

MINI PROJECT REPORT ON C -PROGRAMMING

Module: C-Programming

Project title : Simple-Calculator

Submitted by

Sujata Naikar

https://github.com/sujatann/M1_Simple-Calculator_Utility.git

[PS:99007669]

AIM: To design a simple calculator using c programming.

CONTENTS

1.Objectives and abstract

2.Certificates

3.Requirements

4.Architecture

5.Implementation

6.Testplan

7.References

1.OBJECTIVES

- To make a user interactive program , a simple calculator.
- Two numbers are accepted to run the operation for addition, subtraction, multiplication and division, and only one number is asked as an input for the operations such as to find out squares and cubes.
- Exceptions have been specified in the code .

ABSTRACT

The project that I would be doing will mainly stress on performing simple mathematic calculations that are required while adding, subtracting, dividing, multiplying any two numbers and gives squares and cubes of any particular number that you ask for. This project consists of a program which takes an arithmetic operator +, -, *, / and two operands from the user. Then, it performs the calculation on the two operands depending upon the operator entered by the user.

SOFTWARES REQUIRED

- Visual studio code, github, ubuntu-wsl.

There is a folder as milestone-1, which in short gives idea about what all is to be covered in carrying out the project.

❖ Milestone-1

- Folder structure.
- Idea
- Requirements
- Design
- Test plan

❖ ISSUE THAT WAS CREATED AT THE STARTING OF ANALYSING THE PROBLEM STATEMENT.

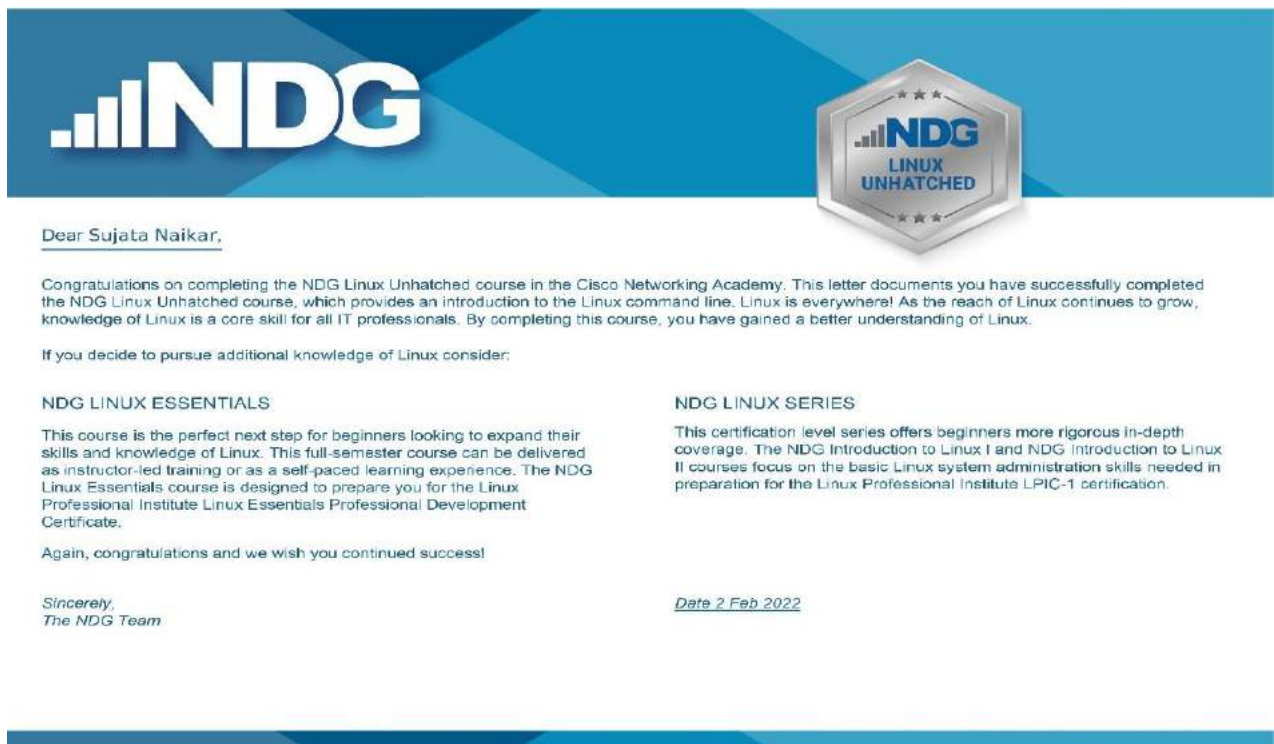
- 1.Capture a well processed idea.
- 2.Requirements for the project.
 - low level and high level
- 3.Design(behavioural and structural)
- 4.Test plan and Validation.

