AIM:To Find Roots Of Quadratic Equation.

THEORY:

The if - else statement is used to carry out a logical test and then take one of two possible actions, depending on the outcome of the test (i.e. whether the outcome is true or false).

The else portion of the if - else statement is optional. Thus, in its simplest general form, the statement can be written

```
if (expression)
statement
```

The expression must be placed in parentheses, as shown. In this form, the statement will be executed only if the expression has a nonzero value (i.e., if expression is true). If the expression has a value of zero (i.e., if expression is false), then the statement will be ignored.

The statement can be either simple or compound. In practice, it is often a compound statement which may include other control statements.

The general form of an if statement which includes the else clause is

```
 if (expression)
statement 1
else
statement 2
```

If the expression has a nonzero value (i.e., if expression is true), then statement 1 will be executed. Otherwise (i.e., if expression is false), statement 2 will be executed.

ALGORITHM:

```
1) Start
2) Declare the required variables
3) Read the coefficients a,b,c of quadratic equation
4) Calculate $d=b^2-4ac$
5) IF $d<0$,display" roots are imaginary" & goto step 6
   else
   calculate $x=(-b+ sqrt(d))/2a$
             $y=(-b-sqrt(d))/2a$
6) Display real roots x,y
7) Stop.
```

POST LAB ASSIGNMENT:

1) Explain different data types in c++.

AIM: To Check Whether A Year Is A Leap Year Using if-else.

THEORY:

The if - else statement is used to carry out a logical test and then take one of two possible actions, depending on the outcome of the test (i.e. whether the outcome is true or false).

The else portion of the if - else statement is optional. Thus, in its simplest general form, the statement can be written

```
if (expression)
statement
```

The expression must be placed in parentheses, as shown. In this form, the statement will be executed only if the expression has a nonzero value (i.e., if expression is true). If the expression has a value of zero (i.e., if expression is false), then the statement will be ignored.

The statement can be either simple or compound. In practice, it is often a compound statement which may include other control statements.

The general form of an if statement which includes the else clause is

```
 if (expression)
statement 1
else
statement 2
```

If the expression has a nonzero value (i.e., if expression is true), then statement 1 will be executed. Otherwise (i.e., if expression is false), statement 2 will be executed.

ALGORITHM:

```
1) Start
2) Take year as an input.
3) Check
       If year%4=0 and year % 100 != 0 or year%400=0 then
             print”It is a Leap year”
        else
             print” Not a Leap year”
4) Stop
```

POSTLAB ASSIGNMENTS:

1) Explain if-else control structure.

2) Write short note about operators ,their precedence and associativity.

AIM: To find grade of a student.

THEORY:

The nested if-else structure is

```
if e1 if e2 s1
      else s2
else if e3 s3
      else s4
```

where e1, e2 and e3 represent expressions, and s1, s2, s3 and s4 represent statements. In this situation, one complete if - else statement will be executed if el is nonzero (true), and another complete if - else statement will be executed if el is zero (false). It is, of course, possible that s1, s2, s3 and s4 will contain other if - else statements. We would then have multilayer nesting.

ALGORITHM:

1) Start.

2) Read the marks of a student.

3) Check if the marks is between 100 and 91,display the student has secured first grade.

Else check if the marks is between 90 and 81,display the student has secured second grade.

Else check the marks is between 80 and 71,display the student has secured third grade.

Else check the marks is between 70 and 61,display the student has secured fourth grade.

Else display the student has secured last grade.

4) Stop.

AIM:To Implement Calculator using switch and break statements.

THEORY:

The general format for switch statement is:

```
case expression 1:

                    Block of
                    statements

                    break;

case expression 2 :

                    Block of statements

                    break;

  case expression m :

                    Block of statements

                    break;
```

where expression 1, expression 2, ... , expression m represent constant, integer-valued expressions. Usually, each of these expressions will be written as either an integer constant or a character constant. Each individual statement following the case labels may be either simple or complex.

When the switch statement is executed, the expression is evaluated and control is transferred directly to the group of statements whose case- label value matches the value of the expression. If none of the case-label values matches the value of the expression, then none of the groups within the switch statement will be selected. In this case control is transferred directly to the statement that follows the switch statement.

