Contents

INTRODUCTION

Scientific Calculator:

The calculator was written by Rolf Howarth in 1996.

A fully featured scientific calculator with proper operator is implemented including logarithmic, trigonometric functions, Simple Interest, Compound Interest, Addition, Subtraction, Multiplication, Division, Modulo, Power value, Square, Cube, Square root, Cube root, LCM, HCF, Integration, Permutations and Combinations.

The calculator is written in C language for the personal mini-project. In this Calculator we can give our own value and choose the operator we want to do calculation which will give you the perfect result. The libraries used for implementation in project are <stdio.h>, <math.h>

<stdlib.h> and the function used is fact().

Basic Functions

Addition

The addition (Sum function) is used by clicking “+” button, which gives the result as a + b.

Subtraction

The subtraction(minus function) is used by clicking “+” button, which gives the result as a - b.

Multiplication

The Multiplication (product function) is used by clicking on the “*” button. The function results in a * b.

Division

The division(divide function) is used by clicking on the “/” button. The function results in a/b.

Modulo

The modulo function (remainder) is used by clicking on the “%” button. The result of the function is a % b.

LCM

The LCM (Lowest Common Factor) function is used by clicking on the“L”button. The results of the function will be the smallest number that is a multiple of each value.

HCF

The HCF (Highest Common Factor) function is used by clicking on the “H” button. The result of the function will be the highest number that is multiple of each value.

SI

The SI (Simple Interest) function is used by clicking on the “S” button. The result of the function will be (principle * rate * time) / 100.

CI

The CI (Compound Interest) function is used by clicking on the “C” button. The result of the function will be (principle * (1+ (rate/100) ^time) – principle.

Integration

The Integration function is used by clicking on the “I”button. The result of the function will be area of curve in the graph.

Power

The power function is used by clicking on the “^” button. The result of the function will be the a ^ b.

Square root

The square root function is used by clicking on the “4” button. The result of the function will be the x ^ 2.

Cube Root

The Cube root function is used by clicking on the “5” button. The result of the function will be the x ^ 3.

Sine

The Sine function is used by clicking on the “s” button. The result of the function will be the sin (a), which needs the input in degree.

Cosine

The Cosine function is used by clicking on the “c” button. The result of the function will be the cos (a), which needs the input in degree.

Tangent

The Tangent function is used by clicking on the “t” button. The result of the function will be the tan (a).

Logarithms

The Logarithms function is used by clicking on the “l” button. The result of the function will be the log a base e.

Permutation

The permutation function is used by clicking on the “p” button. The result of the function will be the npr = n! / (n-r)!

Combination

The Combination function is used by clicking on the “b” button. The result of the function will be the ncr = n! / r! (n – r)!.

Square

The Square func is used by clicking on the “2”button. The result of the function will be a ^2.

Cube

The Cube func is used by clicking on the “3” button. The result of the function will be a ^ 3.

Inverse

The Inverse function is used by clicking on the “i” button. The result of the function will be the 1/b.

PROPOSED SYSTEM

The following documentation is a project the “Name of term paper allotted”. It is a detailed summary of all the drawbacks of the old system and how the new proposed system overcomes these shortcomings. The new system takes into account the various factors while designing a new system. It keeps into account the economical bandwidth available for the new system.

The foremost thing that is taken care is the need and requirements of the user.

Description :

Before developing software we keep following things in mind that we can develop powerful and quality software.

Problem Statement :

Problem statement was to design a module:

- Which is user friendly.
- Which will restrict the user from accessing other user’s data.
- Which will help user in viewing his data and privileges.
- Which will help the administrator to handle all the changes.
- Which is multifunctional.

Function to be provided :

The system will be user friendly and completely menu driven so that the users shall have no problem in using all options.

- The system will be efficient and fast in response.
- The system will be customized according to needs.

System Requirement :

Operating System: Windows XP, Windows 10, Linux, Mac OS

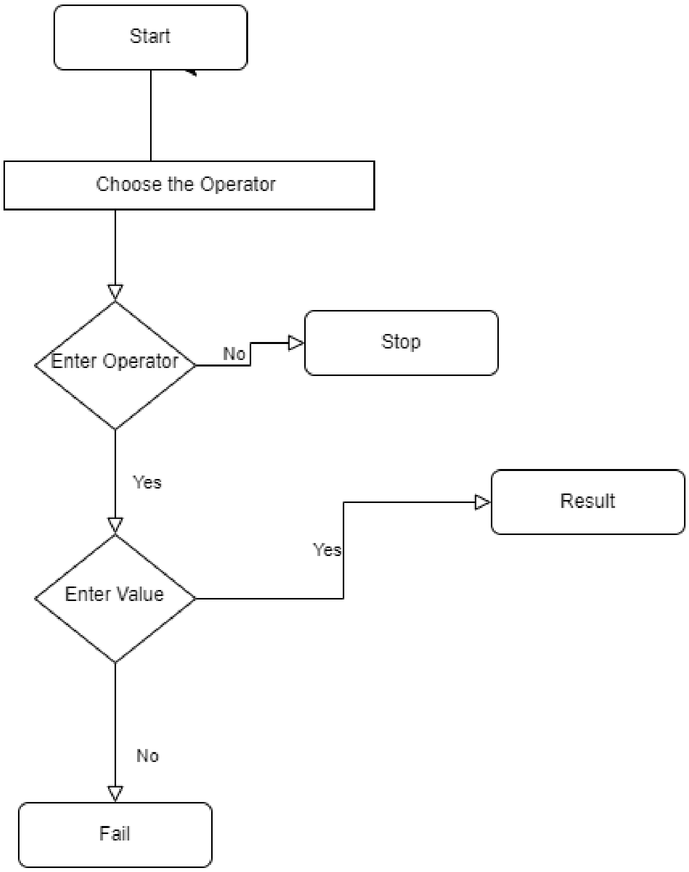
Language: C Language

Processor: Pentium IV, Intel core i3, Ryzen, RAM : 512 MB . Hard disk: 2 GB

System Design

Then we began with the design phase of the system. System design is a solution of 5W and 1H, “Why” “What”, “Who”, “When”, “Where”and “How to” approach to the creation of a new system. It translates system requirements into ways by which they can be made operational. It is a translational from a user oriented document to a document oriented programmers. For that it provides the understanding and procedural details necessary for the implementation. Here we use Flowchart to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs. It should overcome all the drawbacks of the old existing system and most important of all meet the user requirements.

