CALCULATOR
PROJECT REPORT

Description

Simple calculator is a project to allow users to calculate basic four operations in mathematics such as addition, subtraction, multiplication, division but in this simple project, i also included three additional operations such as reminder, power and modulus.

The input includes the command such that we can give numbers based on the operation that we want to perform. For example the operation we needed is addition, we have to enter 1 as a command and that particular operation works.

Requirements

The switch-case statement is used to write a simple calculator program in C language. The remainder operator "%" is used with data values. To print the remainder operator "%%" is used in the first "printf()" function. The variable "ch" store the operator, similarly the variables num1 and num2 stores the two numbers. The switch has condition "ch" so the entered operator is matched with the case label in which the case label statements are executed and display result on the screen. If the entered operator does not matched with the case labels then the default statement will be executed and it will display the message "Error! Invalid Operator." to the screen.

High Level Requirements

- User should able to view operation list.
- User should able to select the operation.
- The system should perform the given operation.
- The system should display the correct result.

Low Level Requirements

Hardware – User needs a system with keyboard.

Software – Windows or Linux OS, any C compiler (eg. Vs code) or Linux terminal(wsl).

SWOT Analysis

1.Strength

It is innovative, user friendly and long lasting.

2.Weakness

Does not include all the operations rather have only addition, subtraction, multiplication, division, reminder, power and modulus.

3.Threats

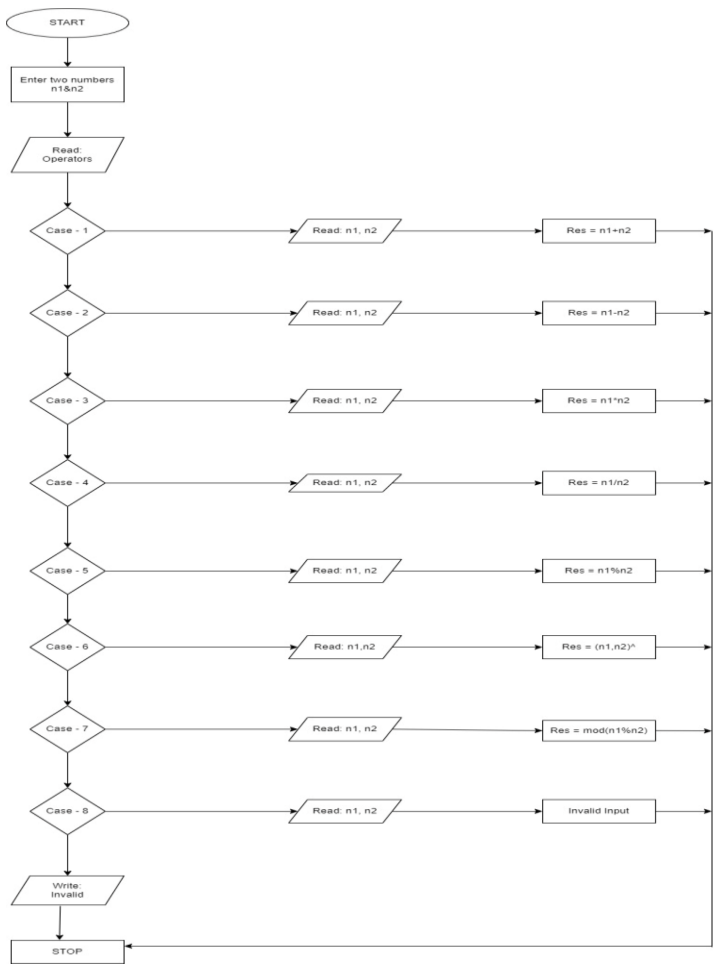
The slower growth in the technological innovation will also bring a significant threat in the upcoming modern world.

