



DEPARTMENT OF BASIC SCIENCE AND HUMANITIES  
INSTITUTE OF ENGINEERING AND MANAGEMENT,  
KOLKATA

## **“SCIENTIFIC CALCULATOR”**

**Submitted by:-**

Name of the Student: Debanjan Dutta

Enrolment Number: 12022002002154

Registration Number: 221040110239

Section: B

Class Roll Number: 66

Stream: CSE

Subject: Programming for Problem Solving Using C

Subject Code: ESC-103 (Pr)

Under the supervision of:-

**Prof. Swarnendu Ghosh**

**Academic Year: 2022-26**

(PROJECT REPORT SUBMITTED IN FULFILLMENT OF THE  
REQUIREMENTS FOR THE SECOND SEMESTER)



## **CERTIFICATE OF RECOMMENDATION**

We hereby recommend that the project prepared under our supervision by **Debanjan Dutta**, entitled “**Scientific Calculator**” be accepted in fulfillment of the requirements for the degree of fulfillment of the second semester.

---

Head of the Department  
IEM, Kolkata

---

Project Supervisor  
Basic Science and Humanities

# ***1. Introduction:***

In this project, we aim to build a scientific calculator using the C programming language. The calculator will be capable of performing various mathematical operations, including trigonometric and logarithmic functions. The program will be developed using a modular approach, with each function implemented separately and integrated into a main program.

## ***2. Variable Description:***

The different variables used in this project are listed under: -

- long long intResult: a global variable that holds the integer result of the calculation.
- double k: a global variable that is used as a flag to determine whether a single input or two inputs should be taken.
- double result: a global variable that holds the result of the calculation.
- double a, b: local variables used in the addition, subtraction, multiplication, and division functions to hold input values.
- int ch: a local variable used in the menu() function to hold the user's choice.
- long long n: a local variable used in the factorial() function to hold the input value for the factorial calculation.
- long long f, i: local variables used in the factorial() and factorialReturn() functions to hold intermediate values during the factorial calculation.
- double PI: a preprocessor definition for the constant value of pi.
- long long a, b: local variables used in the mod() function to hold input values for the remainder calculation.

## ***3. Function Description:***

The program will have several functions that will be used to perform mathematical operations. These functions will include: -

- Menu: A function that displays a menu and returns the user's choice.
- Addition: returns the sum of num1 and num2
- Subtraction: returns the difference between num1 and num2
- Multiplication: returns the product of num1 and num2
- Division: returns the quotient of num1 and num2
- Remainder: returns the remainder when num1 is divided by num2
- Factorial: returns the factorial of num
- Sine: returns the sine value of the given angle in degrees
- Cosine: returns the cosine value of the given angle in degrees
- Tangent: returns the tangent value of the given angle in degrees
- Log base e: returns the natural logarithm (log base e) of num
- Log base 10: returns the logarithm (log base 10) of num
- Exponential: returns the exponential value of num (e raised to the power of num)
- Square root: returns the square root of num
- Cube root: returns the cube root of num
- Power: returns the value of base raised to the power of exponent
- Absolute value: returns the absolute value of num
- Sin inverse: returns the inverse sine value of the given value in degrees
- Cos inverse: returns the inverse cosine value of the given value in degrees
- Tangent inverse: returns the inverse tangent value of the given value in degrees
- Ceiling function: returns the smallest integer greater than or equal to num

- Floor function: returns the largest integer less than or equal to num
- Permutation: returns the number of permutations of r objects from a set of n objects
- Combination: returns the number of combinations of r objects from a set of n objects.