Flow Chart



Coding :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define N 1000
int fact(int);
void main()
{
  // Declataion of Variable
  float valueOne, valueTwo;
  char operator;

        printf("Addition = + \n");
        printf("Subtraction = - \n");
        printf("Multiplication = * \n");
        printf("Division = / \n");
        printf("Modulo = % \n");
        printf("LCM = L \n");
        printf("HCF = H \n");
        printf("Simple Interest = S \n");
        printf("Compound Interest = C \n");
        printf("Integration = I \n");
        printf("power = ^ \n");
        printf("Square root =  4 \n");
        printf("Cube root =  5 \n");
        printf("sin =  s\n");
        printf("cos =  c\n");
        printf("tan = t\n");
        printf("log base e = l\n");
        printf("Permutation = p\n");
        printf("Combination = b\n");
    // Operator Declaration for the calculation
    printf("Enter the calculation Operator\n");
    scanf("%c", &operator);
  // Starting of switch case for conditional statements.
    switch(operator)
    {
      case '+':
      // Addition
        printf("Enter Value 1 and Value 2:\n");
        scanf("%f %f", &valueOne, & valueTwo);
        float answer3 = (valueOne)+(valueTwo);
        printf("Your Result of Addition is : %.2f", answer3);
        break;
      case '-':
      //Subtraction
        printf("Enter Value 1 and Value 2:\n");
        scanf("%f %f", &valueOne, & valueTwo);
        float answer4 = (valueOne)-(valueTwo);
        printf("Your Result of Subtraction is : %.2f", answer4);
        break;
```

```
case '*':
// Multiplication
  printf("Enter Value 1 and Value 2:\n");
  scanf("%f %f", &valueOne, & valueTwo);
  float answer2 = (valueOne)*(valueTwo);
  printf("Your Result of Multiplication is : %.2f", answer2);
  break;
case '/':
//Division
  printf("Enter Value 1 and  Value 2:\n");
  scanf("%f %f", &valueOne, & valueTwo);
  float answer1 = (valueOne)/(valueTwo);
  printf("Your Result of division is : %.2f", answer1);
  break;
case '%':
// Modulo
  printf("Enter Value 1 and Value 2:\n");
  scanf("%f %f", &valueOne, & valueTwo);
  int answer7 = (int)valueOne % (int)valueTwo;
  printf("Your Modulo value is : %d", answer7);
  break;
case 'L':
// LCM
  printf("Enter Value 1 and Value 2 for LCM :\n");
  scanf("%f %c %f", &valueOne, &operator, & valueTwo);
  int v1 = (int)valueOne;
  int v2 = (int)valueTwo;
  int max = (v1 > v2) ? v1 : v2;
  while (1) {
    if (max % v1 == 0 && max % v2 == 0) {
      printf("The LCM of %d and %d is : %d", v1, v2, max);
      break;
    }
    ++max;
  }
  break;
case 'H':
// HCF
  printf("Enter Value 1 and Value 2 for HCF :\n");
  scanf("%f %f", &valueOne, & valueTwo);
  int i, HCF;
  int val1 = (int)valueOne;
  int val2 = (int)valueTwo;
  for(i=1; i <= val1 && i <= val2; ++i)
    {
    // Checks if i is factor of both integers
      if(val1%i==0 && val2%i==0)
        HCF = i;
    }
  printf("H.C.F of %d and %d is : %d", val1, val2, HCF);
  break;
case 'S':
//Simple Interest
  printf("Enter Principle and Rate :\n");
```

```c
  scanf("%f %f", &valueOne, & valueTwo);
  int t;
  printf("Enter Time:\n");
  scanf("%d",&t);
  float SI = (valueOne * valueTwo * t)/100;
  printf("Your SI is : %.2f", SI);
  break;
case 'C':
// Compound Interest
  printf("Enter Principle and Rate :\n");
  scanf("%f %f", &valueOne, & valueTwo);
  int time;
  printf("Enter Time:\n");
  scanf("%d",&time);
  float CI = valueOne * pow((1 + valueTwo / 100), time) - valueOne;
  printf("Your CI is : %.2f", CI);
  break;
case 'I':
// Definite Integration
  printf("Enter Value 1 and Value 2:\n");
  scanf("%f %f", &valueOne, & valueTwo);
  float j, sum = 0;
  int x, y;
  if (valueOne > valueTwo) {
    j = valueOne;
    valueOne = valueTwo;
    valueTwo = j;
  }
  for (j = valueOne; j < valueTwo; j += (valueTwo - valueOne) / N) {
    /* Define your function below, and include the suitable header files */
    y = x * x + 2 * x - 4;
    sum += y * (valueTwo - valueOne) / N;
  }
  printf("Value of integration is :%.3f", sum);
  break;
case '^':
// Power Value
  printf("Enter Number and Power value:\n");
  scanf("%f %f", &valueOne, & valueTwo);
  float answer5 = pow(valueOne, valueTwo);
  printf("Your Result of Power is : %.2f", answer5);
  break;
case '4':
  //  Square root calculation
  printf("Enter value: \n");
  scanf("%f", &valueOne);
  float valTwo = 0.5;
  float ans_square = pow(valueOne, valTwo);
  printf("Your Result of Square Root is : %.2f", ans_square);
  break;
case '5':
  // Cube root calculation
  printf("Enter value: \n");
  scanf("%f", &valueOne);
```