Architecture

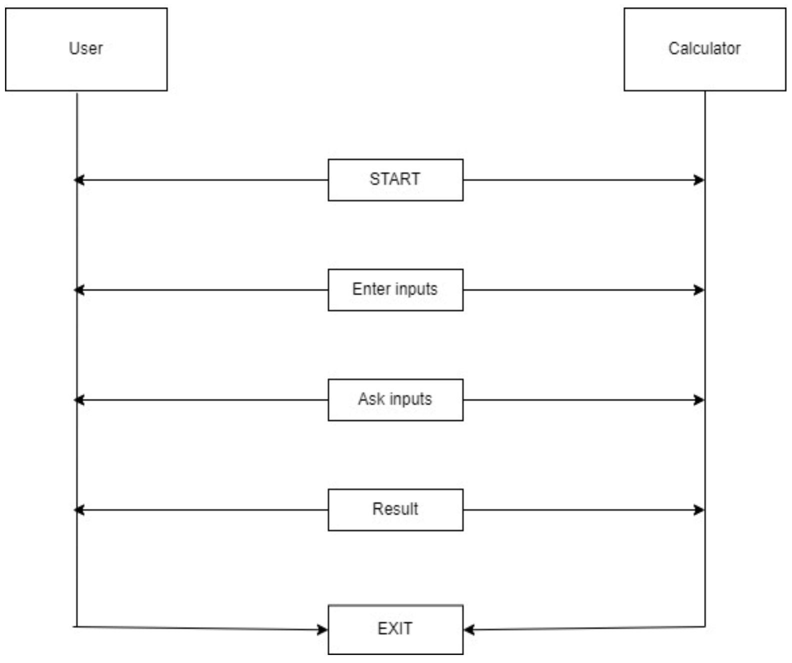
Architecture is a diagrammatic represtation which shows how the operations and function works in the program.