2. CERTIFICATES INCLUDE:

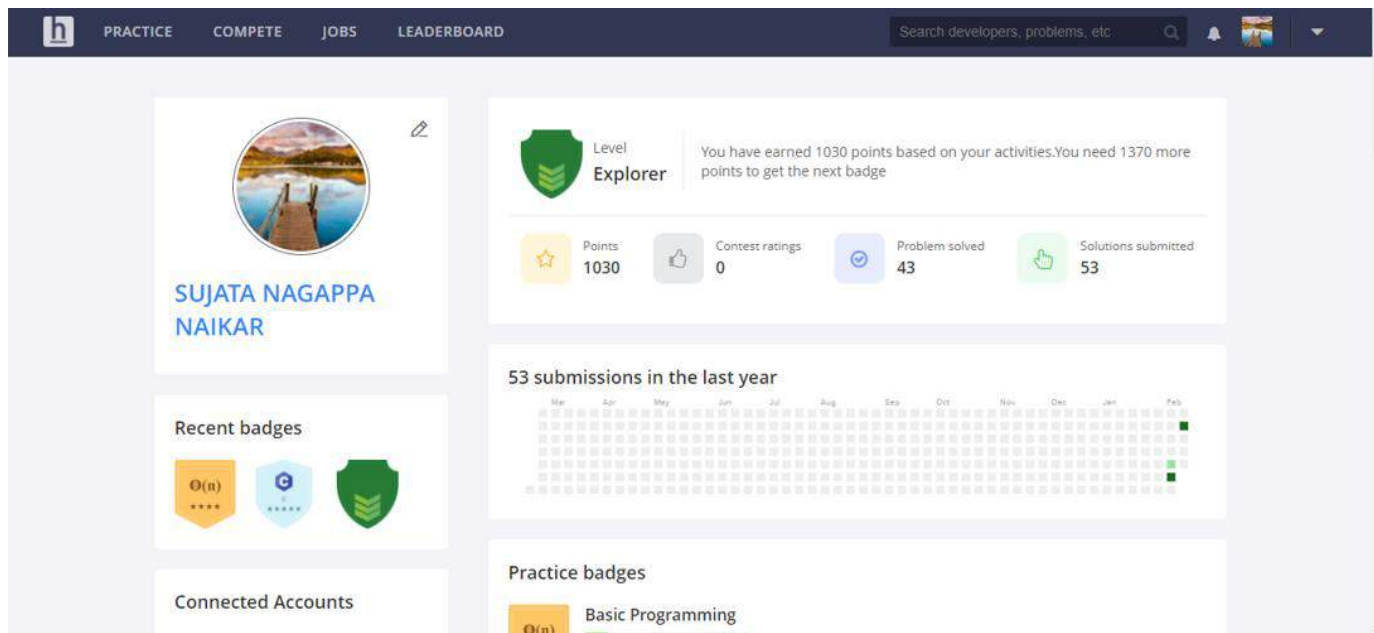
2.1 Sololearn.



2.2 Cisco NDG Linux.



2.3 Screenshot of HackerEarth points.



2.4 Screenshot of Github learning.

What went well

Now, you have your slide deck deployed [here](#) and you've gone through the GitHub Flow.