```
      float valOne = 0.3333;
      float ans_cube = pow(valueOne, valOne);
      printf("Your Result of Cube Root is : %.2f", ans_cube);
      break;
    case 's':
      // Sin value calculation
      printf("Enter the value of angle in degree :\n");
      scanf("%f",& valueOne);
      float degree1 = (3.14 / 180)*valueOne;
      float s = sin(degree1);
      printf("Sin value is %.2f", s);
      break;
    case 'c':
      // Cos value calculation
      printf("Enter the value of angle in degree :\n");
      scanf("%f",& valueOne);
      float degree2 = (3.14 / 180)*valueOne;
      float c = cos(degree2);
      printf("Cos value is %.2f", c);
      break;
    case 't':
      // Tan Value calculation
      printf("Enter the value of angle in degree :\n");
      scanf("%f",& valueOne);
      float degree3 = (3.14 / 180)*valueOne;
      float tanval = tan(degree3);
      printf("Tan value is %.2f", tanval);
      break;
    case 'l':
      // log value calculation of base e
      printf("Enter value: \n");
      scanf("%f", &valueOne);
      float ans_log = log(valueOne);
      printf("Your Result of Cube Root is : %.2f", ans_log);
      break;
    case 'p':
      // Permutation calculation
      printf("Enter Value 1 and Value 2 : \n");
      scanf("%f %f", &valueOne, &valueTwo);
      float npr=fact(valueOne)/fact(valueOne-valueTwo);
      printf("Result of Permutation is %.1f: ", npr);
      break;
    case 'b':
      // Permutation calculation
      printf("Enter Value 1 and Value 2 : \n");
      scanf("%f %f", &valueOne, &valueTwo);
      float ncr=fact(valueOne)/(fact(valueTwo)*fact(valueOne-valueTwo));
      printf("Result of Permutation is %.1f: ", ncr);
      break;
    case '2':
      //  Square calculation
      printf("Enter value: \n");
      scanf("%f", &valueOne);
      float valtwo = 2;
```