1.Behaviour Diagram

1.1 Flow Chart

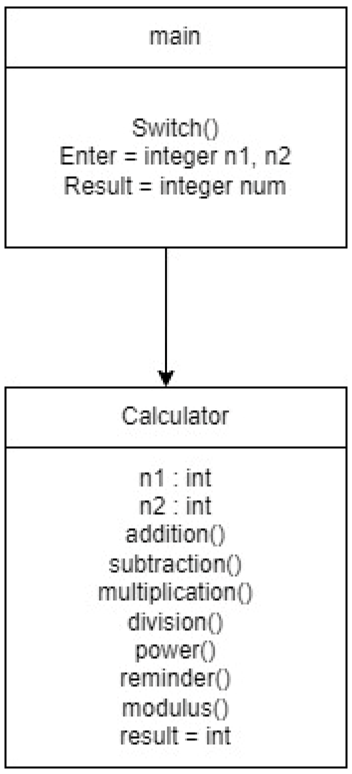


1.2 Sequence Diagram

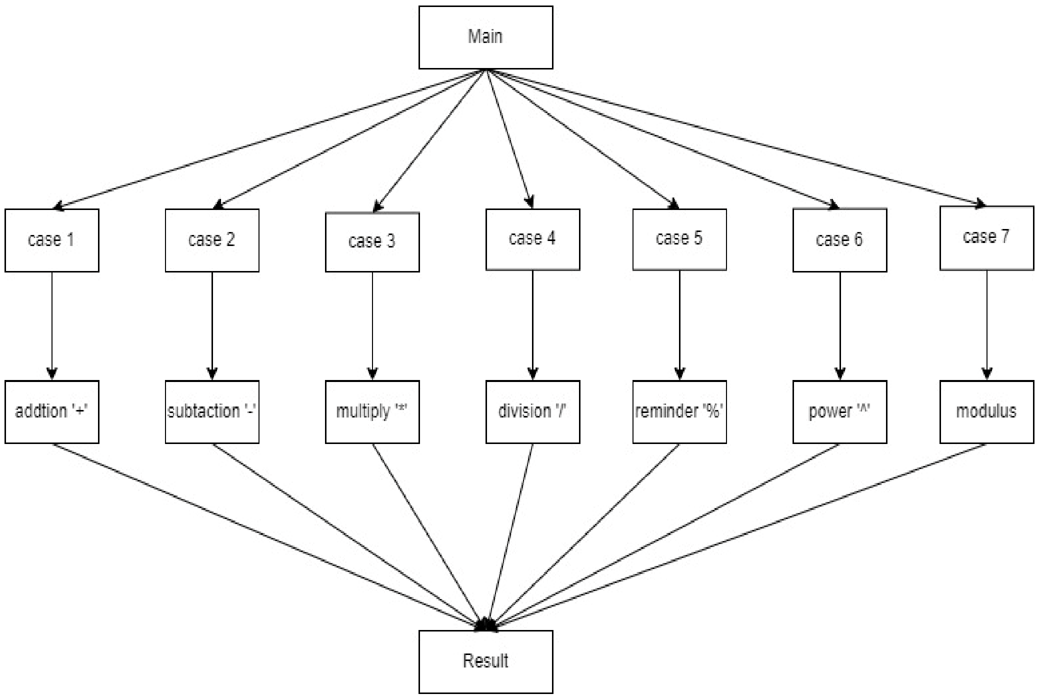


2.Structural Diagram

2.1 Class Diagram



2.2 Component Diagram



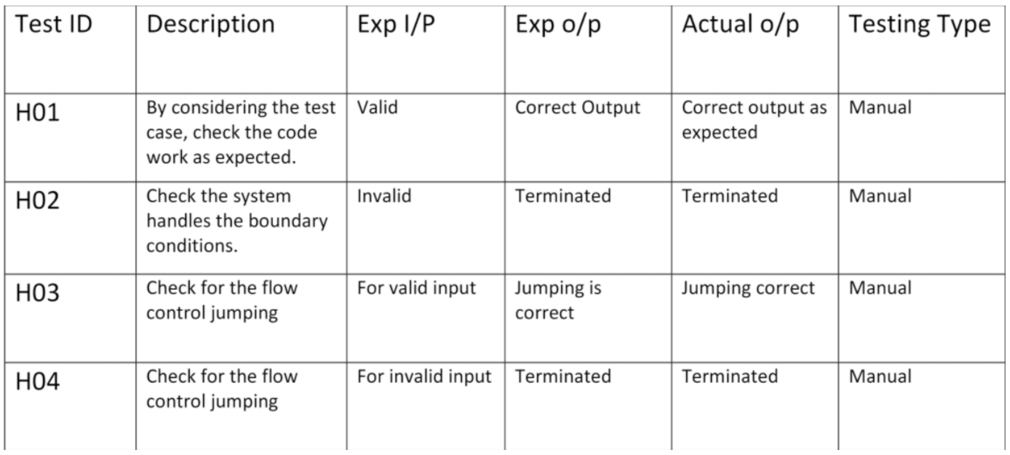
Implementation

The implementation includes the main code and its functional codes. These functional files shows how the every functions of the code works. It includes the functions such as addition, subtraction, multiplication, division, reminder, power and modulus.

Test Plan and Output

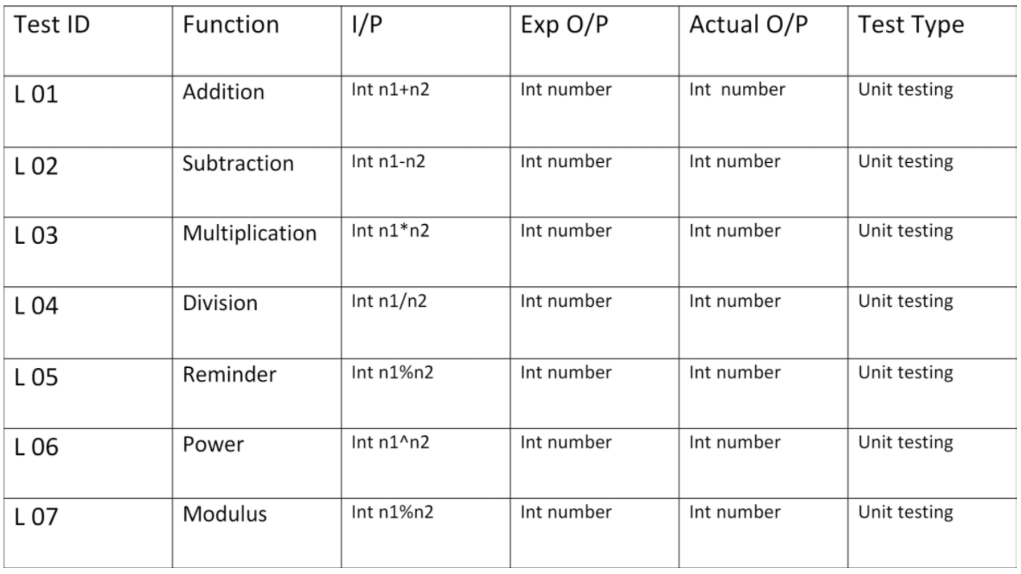
It includes the High Level and Low Level requirement of the program for implementation and the code to run.

