

High Level Design & Low Level Design

**Index**

1. Introduction ------------------------------------------------ 3

1.1 Intended audience ------------------------------------------------ 3

1.2 Project purpose ------------------------------------------------ 3

1.3 Key project objective ------------------------------------------------ 3

1.4 Project scope ------------------------------------------------ 3

2. Design overview ------------------------------------------------ 4

2.1 Design objective ------------------------------------------------ 4

2.2 Design alternative ------------------------------------------------ 4

2.3 User interface paradigms ------------------------------------------------ 5

3. System Architecture -------------------------------------------------5

3.1 Structure Details ------------------------------------------------5

3.2 Detailed System Design ------------------------------------------------5

3.3 Flowchart of Server side ------------------------------------------------6

3.4 Flowchart of Client side ------------------------------------------------7

4. Environment Description ------------------------------------------------7

4.1 Time zone support ------------------------------------------------7

4.2 Language support ------------------------------------------------7

4.3 User Desktop Requirements ------------------------------------------------7

4.4 Side Requirements ------------------------------------------------7

4.5 Configuration ------------------------------------------------8

5. Reference ------------------------------------------------8

6. Tools Report -------------------------------------------------9

6.1 CUnit ------------------------------------------------9

6.2 Gcov ------------------------------------------------9

6.3 Gprof ------------------------------------------------13

7. Testing ------------------------------------------------15

7.1 Unit Testing ------------------------------------------------15

7.2 Integration Testing —---------------------------------------------16

8. Requirements Traceability Matrix(RTM) ------------------------------------------------18

1. **Introduction**

The remote calculator is specifically developed for calculation with facilities of addition, subtraction, multiplication, division and exponentiation. Secondly, the application also allows only the authorized user to login to perform the calculations. It works on the client server model where the client can use the calculator if and only if it enters the correct username and password.

**1.1 Intended Audience: -**

The target audience set for this project can be identified as a client who wants to perform some arithmetic calculations like addition, subtraction, multiplication and division.

**1.2 Project Purpose: -**

The remote calculator is a project that helps us understand the basic concepts of functions, socket programming , and data structures.

**1.3 Key Project Objectives: -**

1. Allow the client to be authenticated
2. Server asks for input from the client
3. Server calculates the answer
4. Server sends over the answer to the client
5. Answer is calculated according to the BODMAS rule.

**1.4 Project scope : -**

This project aims to create the development of an Remote calculator application. The remote calculator is specifically developed for calculation with facilities of addition, subtraction, multiplication, division and exponentiation. Secondly, the application also allows only the authorized user to login to perform the calculations. It works on the client server model where the client can use the calculator if and only if it enters the correct username and password.

**2. Design Overview: -**

|  |  |
| --- | --- |
| Name of the Module | Get details |
| Handled by | Sukanya Sahoo |
| Description | It will get the login credentials from user |

|  |  |
| --- | --- |
| Name of the Module | Check format and check format operator |
| Handled by | Yashwanth Tadepalli |
| Description | It will check the format of input Operands and Operators String |

|  |  |
| --- | --- |
| Name of the Module | Solve and Evaluate |
| Handled by | Anisha Surana |
| Description | It will form the expression and evaluates it |

|  |  |
| --- | --- |
| Name of the Module | Server |
| Handled by | Anshul Kumar |
| Description | It forms the TCP socket and communicate with the Client to perform the requested operations |

|  |  |
| --- | --- |
| Name of the Module | Client |
| Handled by | Abhinav Kumar Singh |
| Description | It will connect to the Server and send the requests from user and fetches the result. |

## Design Objectives:

1. Login into the server using the credentials.
2. Input the string of numbers and operands.
3. Form the expression on server side.
4. Solve the expression based on the Precedence of operators .
   1. **Design Alternative: -**

We have used a stack data structure to evaluate the infix expression that is used to calculate the ultimate result.

### 2.3 User Interface Paradigms: -

The remote calculator allows only the registered user to Login and perform the Calculations by giving the two line Input. Where first line should be the number of Operands followed by the operands separated by colon(:). And the second line should be the operators separated by semi-colon(;). The client will take all the inputs and send it to the server for validation and calculations, and then fetches the results from server.