```
        float ans_square2 = pow(valueOne, valtwo);
        printf("Your Result of Square Root is : %.2f", ans_square2);
        break;
      case '3':
        //  Cube calculation
        printf("Enter value: \n");
        scanf("%f", &valueOne);
        float valthree = 3;
        float ans_square3 = pow(valueOne, valthree);
        printf("Your Result of Square Root is : %.2f", ans_square3);
        break;
      case 'i':
        //  Square root calculation
        printf("Enter value: \n");
        scanf("%f", &valueOne);
        float ans_inverse = 1 / valueOne;
        printf("Your Result of Square Root is : %.2f", ans_inverse);
        break;
      default:
        printf("Fail");
    }
}
int fact(int valueOne)
{
    int i,f=1;
    for(i=1;i<=valueOne;i++)
    {
        f=f*i;
    }
    return f;
}
```

APPLICATIONS

In most countries, students use calculators for schoolwork. There was some initial resistance to the idea out of fear that basic arithmetic skills would suffer. There remains disagreement about the importance of the ability to perform calculations "in the head", with some curricula restricting calculator use until a certain level of proficiency has been obtained, while others concentrate more on teaching estimation techniques and problem-solving. Research suggests that inadequate guidance in the use of calculating tools can restrict the kind of mathematical thinking that students engage in. Others have argued that calculator use can even cause core mathematical skills to atrophy, or that such use can prevent understanding of advanced algebraic concepts.

There are other concerns - for example, that a pupil could use the calculator in the wrong fashion but believe the answer because that was the result given. Teachers try to combat this by encouraging the student to make an estimate of the result manually and ensuring it roughly agrees with the calculated result. Also, it is possible for a child to type in −1 × −1 and obtain the correct answer '1' without realizing the principle involved. In this sense, the calculator becomes a crutch rather than a learning tool, and it can slow down students in exam conditions as they check even the most trivial result on a calculator.

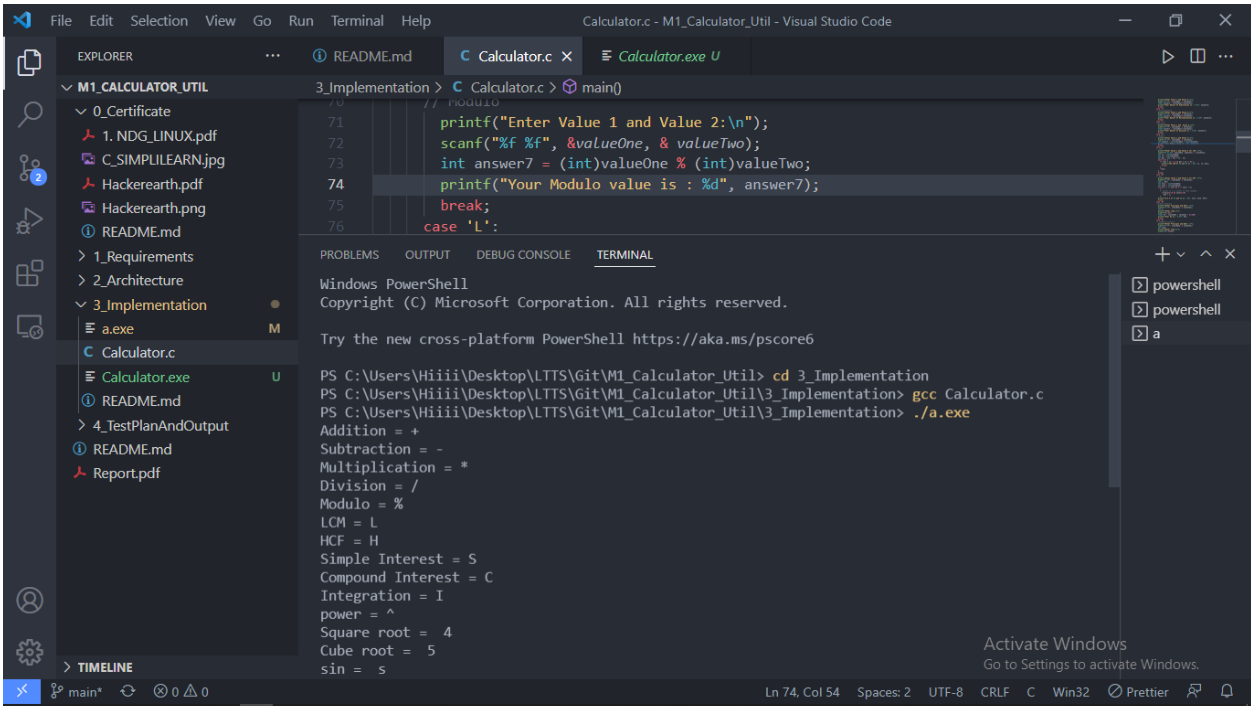
FUTURE SCOPE OF THE PROJECT

Our project will be able to implement in future after making some changes and modifications as we make our project at a very low level. So the modifications that can be done in our project are:

To make it screen touch so no need to touch key buttons and one more change which can we made is to add snaps of the person who use it

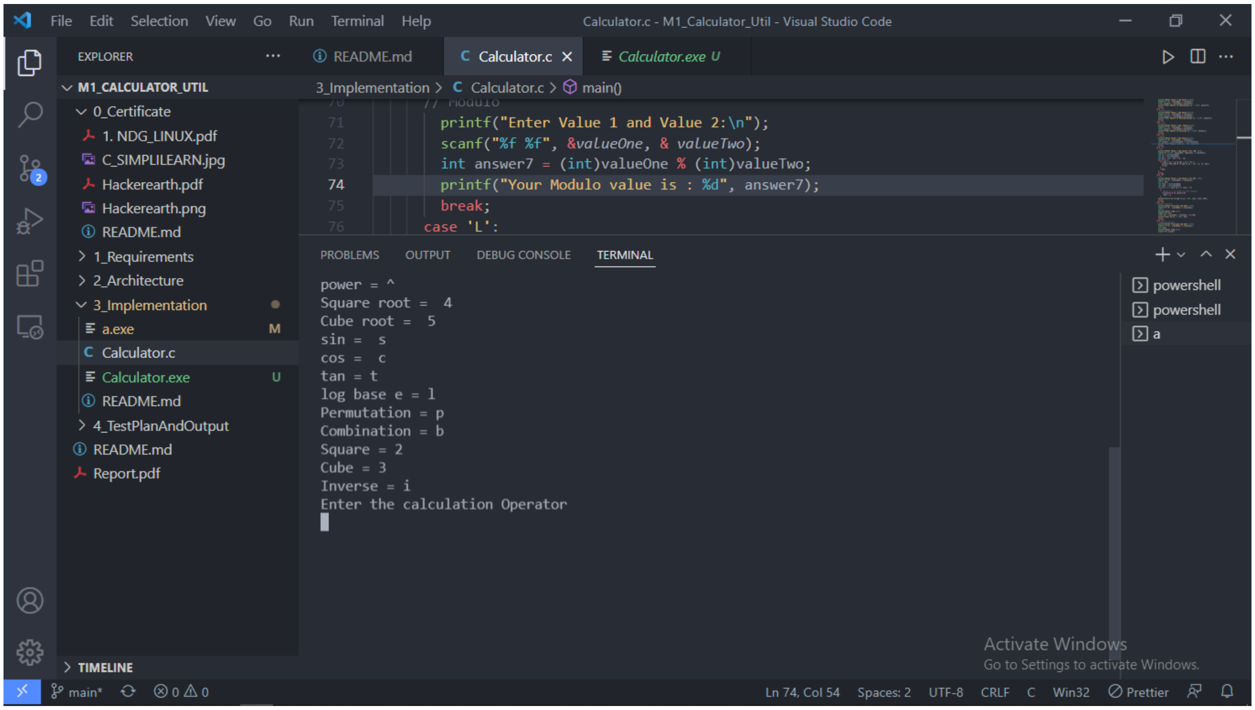
TESTING

Testing is the major control measure used during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only, has to uncover errors introduced during coding, but also errors introduced during previous phase. Thus the goal of testing is to uncover the requirements, design and coding errors in the programs. The Source code declared above for the program of Scientific Calculator has been tested and it has been found that the above source code is okay and correct. The program involves much type of conversions. These conversions have to done carefully.



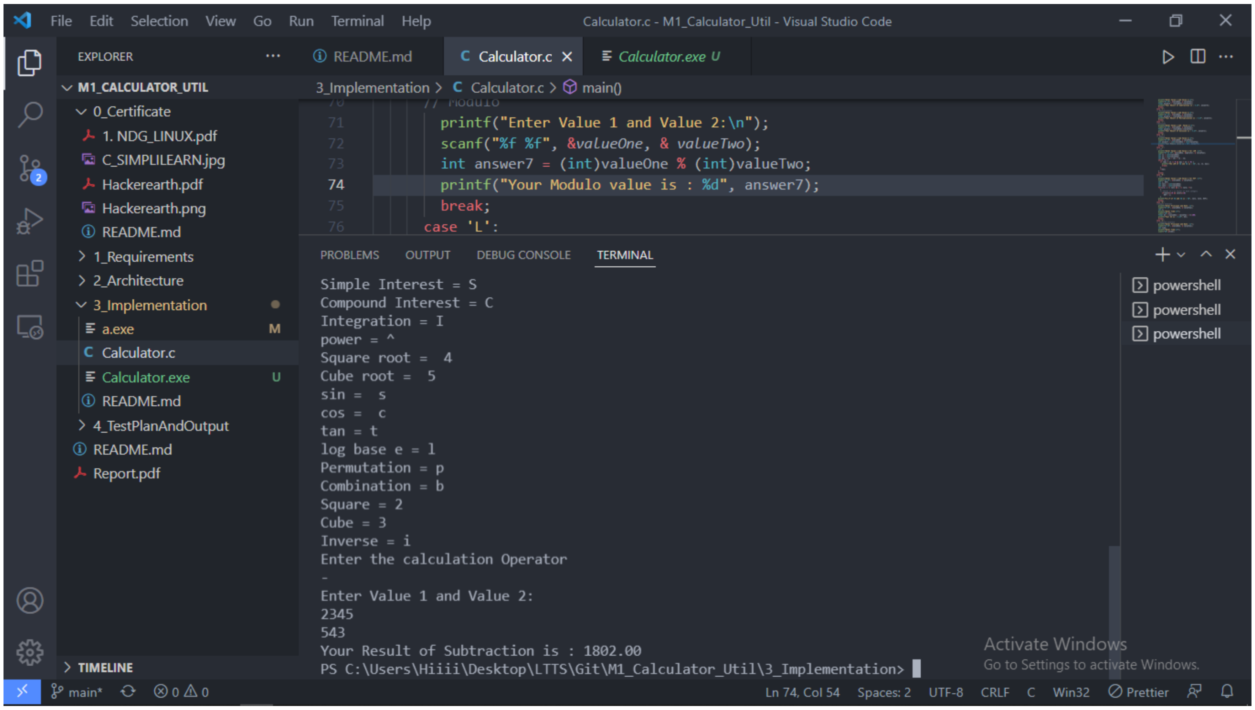
```
            // Modulo
            printf("Enter Value 1 and Value 2:\n");
            scanf("%f %f", &valueOne, & valueTwo);
            int answer7 = (int)valueOne % (int)valueTwo;
            printf("Your Modulo value is : %d", answer7);
            break;
          case 'L':
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util> cd 3_Implementation
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation> gcc Calculator.c
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation> ./a.exe
Addition = +
Subtraction = -
Multiplication = *
Division = /
Modulo = %
LCM = L
HCF = H
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
```



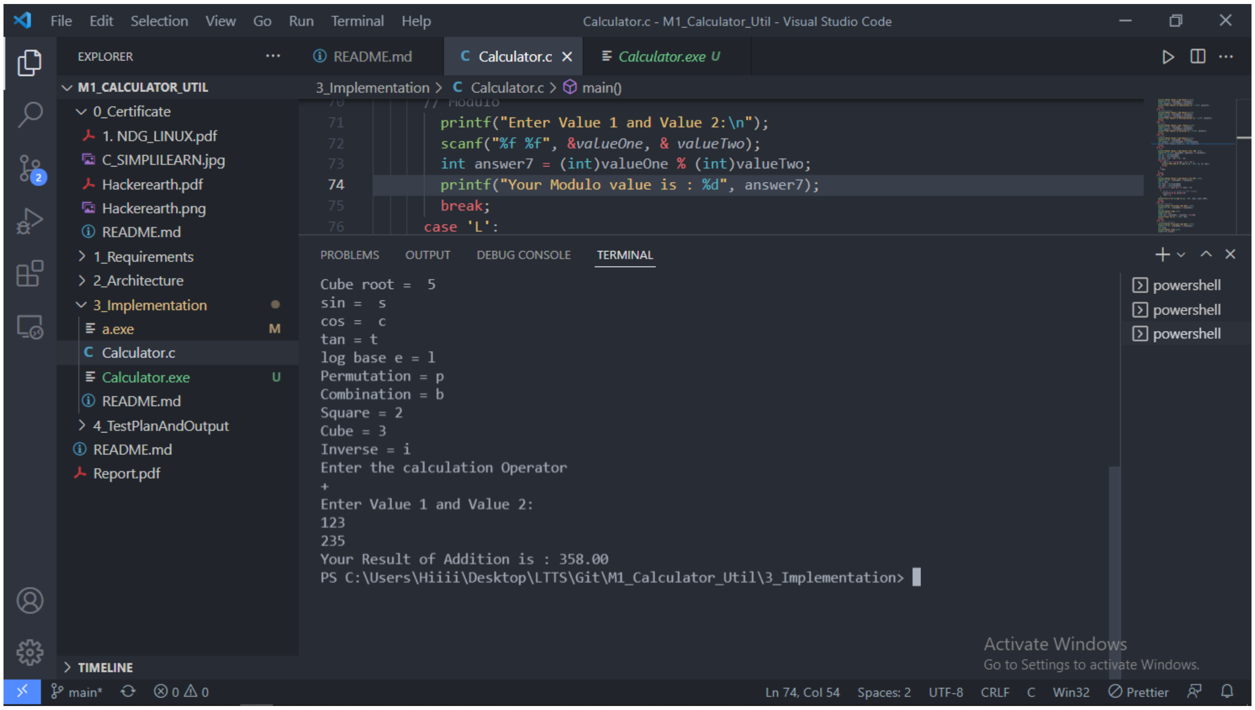
```c
      // Modulo
        printf("Enter Value 1 and Value 2:\n");
        scanf("%f %f", &valueOne, & valueTwo);
        int answer7 = (int)valueOne % (int)valueTwo;
        printf("Your Modulo value is : %d", answer7);
        break;
      case 'L':
```