1.High Level Requirement



Test ID	Description	Exp I/P	Exp o/p	Actual o/p	Testing Type
H01	By considering the test case, check the code work as expected.	Valid	Correct Output	Correct output as expected	Manual
H02	Check the system handles the boundary conditions.	Invalid	Terminated	Terminated	Manual
H03	Check for the flow control jumping	For valid input	Jumping is correct	Jumping correct	Manual
H04	Check for the flow control jumping	For invalid input	Terminated	Terminated	Manual

2. Low Level Requirement



Test ID	Function	I/P	Exp O/P	Actual O/P	Test Type
L 01	Addition	Int n1+n2	Int number	Int number	Unit testing
L 02	Subtraction	Int n1-n2	Int number	Int number	Unit testing
L 03	Multiplication	Int n1*n2	Int number	Int number	Unit testing
L 04	Division	Int n1/n2	Int number	Int number	Unit testing
L 05	Reminder	Int n1%n2	Int number	Int number	Unit testing
L 06	Power	Int n1^n2	Int number	Int number	Unit testing
L 07	Modulus	Int n1%n2	Int number	Int number	Unit testing

Main Code

```
#include<stdio.h>
#include<stdlib.h>

//function declarations

void display(float n1, float n2, char ch, float result);
void add(float n1, float n2);
void subtract(float n1, float n2);
void multiply(float n1, float n2);
void divide(float n1, float n2);
void rem(float n1, float n2);
void power(float n1, float n2);
void modulus(float n1, float n2);

//main function

int main()
{
```