One of the labeled groups of statements within the switch statement may be labeled default. This group will be selected if none of the case labels matches the value of the expression. (This is a convenient way to generate error messages or error correction routines). The default group may appear anywhere within the switch statement-it need not necessarily be placed at the end. If none of the case labels matches the value of the expression and the default group is not present (as in the above example), then the switch statement will take no action.

ALGORITHM:

1) Start.

```
2) Read two numbers and choice of operation
3) Using a Switch Statement perform the following
    if choice is 1,add two numbers and store the result.
    if choice is 2,subtract two numbers and store the result.
    if choice is 3 ,multiply two numbers and store the result.
    if choice is 4,divide two numbers and store the result.
4) Display the result of operation
5) Stop.
```

POSTLAB ASSIGNMENT :

1) Explain the difference between break and continue using example.

AIM: Calculate Cosine Of An Angle Up to A Particular Precision.

THEORY:

The for statement is the third and perhaps the most commonly used looping statement in C++. This statement includes an expression that specifies an initial value for an index, another expression that determines whether or not the loop is continued and a third expression that allows the index to be modified at the end of each pass.

The general form of the for statement is

for (expression 1; expression 2; expression 3) statement

where expression 1 is used to initialize some parameter (called an index) that controls the looping action, expression 2 represents a condition that must be satisfied for the loop to continue execution, and expression 3 is used to alter the value of the parameter initially assigned by expression 1. Typically, expression 1 is an assignment expression, expression 2 is a logical expression and expression 3 is a unary expression or an assignment expression.

When the for statement is executed, expression 2 is evaluated and tested before each pass through the loop, and expression 3 is evaluated at the end of each pass.

ALGORITHM:

```
1) Start
2) Read the angle in degrees
3) Convert angle to radians using formula angle=(angle*3.142)/180
4) Initialize sum=0,term=1 and sign=1,i=1
5) Calculate sum=sum+sign*term
6) Perform sign=sign*-1 and increment i
7) Calculate term=term*$angle^2$/(2*i)*(2*i-1)
8) Check whether term>0.000001,if true goto step 6 else continue
9) Print the cosine value of an angle
10) Stop.
```

POST LAB ASSIGNMENT :

1) Explain the MATH library functions ex. sqrt,pow.

AIM: Write a program to find Armstrong nos within any range .

Armstrong no ex: $153=1^3+5^3+3^3$

THEORY :

The **while** statement is used to carry out looping operations. The general form of the statement is

while (expression) statement

The included **statement** will be executed repeatedly, as long as the value of **expression** is not zero. This **statement** can be simple or compound, though it is typically a compound statement. It must include some feature which eventually alters the value of **expression,** thus providing a stopping condition for the loop.

In practice, the included **expression** is usually a logical expression that is either true or false. (Remember that **true** corresponds to a nonzero value, and **false** corresponds to zero.) Thus, the **statement** will continue to execute as long as the logical expression is true.

ALGORITHM:

```
1) Start
2) Declare the required no of variables.
3) Read the two range values from user
4) Within that range for every number check whether number is Armstrong or not using following steps
    a) Extract every digit of the no using mod operator
    b) Cube it and add it to sum variable which is initialized to 0.
    c) Continue step a & b till no is not equal to 0.
    d) Then check whether sum variable value is same as original no
    e)  If yes print the no otherwise continue with next no.
5) End.
```

POSTLAB ASSIGNMENT:

1) Explain the difference between while and do while control structure.

AIM: To find octal equivalent of a decimal number.

THEORY:

do-while loop:

do while loop statement consists of a block of code and a Boolean condition. First the code block is executed, and then the condition is evaluated. If the condition is true, the code block is executed again until the condition becomes false.

Syntax:

```
do{
Block of statements
} while(condition);
```

ALGORITHM:

1) Start
2) Read the number
3) Extract remainder by dividing the number by 8.
4) Form the new octal number from the remainder.
5) Divide the number by 10 to form new number.
6) Repeat steps 3 to 5, till number not equal to zero.
7) Stop.

AIM: To perform binary search on elements in 1-D array.

THEORY:

Array:

An array is a data structure which allows a collective name to be given to a group of elements which all have the same type. An individual element of an array is identified by its own unique index. The index must be an integer and indicates the position of the element in the array. Thus the elements of an array are ordered by the index.