Before I say good-bye, here's a recap of all the tasks you've accomplished in your repository:

- You learned about issues, pull requests, and the structure of a GitHub repository
- You learned about branching
- You created a commit
- You viewed and responded to pull request reviews
- You edited an existing file
- You enabled GitHub Pages
- You made your first contribution! 🎉

Give us feedback

We'd love to hear what you thought of this course. Please share your feedback in [this brief survey](#).

What's next?

We hope this course helped you feel more comfortable using GitHub. Remember: You're always welcome to repeat course topics if you'd like additional practice. If you're ready to build on your new skills, here are some ideas.

Contribute to a project

To find projects to contribute to through trending topics and curated collections, check out [GitHub Explore](#).

Keep learning

3.Requirements

High level Requirements

- Able to perform operations such as,
 1. Addition.
 2. Substraction.
 3. Multiplication.
 4. Division.
 5. Square.
 6. Cube.

Low level Requirements

1. Taking the two numbers to do any operation.
2. A code is written where different commands are used for different operations.
3. In case of squaring and cubing a number take only one number as an input.
4. The answer must be displayed on the screen.

❖ SWOT Analysis

1. Strength
 - Capable of adding, subtracting, dividing, multiplying any two numbers and gives squares and cubes of any particular number that you ask for.
2. Weakness
 - As the main intension is to work with two numbers provided more will not be supported.
3. Opportunities
 - In the basic step to make someone understand the simple operations of mathematics it is most useful one.
4. Threat
 - ❖ Will not ask for more than two numbers.

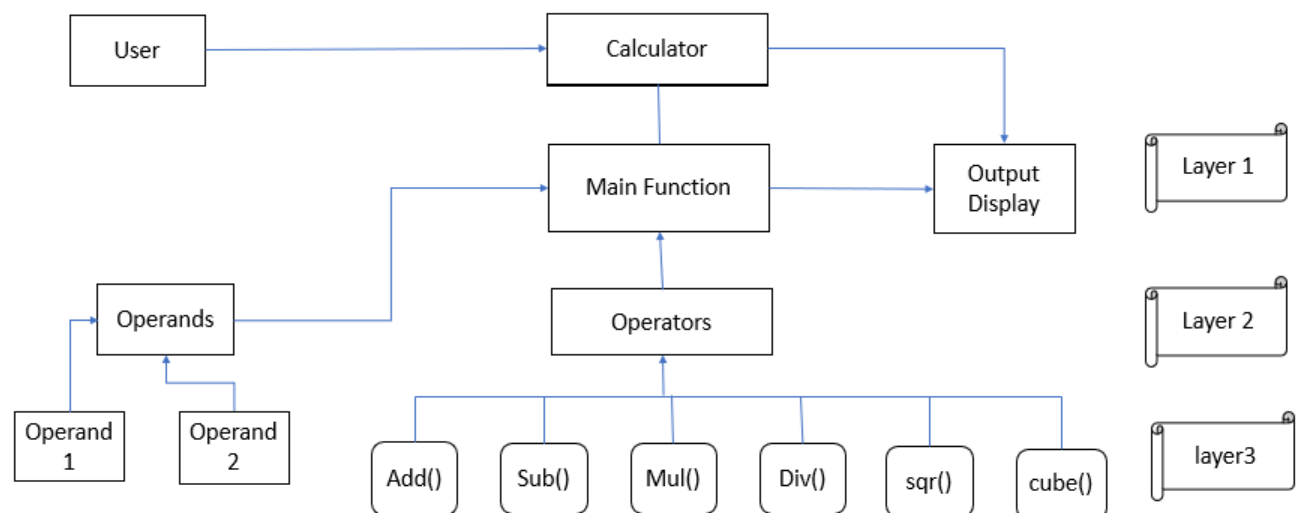
❖ 5W's and 1 H

- When to use is in need of the requirement of basic calculations.
- Where: can be in schools, working place, counters etc.
- Why can be stated in case to know the exact calculations.
- Who is someone who wants to take use of it and find solution for their problems.
- What is by using numbers in the calculator.
- H for How, is by taking two numbers as input and carrying out any operation of the choice.