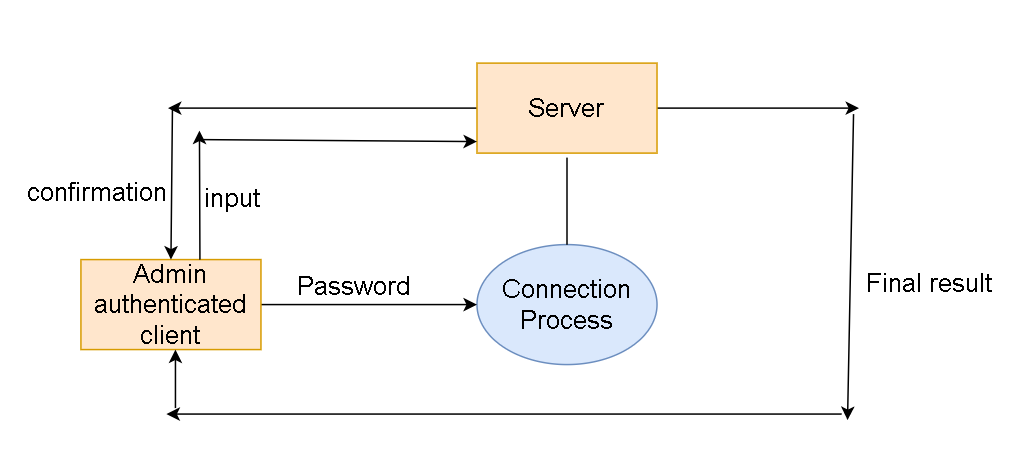
**3.SYSTEM ARCHITECTURE: -**

**3.1 Structure Details:**

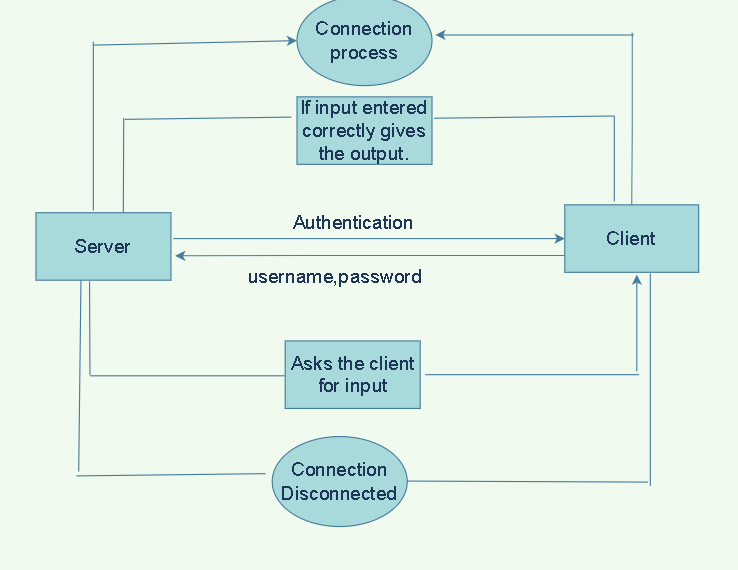
The system consists of a structure :

* **User** : This structure contains the username and password.