Declaration of Array	datatype arrayname[size];
Initialization	arrayname[ ]={ set of values};
Accessing Array elements:	arrayname[index];

ALGORITHM

```
1)  Start.
2)  Read the number of elements in search
3)  Read the elements and store in array.
4)  Read the number to be searched and store it in variable x.
5)  Intialize  min = 0 and max := N-1
6)     Repeat steps 6 to 8 till min>max
7)      mid = (min+max) / 2;
8)      check x > A[mid] then min = mid + 1
    else  max := mid – 1
9)     check A[mid] equals elements to be searched ‘x’
    then display position of element as mid
    else display element not found.
10)    Stop
```

POSTLAB ASSIGNMENTS:

1) Explain parameter passing techniques :call by value and call by reference.

AIM: Write a program to reverse the words of a String.

ALGORITHM:

1)Start

2)Read any string from user.

3)Search for a space within a string by checking it's individual character , whenever space is Encountered ,reverse all the characters before it.

4)Repeat step 3 till the end of the string is not Encountered

5)Print the string after reversal.

6)Stop

POSTLAB ASSIGNMENT:

1) Explain the 5 inbuilt string functions.

AIM: To find transpose of a matrix.

THEORY:

Array 2-D declaration: datatype arrayname[row][col];

Array 2-D element access: arrayname[row_index][col_index];

Array 2-D initialization: arrayname[row_index][col_index]={ { elements of first row},

{elements of second row}, ….} ;

ALGORITHM:

1) Start.
2) Read the number of rows and columns
3) Read the elements in the matrix
4) Interchange the row and column index and print the matrix.
5) Stop.

POSTLAB:

1) Write a program for matrix multiplication.

AIM: Write a program to compute GCD of two numbers using Dijkstra's algorithm using function

THEORY:
Using functions we can structure our programs in a more modular way, accessing all the potential that structured programming can offer to us in C++.

A function is a group of statements that is executed when it is called from some point of the program. The following is its format:

type name (parameter1, parameter2, ...) { statements }

where:

- type is the data type specifier of the data returned by the function.
- name is the identifier by which it will be possible to call the function.
- parameters (as many as needed): Each parameter consists of a data type specifier followed by an identifier, like any regular variable declaration (for example: int x) and which acts within the function as a regular local variable. They allow to pass arguments to the function when it is called. The different parameters are separated by commas.
- statements is the function's body. It is a block of statements surrounded by braces { }.

ALGORITHM:

1) Start
2) Declare the required no. of variables.
3) Read the two nos from the user whose GCD is to be computed.
4) Pass the two values to a function which first finds the maximum between the two and then subtracts smaller form the larger.
5) Call the function again with new values.
6) Repeat step 4 & 5 until the two numbers becomes equal.
7) Print a number as GCD.
8) Stop.

POSTLAB ASSIGNMENT:

1) Write short note on recursive function.

AIM: To calculate bonus on salary using function with default arguments.

THEORY:

To set default arguments to function:

When declaring a function we can specify a default value for each of the last parameters. This value will be used if the corresponding argument is left blank when calling to the function. To do that, we simply have to use the assignment operator and a value for the arguments in the function declaration. If a value for that parameter is not passed when the function is called, the default value is used, but if a value is specified this default value is ignored and the passed value is used instead.

Syntax:

returntype function _name(datatype argument1=value1, datatype argument2=value2, …)

{ Block of statements}

ALGORITHM:

1) Start
2) Read the salary of the user.
3) Check the salary, if salary is greater than 10K, pass the salary and bonus to the function
   else pass only the salary to the function that calculates bonus.
4) Stop

POSTLAB:

AIM: To find factorial of a number.

THEORY:

Recursion:
Simply put, recursion is when a function calls itself. That is, in the course of the function definition there is a call to that very same function. It seems our function will never finish. This might be true is some cases, but in practice we can check to see if a certain condition is true and in that case exit (return from) our function. The case in which we end our recursion is called a base case . Additionally, just as in a loop, we must change some value and incremently advance closer to our base case.

ALGORITHM:

1) Start
2) Read the number
3) Pass the number to factorial function and print the result.
4) Stop.

Algorithm for factorial function

```
1) Check if number equals 1, if true return 1.
   else return number * call to recursive function with decremented number.
```

POSTLAB:

1)Implement Fibonacci series using recursion.