4.Architecture.

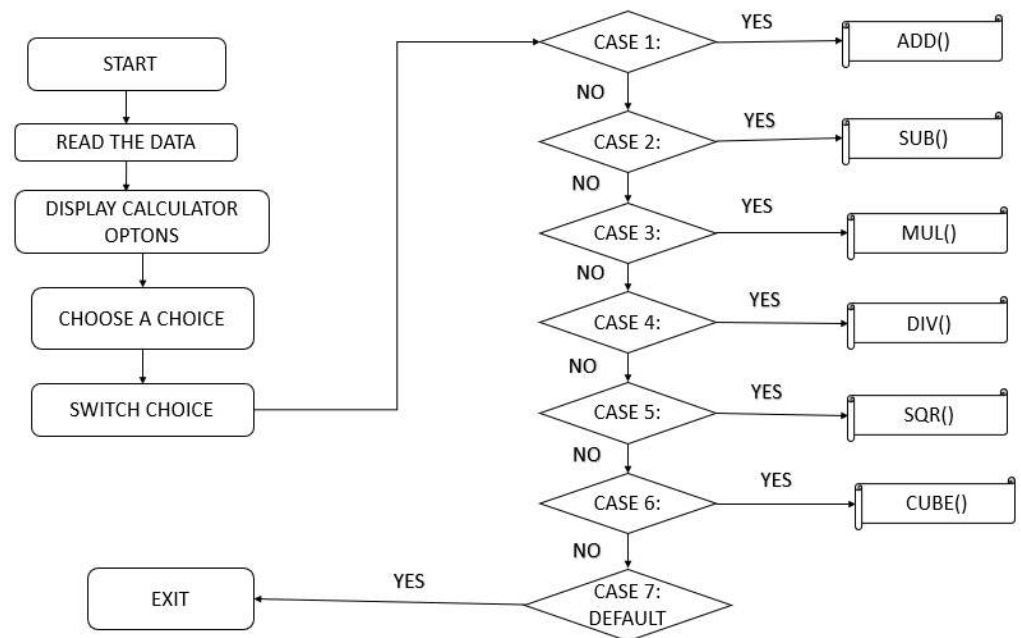
Design

4.1.Structural diagrams

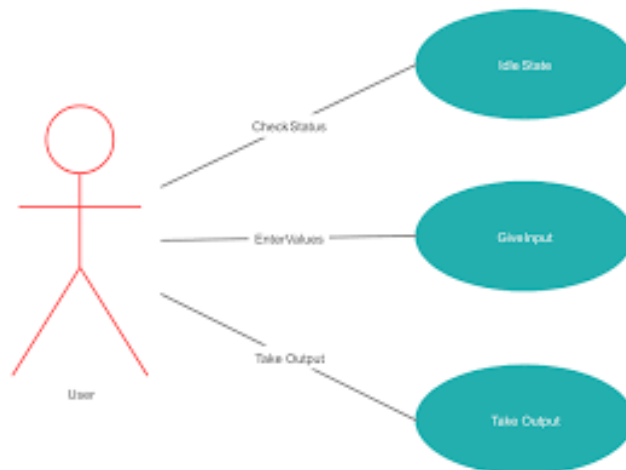


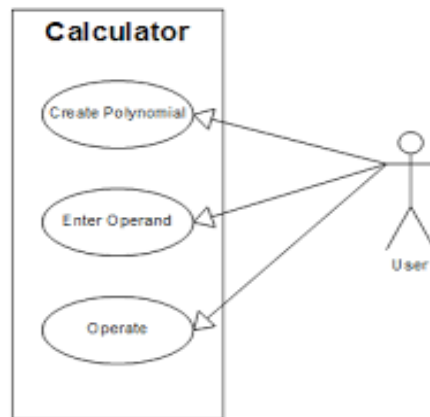
4.2 Behavioural diagrams

- Flowcharts



- Use case diagrams





MODELS:



5.IMPLEMENTATION.

