 50 skipping those that are divisible by 5 using continue statement**

**Program**

```
#include <iostream>
```

```
Int main()
{
  For (int i=1: i<=50: I +=2)
  {
        If (I % 5 == 0)
        {
            Continue ;
        }
        Std   cout << i << std end1;
  }
}
```

xiii. **Write a program that reads an integer and prints its factorial**

**Program**

```
#include<iostream>
Int main()
{
      Int n, face =1;
      Std :: cout <<"Enter a number ";
      Std :: cin>>n;
      For (int i=1; i<=n; ++i)
      {
          Fact *=I;
```

```
}
Std cout<< "Factorial of <<n<< is = "<< fact << std end1;
```

xiv. Write a program that reads the coefficients, a, b and c of the quadratic equation
$ax^2+bx+c=0$ and prints the real solution of x, using the formula

Note: that if there is only one real solutions. If it is greater than zero then there are two real solutions and if it is less than zero then print the message "No real solutions"

Program

```
#include <iostream>
#include <cmath>
Int main ( )
{
Double a, b, c;
Double x1=0, x2=0;
Std : :cout<<"Enter coefficients of a, b and c";
Std : :cin>>a>>b>>c;
If (b*b=4*a*c<0)
{
Std Cout<<"No reals solutions"<<std end1;
}
Else if((b*b-4*a*x)==0)
{
X1-b/(4*a*c):
Std: :cout<<"x1 is "<<x1<<std end1;
}
Else
{
X1=(-b+(sqrt(b*b-4*a*c)))/(2*a):
X2=(-b-(sqrt(b*b-4*a*c)))/(2*a):
Std cout<<"x1 is "<<x1<<std: :end1;
Std cout<<"x2 is "<<x2<<std: :end1;
}
}
```