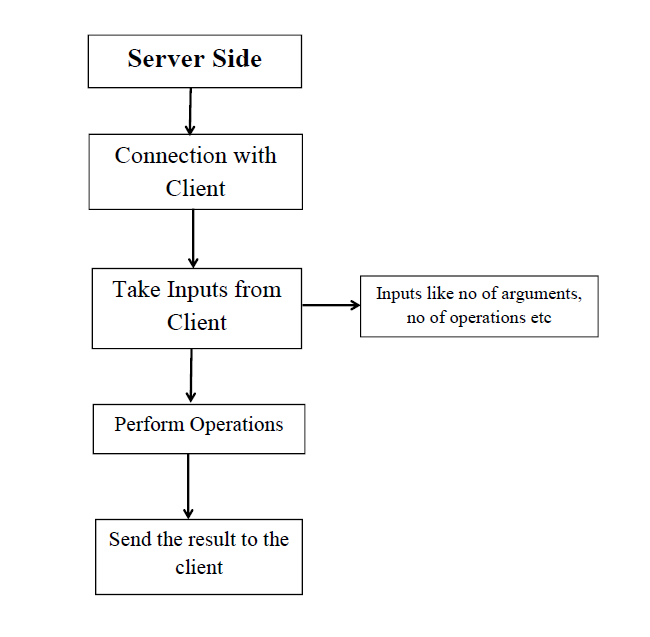
**3.2 Detailed System Design**



**3.1 DFD level 0**

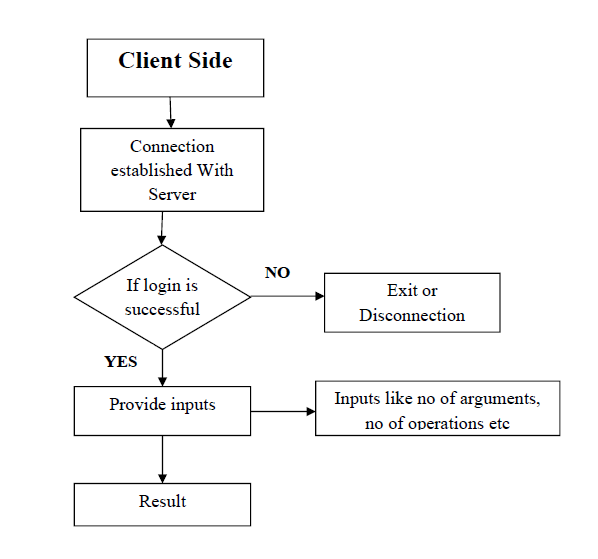


**3.2 DFD level 1**

**3.3 Flowchart of Server Side** 

**3.3 Flow chart for server side of the application**

**3.4 Flow chart of Client side**



**3.4 Flow chart for Client side of the application**

**4. Environment Description: -**

**4.1 Time Zone Support: -** IST- Kolkata

**4.2 Language Support: -** English

**4.3 User Desktop Requirements: -**

* 64-bit processor, 1 GHz or faster
* At least 10 GB free hard drive space
* At least 1 GB RAM **Server**

**4.4 -Side Requirements: -**

* 64-bit processor, 1 GHz or faster
* At least 2 GB free hard drive space
* At least 1GB RAM

**4.4.1** Deployment Considerations: -

* + Local storage is used
  + No network latency to consider
  + To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

**4.4.3**. Database Server Disk Space: -

No such disk space is required as the program is fully functional on online IDE(s) as

well.

**4.4.4**. Integration Requirements: -

* Language: - C
* Tools: - Makefile, Cunit, gprof, gcov
* Complier: - gcc
* Linux Environment

**4.4.5**. Network: - End to End

**4.5 Configuration: -**

**4.5.1**. Operating System: - Linux environment

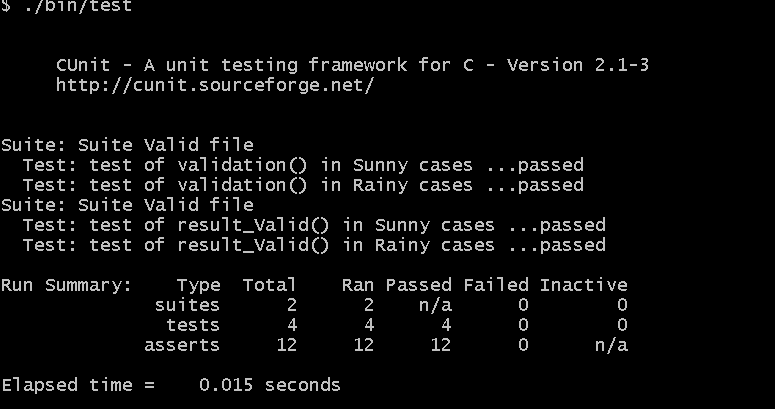
1. **Reference: -**

The references are:

1. <https://www.geeksforgeeks.org/socket-programming-cc/>
2. <https://www.geeksforgeeks.org/introduction-to-stack-data-structure-and-algorithm-tutorials/>
3. <https://www.geeksforgeeks.org/tcp-ip-model/>
4. <https://www.geeksforgeeks.org/layers-of-osi-model/>
5. <https://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>

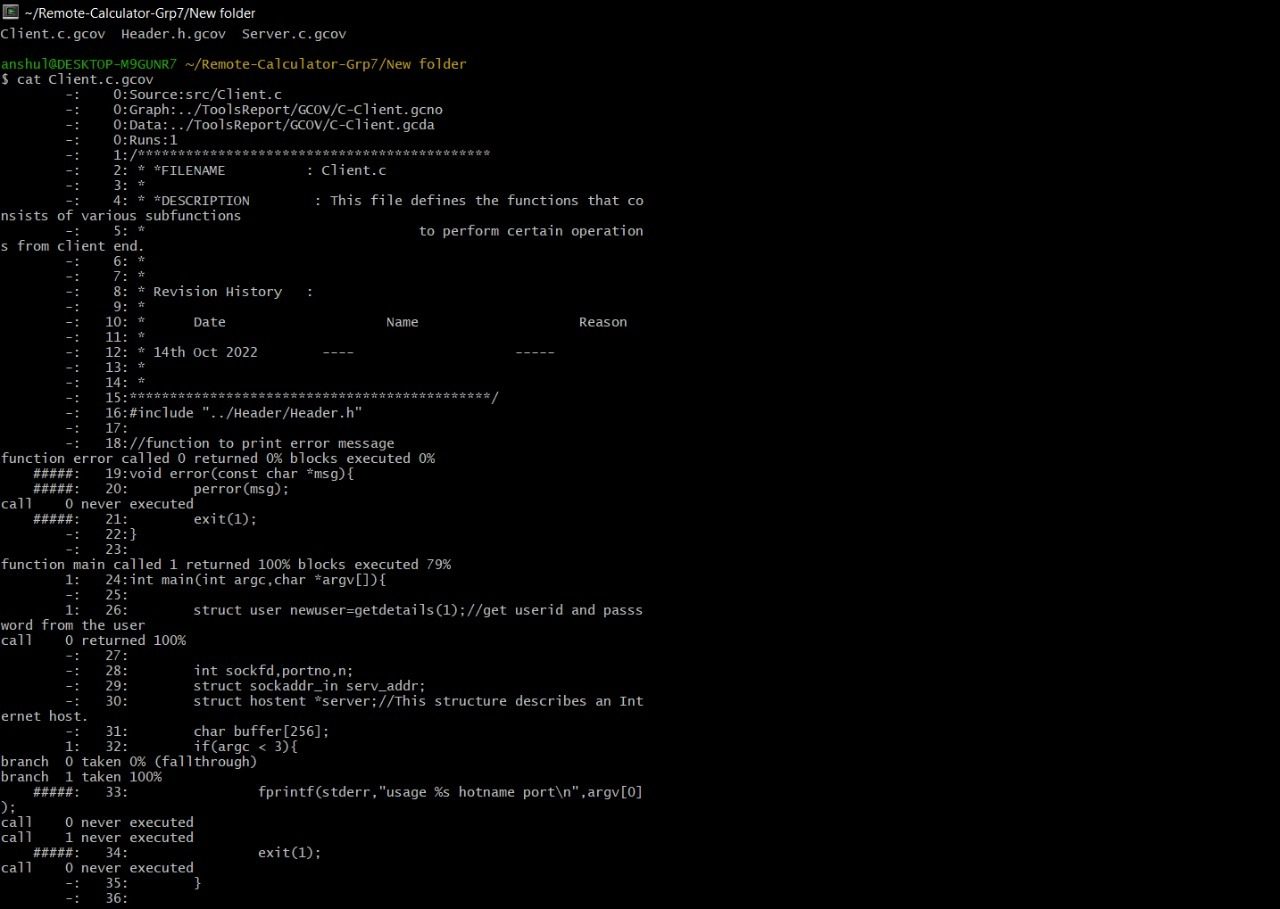
**6. TOOLS REPORT**

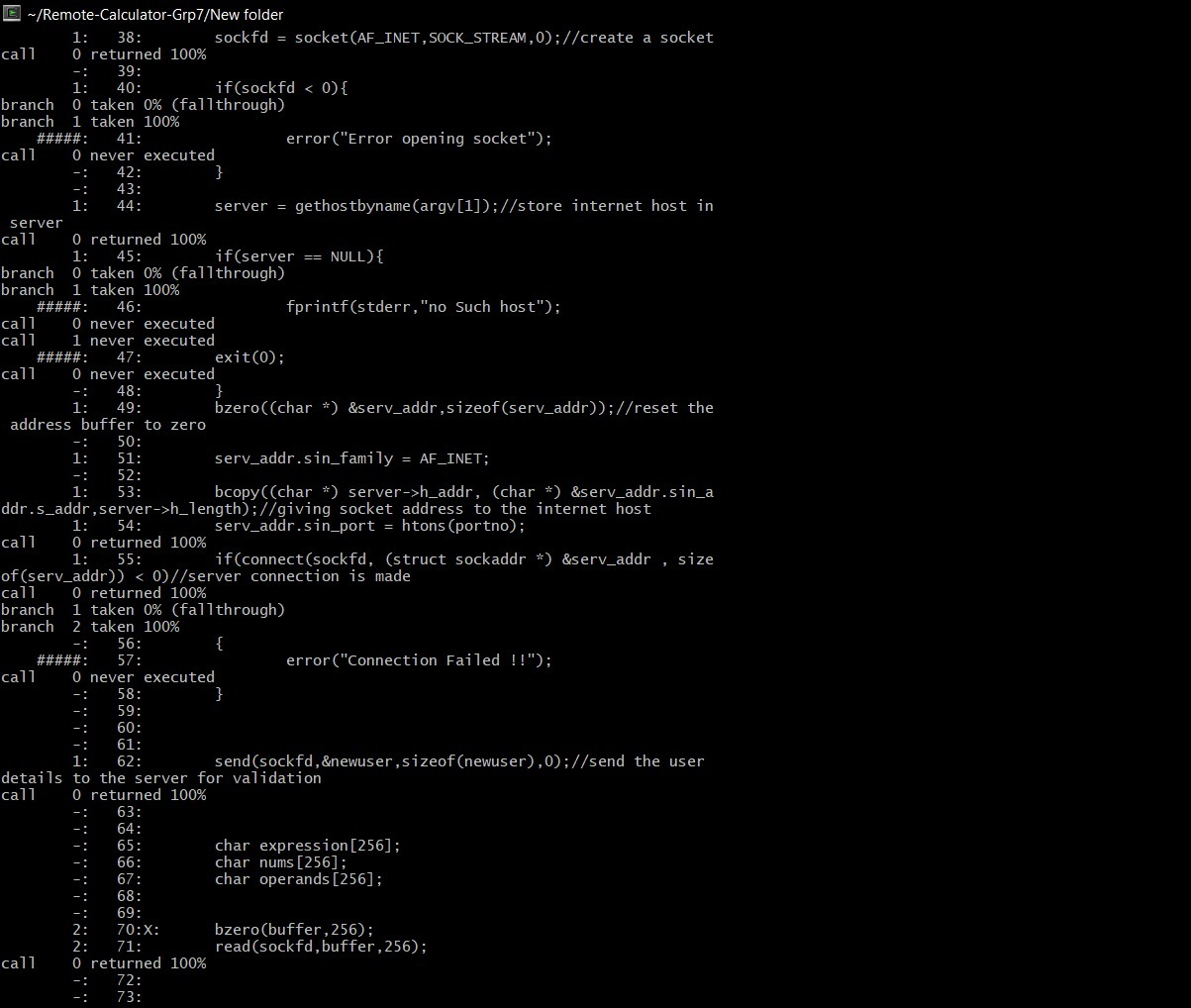