```
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
█
```



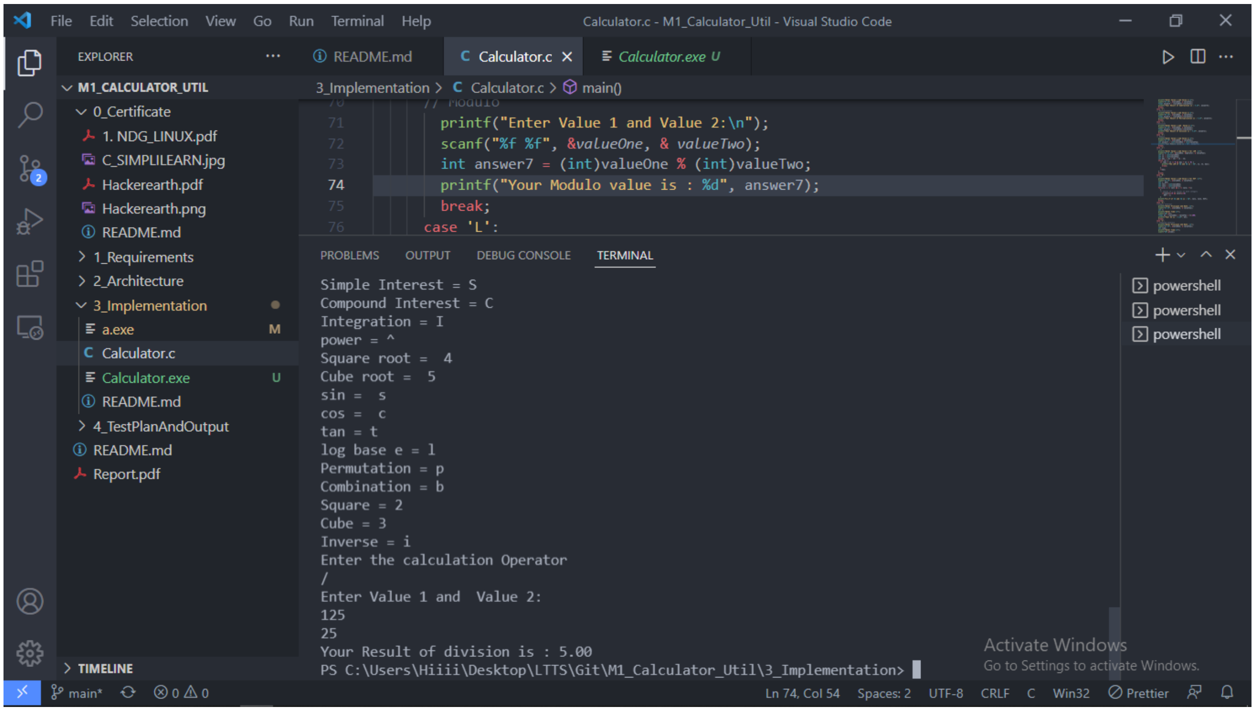
```c
      // Modulo
        printf("Enter Value 1 and Value 2:\n");
        scanf("%f %f", &valueOne, & valueTwo);
        int answer7 = (int)valueOne % (int)valueTwo;
        printf("Your Modulo value is : %d", answer7);
        break;
      case 'L':
```