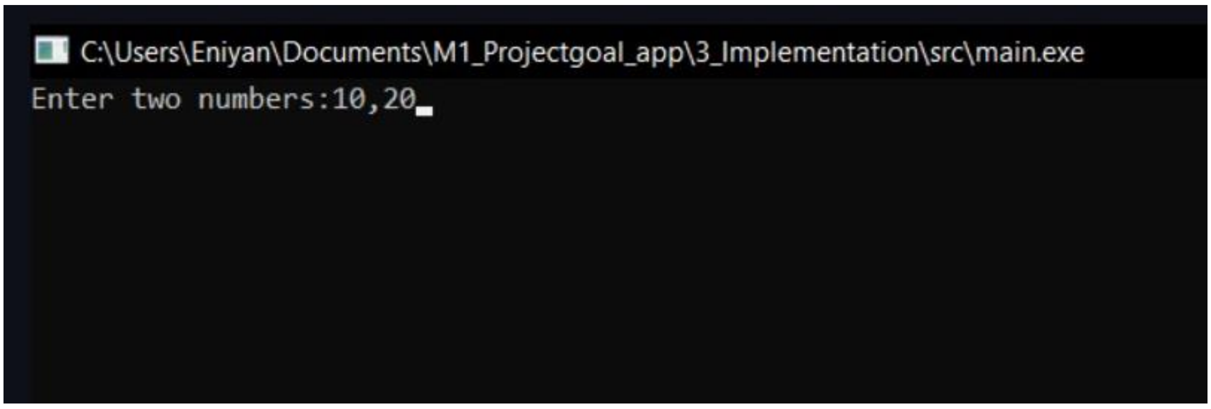
```
float n1, n2;
int ch;

do{
printf("Enter two numbers:");
scanf("%f %f", &n1, &n2);
printf("\n....");
printf("\n1.Addition");
printf("\n2.Subtraction");
printf("\n3.Multiplication");
printf("\n4.Division");
printf("\n5.Remainder");
printf("\n6.Power (x^y)");
printf("\n7.Modulus");
printf("\n8.Close");
printf("\nEnter your option:");
scanf("%d", &ch);

//switch function

switch (ch)
{
case 1:
add(n1,n2);
break;
case 2:
subtract(n1,n2);
break;
case 3:
multiply(n1,n2);
break;
case 4:
divide(n1,n2);
break;
case 5:
rem(n1,n2);
break;
case 6:
power(n1,n2);
break;
case 7:
modulus(n1,n2);
break;
case 8:
printf("Closed");
exit(0);
default:
printf("Invalid input.");
printf("Please enter correct input.");
}
printf("\n....\n");
}
while(1);
return 0;
}

//function for displaying the result

void display(float n1, float n2, char ch, float result)
{
```

```
printf("%.2f %c %.2f = %.2f\n", n1, ch, n2, result);
}

//function for addition of two numbers

void add(float n1, float n2)
{
float result = n1+n2;
display(n1, n2, '+', result);
}

//function for subtraction of two numbers

void subtract(float n1, float n2)
{
float result = n1-n2;
display(n1, n2, '-', result);
}

//function for multiplication of two numbers

void multiply(float n1, float n2)
{
float result = n1*n2;
display(n1, n2, '*', result);
}

//function for division of two numbers

void divide(float n1, float n2)
```

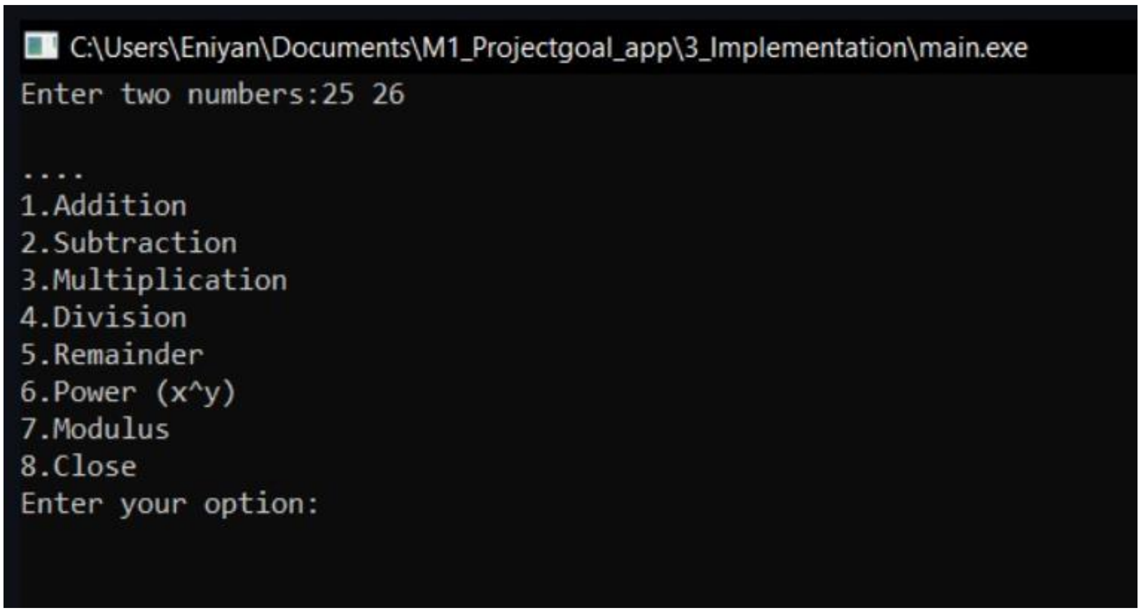
```
{
float result = n1/n2;
display(n1, n2, '/', result);
}

//function for calculating remainder

void rem(float n1, float n2)
{

//modulus operator only works on int data type
//floating numbers are converted to int number

int num1 = n1;
int num2 = n2;
int result = num1%num2;
printf("%d %% %d = %d\n", num1, num2, result);
}

//function for calculating power

void power(float n1, float n2)
{
if(n2<0)
printf("Second number should be positive");
else
{
float result=1.0;
for(int i=1; i<=n2;i++)
{
```

```
result *= n1;
}
display(n1, n2, '^', result);
}
}

//function for modulus

void modulus(float n1, float n2)
{
//modulus operator only works on int data type
//floating numbers are converted to int number

int num1 = n1;
int num2 = n2;
int result = num1 % num2;
printf("The modulus of %d and %d is %d",num1,num2, result);

}
```