## 4. Program:

### *Scientific Calculator using C*

```
// Scientific Calculator Program (in C)

// Header Files
#include <stdio.h>
#include <math.h> // for Trigonometric, Logarithmic and Exponential function
#define PI 3.141592654

// Global variables to hold Results
long long intResult = 0;
double k = 0, result = 0;

// Menu function (to be called in switch)
int menu()
{
    int ch;
    printf("\n1. Addition");
    printf("\n2. Subtraction");
    printf("\n3. Multiplication");
    printf("\n4. Division");
    printf("\n5. Remainder");
    printf("\n6. Factorial");
    printf("\n7. Sine");
    printf("\n8. Cosine");
    printf("\n9. Tangent");
    printf("\n10.log(base e)");
    printf("\n11.log(base 10)");
    printf("\n12.e^x");
    printf("\n13.SquareRoot");
    printf("\n14.CubeRoot");
    printf("\n15.Power");
    printf("\n16.Absolute Value");
    printf("\n17.Sine Inverse");
    printf("\n18.Cosine Inverse");
    printf("\n19.Tangent Inverse");
    printf("\n20.Ceil Function");
    printf("\n21.Floor Function");
    printf("\n22.Permutation (nPr)");
    printf("\n23.Combination (nCr)");
    printf("\n24.Clear");
    printf("\n25.Exit");
}
```

```

    printf("\nEnter your choice: ");

    scanf("%d", &ch);
    return ch;
}

// Function to add numbers
void addition()
{
    double a, b;
    if(k)
    {
        printf("\nEnter a number: ");
        scanf("%lf", &a);
        result += a;
        printf("\nResult = %lf", result);
    }
    else
    {
        printf("\nEnter two numbers: ");
        scanf("%lf%lf", &a, &b);
        result = a + b;
        printf("\nResult = %lf", result);
    }
}

// Function to subtract numbers
void subtraction()
{
    double a, b;
    if(k)
    {
        printf("\nEnter a number: ");
        scanf("%lf", &a);
        result -= a;
        printf("\nResult = %lf", result);
    }
    else
    {
        printf("\nEnter two numbers: ");
        scanf("%lf%lf", &a, &b);
        result = a - b;
        printf("\nResult = %lf", result);
    }
}

// Function to multiply numbers
void multiplication()
{
    double a, b;

```

```

    if(k)
    {
        printf("\nEnter a number: ");
        scanf("%lf", &a);
        result *= a;
        printf("\nResult = %lf", result);
    }
    else
    {
        printf("\nEnter two numbers: ");
        scanf("%lf%lf", &a, &b);
        result = a * b;
        printf("\nResult = %lf", result);
    }
}

// Function to divide numbers
void division()
{
    double a, b;
    if(k)
    {
        printf("\nEnter a number: ");
        scanf("%lf", &a);
        if(a!=0)
        {
            result /= a;
            printf("\nResult = %lf", result);
        }
        else
        {
            printf("Math Error\n");
        }
    }
    else
    {
        printf("\nEnter two numbers: ");
        scanf("%lf%lf", &a, &b);
        if (b!=0)
        {
            result = a / b;
            printf("\nResult = %lf", result);
        }
        else
        {
            printf("Math Error\n");
        }
    }
}
}

```

```

// Function to find remainder
void mod()
{
    long long a, b;
    if(k)
    {
        printf("\nEnter a number: ");
        scanf("%lld", &a);
        intResult %= a;
        printf("\nResult = %d", intResult);
    }
    else
    {
        printf("\nEnter two numbers: ");
        scanf("%lld%lld", &a, &b);
        intResult = a % b;
        printf("\nResult = %lld", intResult);
    }
}

// Function to calculate factorial of a number
void factorial()
{
    long long n, f, i;
    printf("\nEnter a number: ");
    scanf("%lld", &n);
    f = 1;
    for(i = 1; i<=n; i++)
    {
        f = f * i;
    }
    intResult = f;
    printf("\nResult = %lld", intResult);
}

long long factorialReturn(long long n)
{
    long long f, i;
    f = 1;
    for(i = 1; i<=n; i++)
    {
        f = f * i;
    }
    return f;
}

// Function to calculate sine of angle in radians
void sine()
{
    double a;

```

```

    printf("Enter angle in radians: ");
    scanf("%lf", &a);
    result = sin(a);
    printf("\nResult = %lf", result);
}

// Function to calculate cosine of angle in radians
void cosine()
{
    double a;
    printf("Enter angle in radians: ");
    scanf("%lf", &a);
    result = cos(a);
    printf("\nResult = %lf", result);
}

// Function to calculate tangent of angle in radians
void tangent()
{
    double a;
    printf("Enter angle in radians: ");
    scanf("%lf", &a);
    result = tan(a);
    printf("\nResult = %lf", result);
}

// Function to calculate Log (base e)
void logBasee()
{
    double a;
    printf("Enter a number: ");
    scanf("%lf", &a);
    if(a<=0.0)
    {
        printf("Math Error\n");
    }
    else
    {
        result = log(a);
        printf("\nResult = %lf", result);
    }
}

// Function to calculate Log (base 10)
void logBase10()
{
    double a;
    printf("Enter a number: ");
    scanf("%lf", &a);
    if(a<=0.0)