CODE FOR SIMPLE CALCULATOR

```
//simple computer:add,sub,mul,div,sqr,cube
```

```
// program to obtain some basic operations used in simple calculators
```

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



```

#include<stdlib.h>

#include<math.h>

int main()
{
    int choose;

    long no1, no2, m;


    printf(" select any one from the  options  below:")

    "\n1 = Addition\n"
    "\n2 = Subtraction\n"
    "\n3 = Multiplication\n"
    "\n4 = Division\n"
    "\n5 = Squares\n"
    "\n6 = Cubes\n"
    "\n7 = exit\n"
    "\n\nchoose: ");


    scanf("%d", &choose);

    //while loop check whether the choose option  is in the given range
    while(choose< 1 || choose > 7)
    {
        printf("\nchoose the above mentioned option."
        "\nChoose: ");
        scanf("%d", &choose);
    }

```

switch (choose)

{

case 1:

printf(" Please enter any two numbers to add: \n");

scanf("%ld %ld", &no1, &no2);

m = no1 + no2;

printf("Sum = %ld", m);

break;

case 2:

printf("Please enter any two numbers to subtract: \n");

scanf("%ld %ld", &no1, &no2);

m = no1 - no2;

printf("Sub = %ld", m);

break;

case 3:

printf("Please enter any two numbers to multiply: \n");

scanf("%ld %ld", &no1, &no2);

m = no1 * no2;

printf("Pdt= %ld", m);

break;

case 4:

```
printf("Enter the Dividend: ");  
scanf("%ld", &no1);  
printf("Enter the Divisor: ");  
scanf("%ld", &no2);  
  
//while loop checks for the divisor whether it is zero or not
```

```
while(no2 == 0)  
{  
    printf("\nDivisor cannot be zero."  
    "\nEnter divisor once again to confirm: ");  
    scanf("%ld", &no2);  
}
```

```
m = no1 / no2;  
printf("\nQuotient = %ld", m);  
break;
```

case 5:

```
printf("Please enter any number of your choice: \n");  
scanf("%ld", &no1);  
m = no1 * no1;  
printf("Sqr = %ld", m);  
break;
```

case 6:

```
printf("Please enter any number of your choice: \n");
```

```
scanf("%ld", &no1);
```

```
m = no1 * no1 * no1;
```

```
printf("Cube = %ld", m);
```

```
break;
```

case 7:

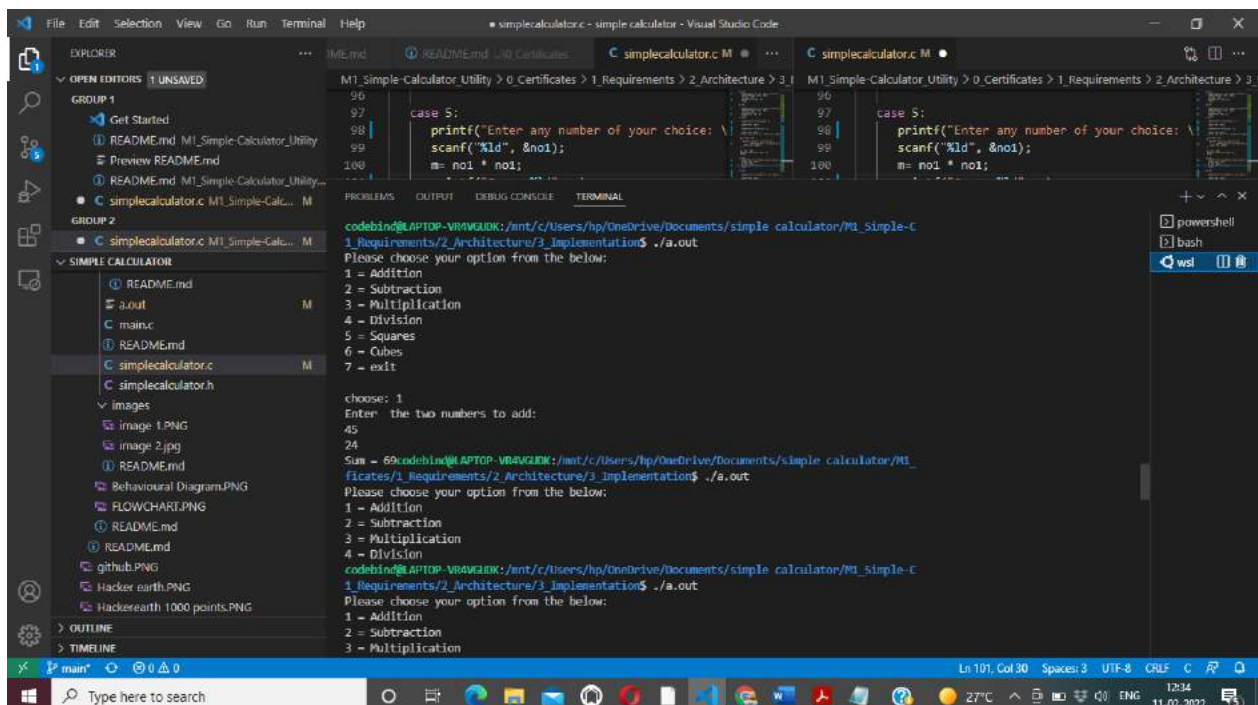
```
return 0;
```

```
default: printf("\nError");
```