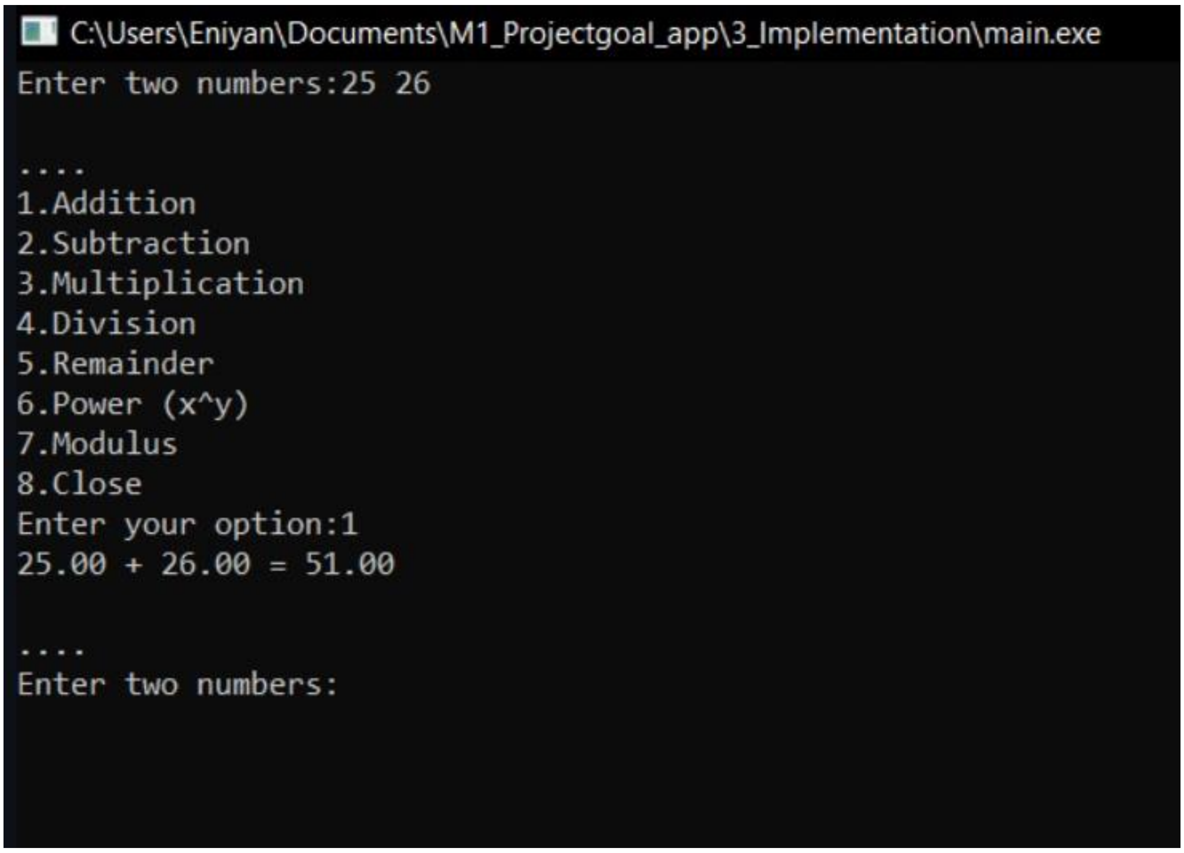
Output



```
C:\Users\Eniyan\Documents\M1_Projectgoal_app\3_Implementation\src\main.exe
Enter two numbers:10,20
```