```



```

    {
        printf("Math Error\n");
    }
    else
    {
        result = log10(a);
        printf("\nResult = %lf", result);
    }
}

// Function to calculate e^x
void eToPowerX()
{
    double a;
    printf("Enter a number: ");
    scanf("%lf", &a);
    result = exp(a);
    printf("\nResult = %lf", result);
}

// Function to find the Square Root of a Number
void squareRoot()
{
    int n;
    printf("\nEnter a number: ");
    scanf("%d", &n);
    if (n<0)
    {
        printf("Math Error\n");
    }
    else
    {
        result = sqrt(n);
        printf("\nResult = %lf", result);
    }
}

// Function to find the Cube Root of a Number
void cubeRoot()
{
    int n;
    printf("\nEnter a number: ");
    scanf("%d", &n);
    result = cbrt(n);
    printf("\nResult = %lf", result);
}

// Function to find the Power of a Number
void power()
{

```

```

    double base, expo;
    printf("Enter a base number: ");
    scanf("%lf", &base);
    printf("Enter an exponent: ");
    scanf("%lf", &expo);
    result = pow(base, expo);
    printf("%.11f^%.11f = %.21f", base, expo, result);
}

// Function to find the Absolute Value of a Number
void absolute()
{
    int n;
    printf("\nEnter a number: ");
    scanf("%lld",&n);
    intResult = abs(n);
    printf("\nResult = %lld", intResult);
}

// Function to compute the arc sine(inverse sine) of an argument
void sineInverse()
{
    double n;
    printf("\nEnter a number: ");
    scanf("%lf",&n);
    if(n>1 || n<-1) // Parameter not in Range
    {
        printf("Not in Range");
    }
    else
    {
        result = asin(n);
        printf("Inverse of sin(%.2f) = %.21f in radians\n", n, result);

        // converting radians to degree
        result = asin(n)*180/PI;
        printf("Inverse of sin(%.2f) = %.21f in degrees\n", n, result);
    }
}

// Function to compute the arc cosine(inverse cosine) of an argument
void cosineInverse()
{
    double n;
    printf("\nEnter a number: ");
    scanf("%lf",&n);
    if(n>1 || n<-1) // Parameter not in Range
    {
        printf("\nNot in Range");
    }
}

```

```

else
{
    result = acos(n);
    printf("\\nInverse of cos(%.2f) = %.21f in radians\\n", n, result);

    // converting radians to degree
    result = acos(n)*180/PI;
    printf("\\nInverse of cos(%.2f) = %.21f in degrees\\n", n, result);
}
}

// Function to compute the arc tangent(inverse tangent) of an argument
void tangentInverse()
{
    double n;
    printf("\\nEnter a number: ");
    scanf("%lf",&n);
    result = atan(n);

    printf("\\nInverse of tan(%.2f) = %.2f in radians", n, result);

    // Converting radians to degrees
    result = (result * 180) / PI;
    printf("\\nInverse of tan(%.2f) = %.2f in degrees", n, result);
}

// This function gives the smallest integer that is greater than or equal to
Number
void ceilF()
{
    double n;
    printf("\\nEnter a number: ");
    scanf("%lf",&n);
    result = ceil(n);
    printf("Ceiling integer of %.2f = %f", n, result);
}

// This function gives the largest integer that is smaller than or equal to Number
void floorF()
{
    double n;
    printf("\\nEnter a number: ");
    scanf("%lf",&n);
    result = floor(n);
    printf("Floor integer of %.2f = %f", n, result);
}

// Function to calculate Permutations (nPr)
void npr()
{