```
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
-
Enter Value 1 and Value 2:
2345
543
Your Result of Subtraction is : 1802.00
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation> █
```



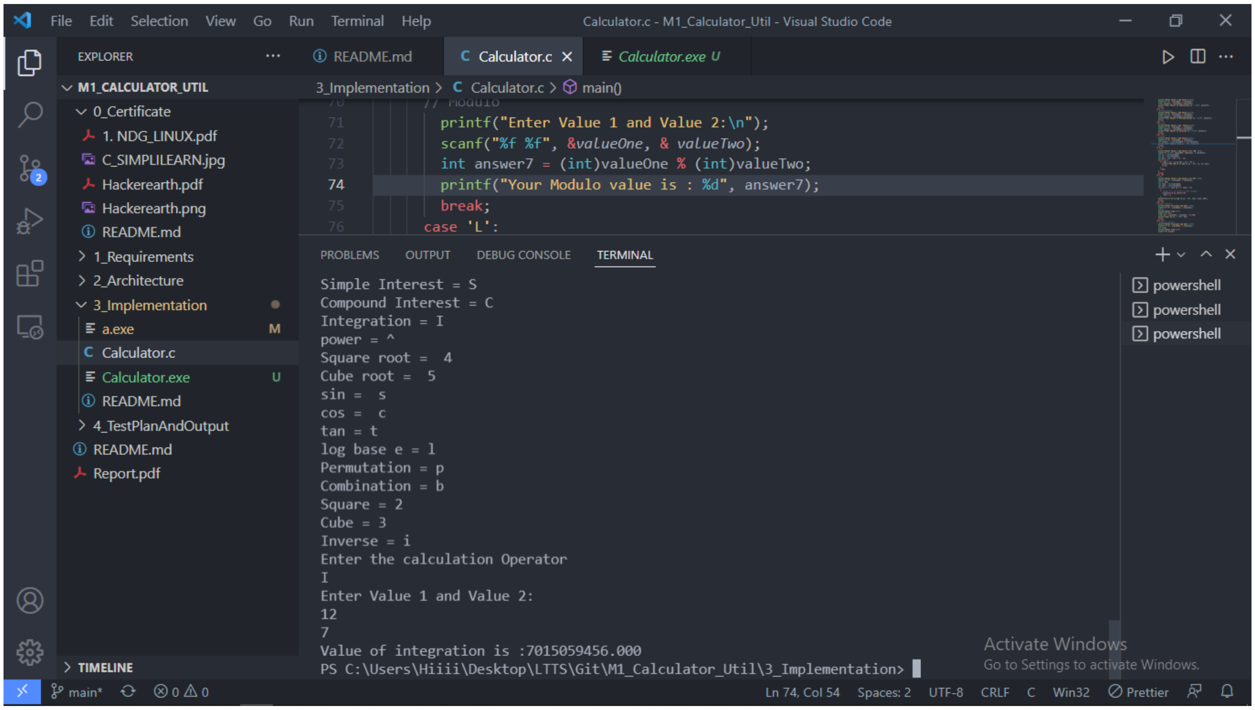
```c
      // Modulo
        printf("Enter Value 1 and Value 2:\n");
        scanf("%f %f", &valueOne, & valueTwo);
        int answer7 = (int)valueOne % (int)valueTwo;
        printf("Your Modulo value is : %d", answer7);
        break;
      case 'L':
```

```
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
+
Enter Value 1 and Value 2:
123
235
Your Result of Addition is : 358.00
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```