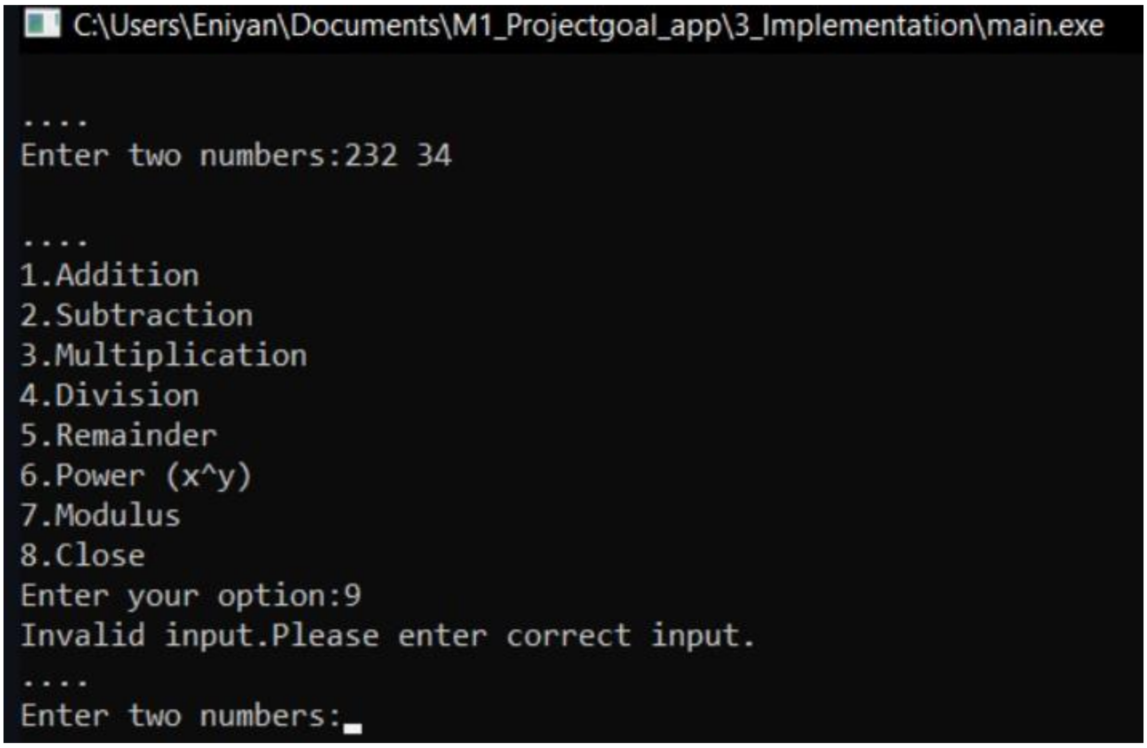


```
C:\Users\Eniyan\Documents\M1_Projectgoal_app\3_Implementation\main.exe
Enter two numbers:25 26

....
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Remainder
6.Power (x^y)
7.Modulus
8.Close
Enter your option:
```



```
C:\Users\Eniyan\Documents\M1_Projectgoal_app\3_Implementation\main.exe
Enter two numbers:25 26

....
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Remainder
6.Power (x^y)
7.Modulus
8.Close
Enter your option:1
25.00 + 26.00 = 51.00

....
Enter two numbers:
```

For Invalid Input



```
C:\Users\Eniyan\Documents\M1_Projectgoal_app\3_Implementation\main.exe

....
Enter two numbers:232 34

....
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Remainder
6.Power (x^y)
7.Modulus
8.Close
Enter your option:9
Invalid input.Please enter correct input.
....
Enter two numbers:_
```

Reference

For Diagrams - https://app.diagrams.net/?src=about