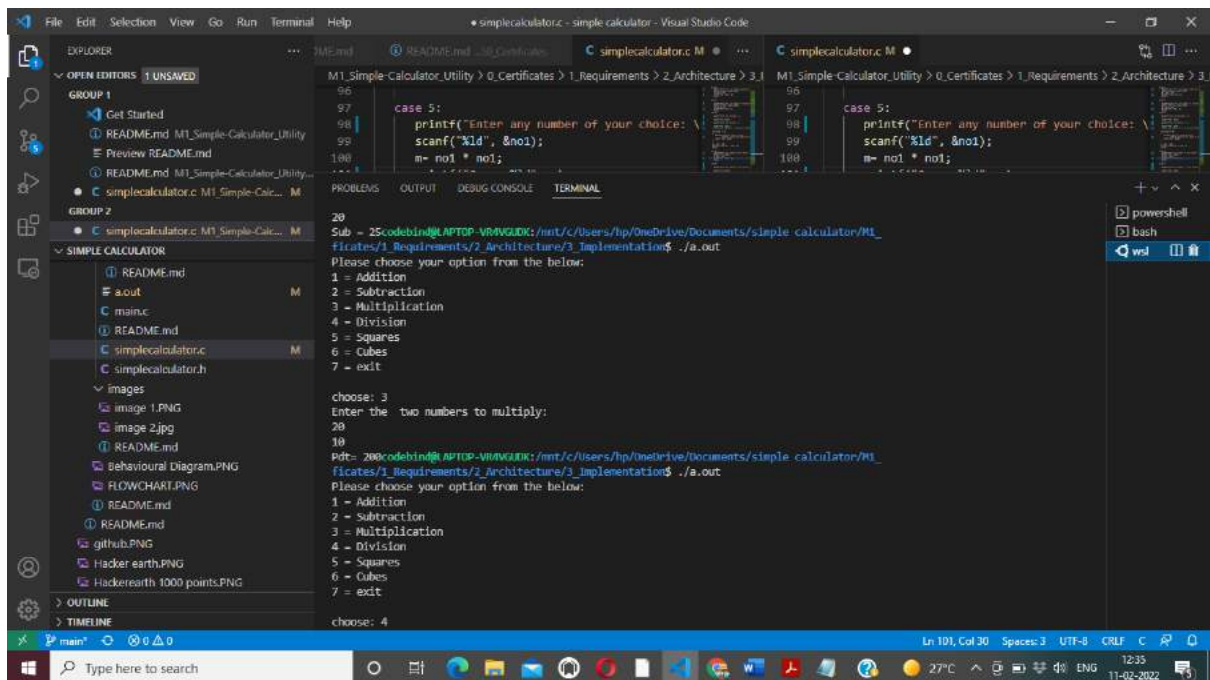
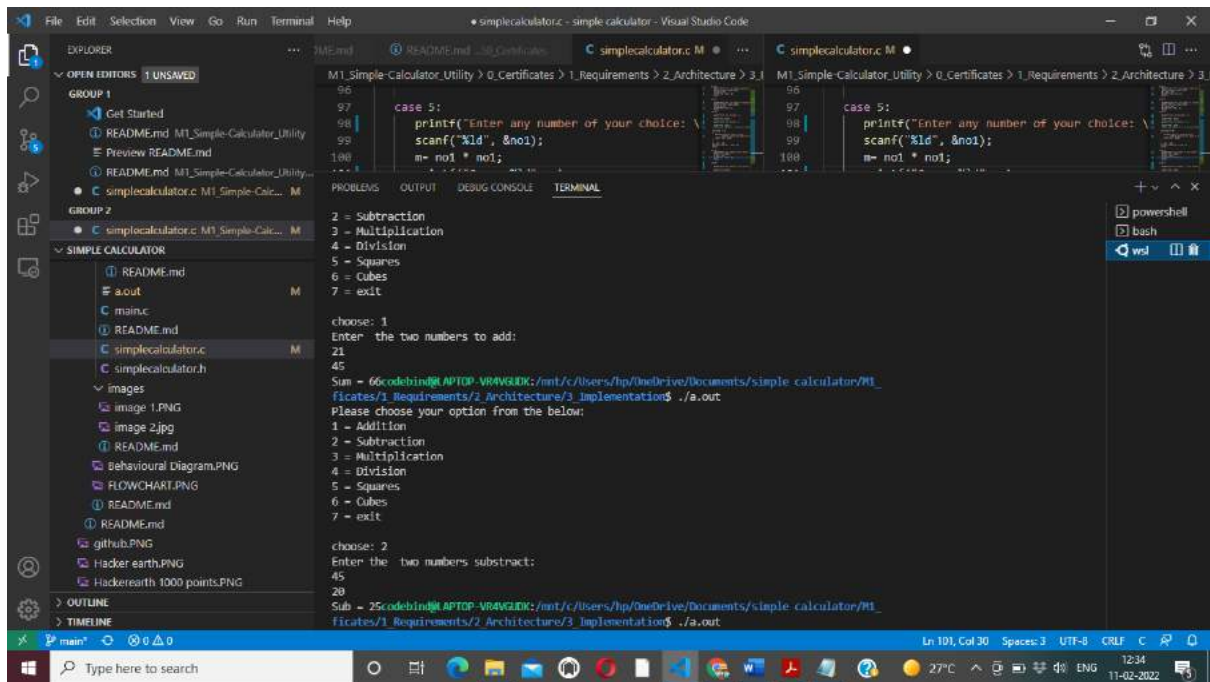
```
}
```

```
}
```

❖ OUTPUT(includes all the results of various operations)



```
codebind@LAPTOP-VIRAVGLEK:/mnt/c/Users/hp/OneDrive/Documents/simple calculator/M1_Simple-C-1_Requirements/2_Architecture/3_Implementation$ ./a.out
Please choose your option from the below:
1 = Addition
2 = Subtraction
3 = Multiplication
4 = Division
5 = Squares
6 = Cubes
7 = exit
choose: 1
Enter the two numbers to add:
45
24
Sum = 69
codebind@LAPTOP-VIRAVGLEK:/mnt/c/Users/hp/OneDrive/Documents/simple calculator/M1_Simple-C-1_Requirements/2_Architecture/3_Implementation$ ./a.out
Please choose your option from the below:
1 = Addition
2 = Subtraction
3 = Multiplication
4 = Division
```



3. For the Multiplication operation after taking two numbers we use '*' operand to do Multiplication.
4. For the Division operation after taking two numbers we use '/' operand to do Division .
5. For the Squaring operation after taking one number we use '*' operand to get the square of the particular number .
6. To Cube a particular number we use '*' opearnd twice in the operation.

7.REFERENCES:

- Referred some raw ideas from web platform and while analysing the code for the project.