```

```

    long long n, r;
    printf("\nEnter two numbers: ");
    scanf("%lld%lld", &n, &r);
    if(n>=r && n>0 && r>=0)
    {
        intResult = factorialReturn(n) / factorialReturn(n - r);
        printf("\nResult = %lld", intResult);
    }
    else
    {
        printf("Math Error\n");
    }
}

// Function to calculate Combinations (nCr)
void ncr()
{
    long long n, r;
    printf("\nEnter two numbers: ");
    scanf("%lld %lld", &n, &r);
    if(n>=r && n>0 && r>=0)
    {
        intResult = factorialReturn(n) / (factorialReturn(r) * factorialReturn(n -
r));
        printf("\nResult = %lld", intResult);
    }
    else
    {
        printf("Math Error\n");
    }
}

// Function to reset global variables
void clear()
{
    printf("\nOld Data Cleared");
    intResult = 0;
    result = 0;
    k = 0;
}

// MAIN
void main()
{
    int l = 0;

    // Main execution Loop
    while(1)
    {

```

```

// Displaying the current results
printf("\n    Old Decimal Result = %f", result);
printf("\n    Old Integer Result = %d", intResult);

// Switch Menu
switch(menu())
{
    case 1: addition();
            k = 1;
            break;

    case 2: subtraction();
            k = 1;
            break;

    case 3: multiplication();
            k = 1;
            break;

    case 4: division();
            k = 1;
            break;

    case 5: mod();
            k = 1;
            break;

    case 6: factorial();
            k = 1;
            break;

    case 7: sine();
            k = 1;
            break;

    case 8: cosine();
            k = 1;
            break;

    case 9: tangent();
            k = 1;
            break;

    case 10: logBasee();
            k = 1;
            break;

    case 11: logBase10();
            k = 1;
            break;
}

```

```
case 12: eToPowerX();  
    k = 1;  
    break;  
  
case 13: squareRoot();  
    k = 1;  
    break;  
  
case 14: cubeRoot();  
    k = 1;  
    break;  
  
case 15: power();  
    k = 1;  
    break;  
  
case 16: absolute();  
    k = 1;  
    break;  
  
case 17: sineInverse();  
    k = 1;  
    break;  
  
case 18: cosineInverse();  
    k = 1;  
    break;  
  
case 19: tangentInverse();  
    k = 1;  
    break;  
  
case 20: ceilF();  
    k = 1;  
    break;  
  
case 21: floorF();  
    k = 1;  
    break;  
  
case 22: npr();  
    k = 1;  
    break;  
  
case 23: ncr();  
    k = 1;  
    break;  
  
case 24: clear();
```

```

        break;

    case 25:l = 1;
        break;

    default:
        printf("\nInvalid Choice !");
    }
    // Waiting for a button to be pressed
    printf("\nPress any button to continue.....");
    getch();

    // Clear screen command to clear screen after each menu iteration
    system("cls");

    // To break out of this menu loop
    if(l == 1)
        break;
}
}

```

## 5. Outputs:

Sample outputs (screenshots) to demonstrate the functionalities in programs are listed below.

```

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Remainder
6. Factorial
7. Sine
8. Cosine
9. Tangent
10.log(base e)
11.log(base 10)
12.e^x
13.SquareRoot
14.CubeRoot
15.Power
16.Absolute Value
17.Sine Inverse
18.Cosine Inverse
19.Tangent Inverse
20.Ceil Function
21.Floor Function
22.Permutation (nPr)
23.Combination (nCr)
24.Clear
25.Exit
Enter your choice: █

```

DIVISION:

Enter your choice: 4

Enter two numbers: 45

5

Result = 9.000000

Press any button to continue.....

CUBE ROOT:

Enter your choice: 14

Enter a number: 27

Result = 3.000000

Press any button to continue.....

SINE:

Enter your choice: 7

Enter angle in radians: 180

Result = -0.801153

Press any button to continue.....

LOG BASE 10:

Enter your choice: 11

Enter a number: 2

Result = 0.301030

Press any button to continue.....

POWER:

Enter your choice: 15

Enter a base number: 4

Enter an exponent: 5

$4.0^{5.0} = 1024.00$

Press any button to continue.....

PERMUTATION:

Enter your choice: 22

Enter two numbers: 4

3

Result = 24

Press any button to continue.....



COMBINATION:

Enter your choice: 23

Enter two numbers: 5

4

Result = 5

Press any button to continue.....

FLOOR:

Enter your choice: 21

Enter a number: 6.987

Floor integer of 6.99 = 6.000000

Press any button to continue.....

EXIT:

Enter your choice: 24

Old Data Cleared

Press any button to continue.....

**THANK YOU!**