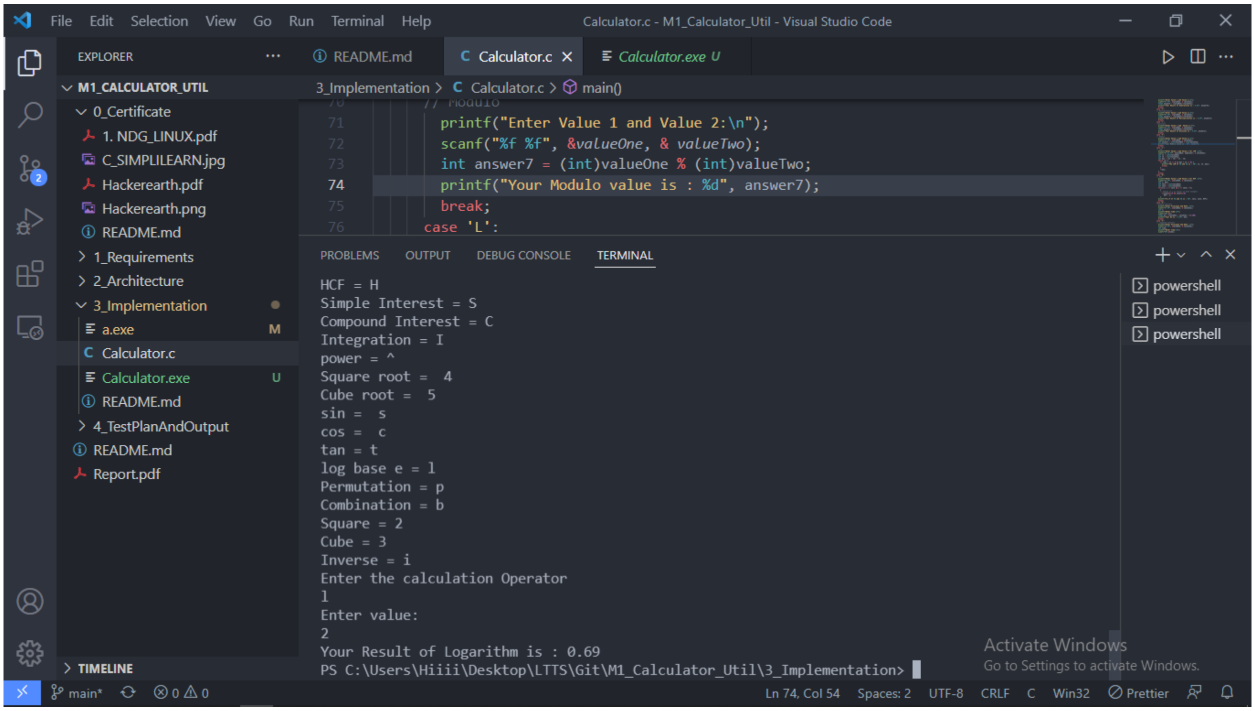
```c
        // Modulo
          printf("Enter Value 1 and Value 2:\n");
          scanf("%f %f", &valueOne, & valueTwo);
          int answer7 = (int)valueOne % (int)valueTwo;
          printf("Your Modulo value is : %d", answer7);
          break;
        case 'L':
```

```
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
/
Enter Value 1 and  Value 2:
125
25
Your Result of division is : 5.00
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```



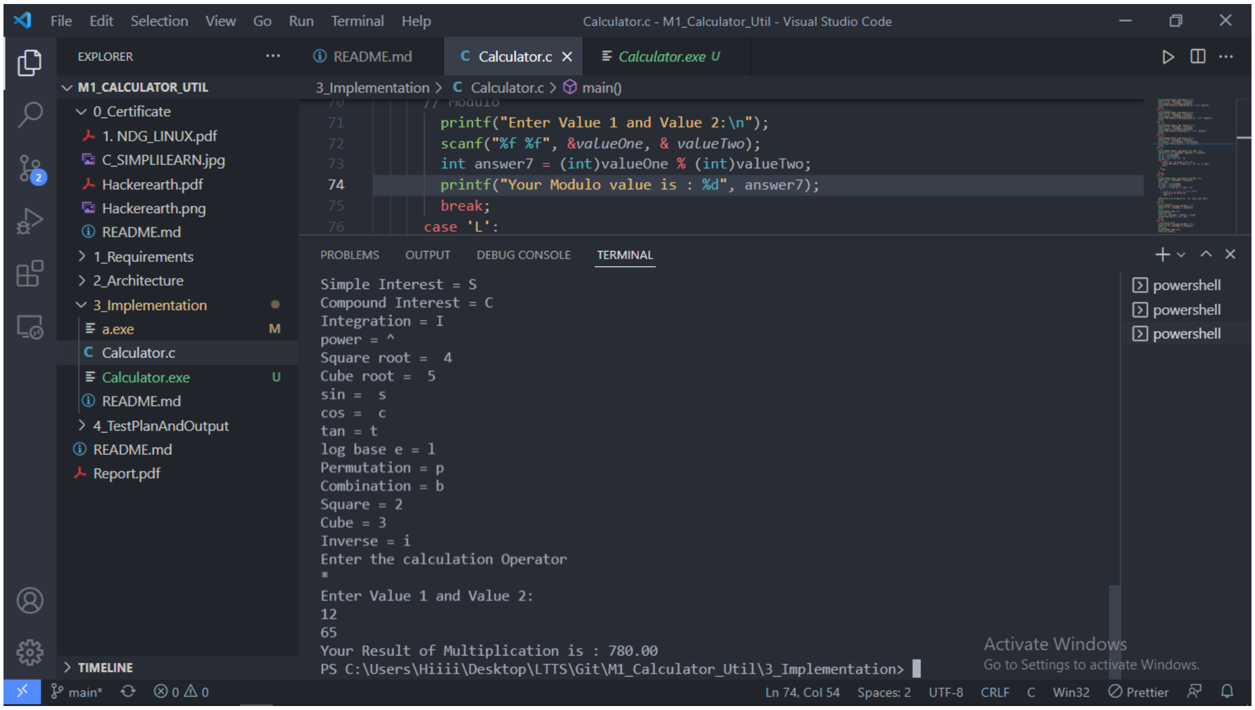
```c
        // Modulo
          printf("Enter Value 1 and Value 2:\n");
          scanf("%f %f", &valueOne, & valueTwo);
          int answer7 = (int)valueOne % (int)valueTwo;
          printf("Your Modulo value is : %d", answer7);
          break;
        case 'L':
```

```
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
I
Enter Value 1 and Value 2:
12
7
Value of integration is :7015059456.000
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```



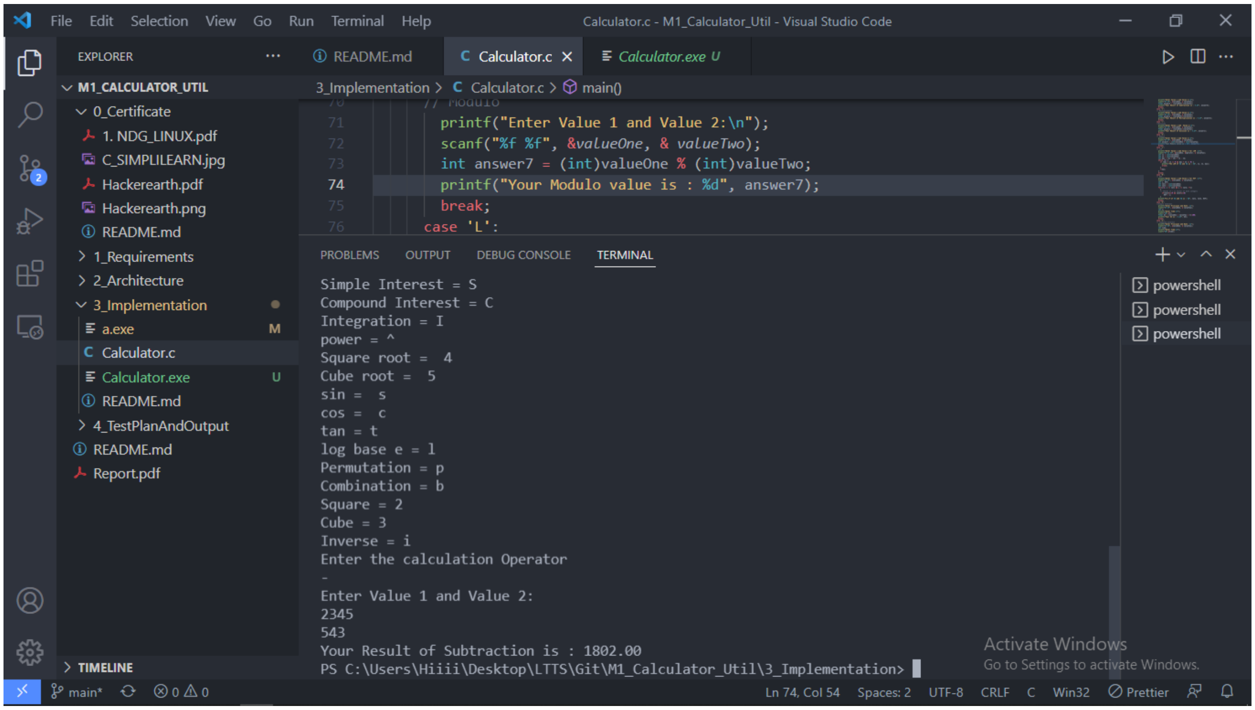
```c
    // Modulo
      printf("Enter Value 1 and Value 2:\n");
      scanf("%f %f", &valueOne, & valueTwo);
      int answer7 = (int)valueOne % (int)valueTwo;
      printf("Your Modulo value is : %d", answer7);
      break;
    case 'L':
```

```
HCF = H
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
l
Enter value:
2
Your Result of Logarithm is : 0.69
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```



```c
    // Modulo
      printf("Enter Value 1 and Value 2:\n");
      scanf("%f %f", &valueOne, & valueTwo);
      int answer7 = (int)valueOne % (int)valueTwo;
      printf("Your Modulo value is : %d", answer7);
      break;
    case 'L':
```

```
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
*
Enter Value 1 and Value 2:
12
65
Your Result of Multiplication is : 780.00
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```



```
        // Modulo
          printf("Enter Value 1 and Value 2:\n");
          scanf("%f %f", &valueOne, & valueTwo);
          int answer7 = (int)valueOne % (int)valueTwo;
          printf("Your Modulo value is : %d", answer7);
          break;
        case 'L':
```

```
Simple Interest = S
Compound Interest = C
Integration = I
power = ^
Square root =  4
Cube root =  5
sin =  s
cos =  c
tan = t
log base e = l
Permutation = p
Combination = b
Square = 2
Cube = 3
Inverse = i
Enter the calculation Operator
-
Enter Value 1 and Value 2:
2345
543
Your Result of Subtraction is : 1802.00
PS C:\Users\Hiiii\Desktop\LTTS\Git\M1_Calculator_Util\3_Implementation>
```

REFERENCES

1. https://bigprof.com/c4engineering/intgr001.htm#:~:text=This%20C%20program%20involves%20dividing,is%20the%20value%20of%20integration.

2. https://codeforwin.org/2015/06/c-program-to-find-hcf-of-two-numbers.html

3. https://en.wikipedia.org/wiki/Integration

4. https://app.diagrams.net/

5. https://www.google.com/search?q=how+to+enter+in+directory+through+cmd&rlz=1C1SQJL_enIN962IN962&oq=how+to+enter+in+directory+through+cmd&aqs=chrome..69i57.10025j0j7&sourceid=chrome&ie=UTF-8

6. https://www.google.com/search?q=doxygenated+auto+documentation&rlz=1C1SQJL_enIN962IN962&oq=doxygenated+auto+documentation&aqs=chrome..69i57.10553j0j7&sourceid=chrome&ie=UTF-8

7. https://www.netdevgroup.com/online/courses/open-source/linux-essentials

8. https://www.sololearn.com/learning/1089

9. https://www.hackerearth.com/@vikkuvikash79097

10. https://git-scm.com/