**6.1 CUnit**

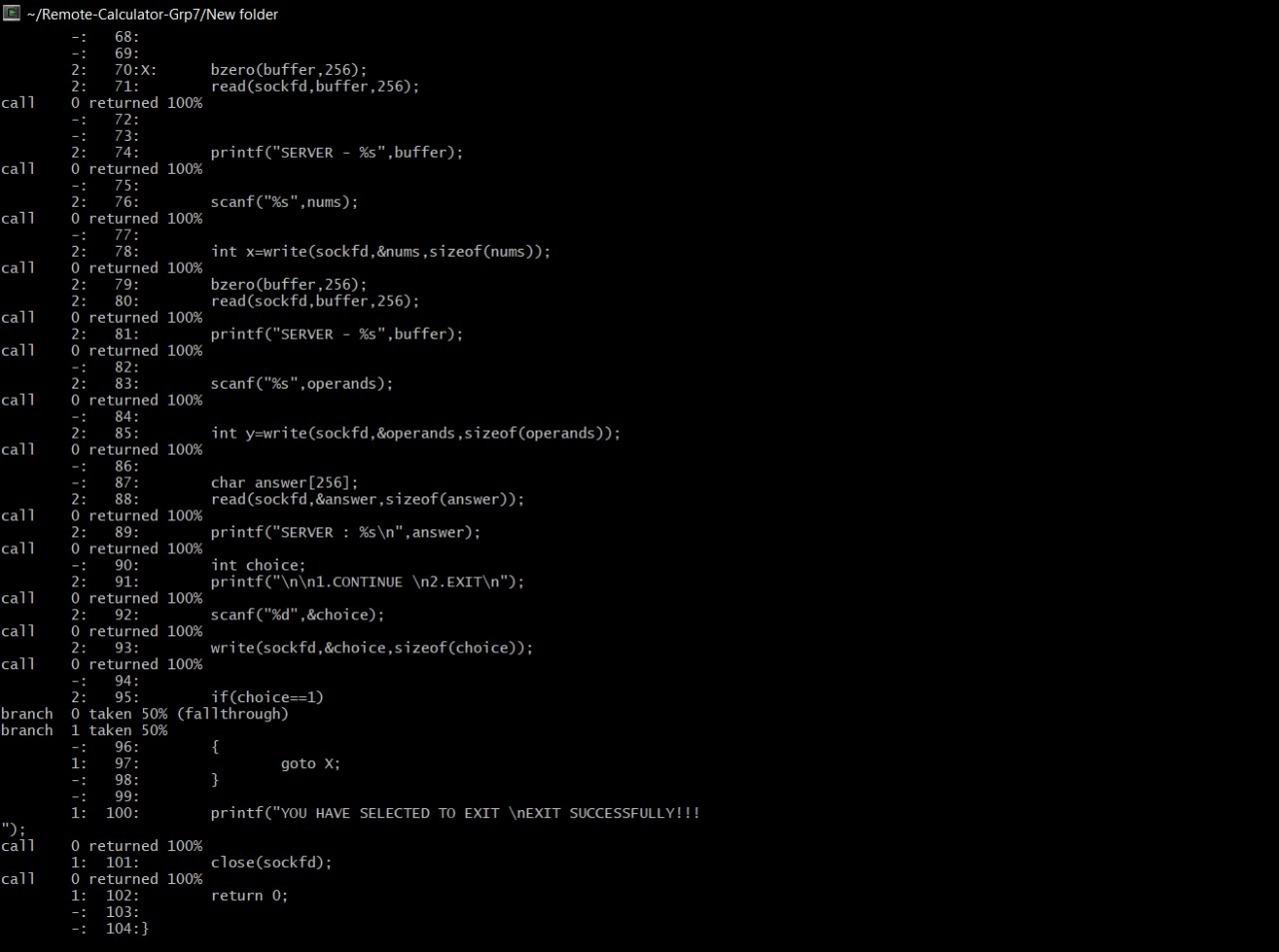
****

**6.2 GCOV**

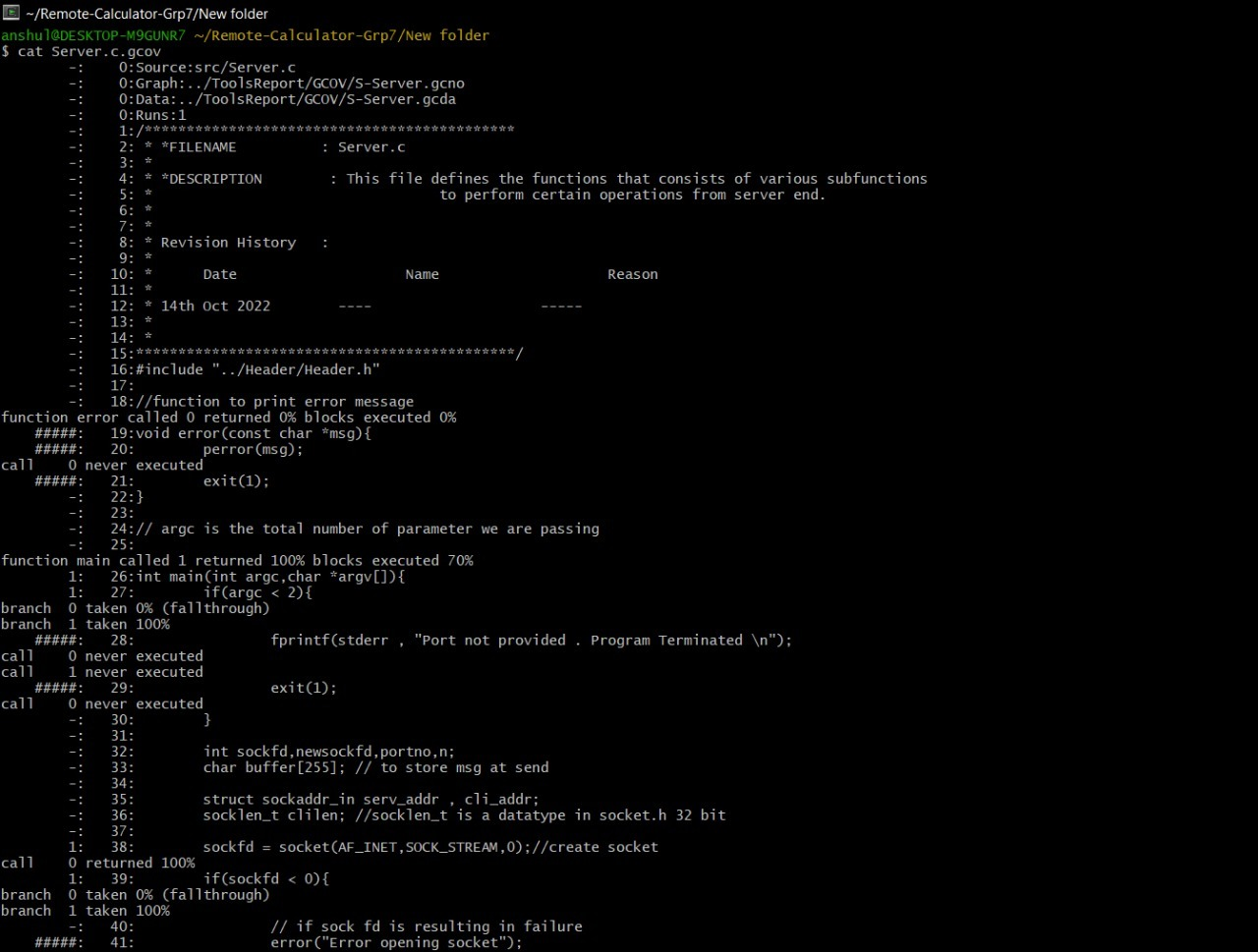
**Client.c.gcov**







**Server.c.gcov:**



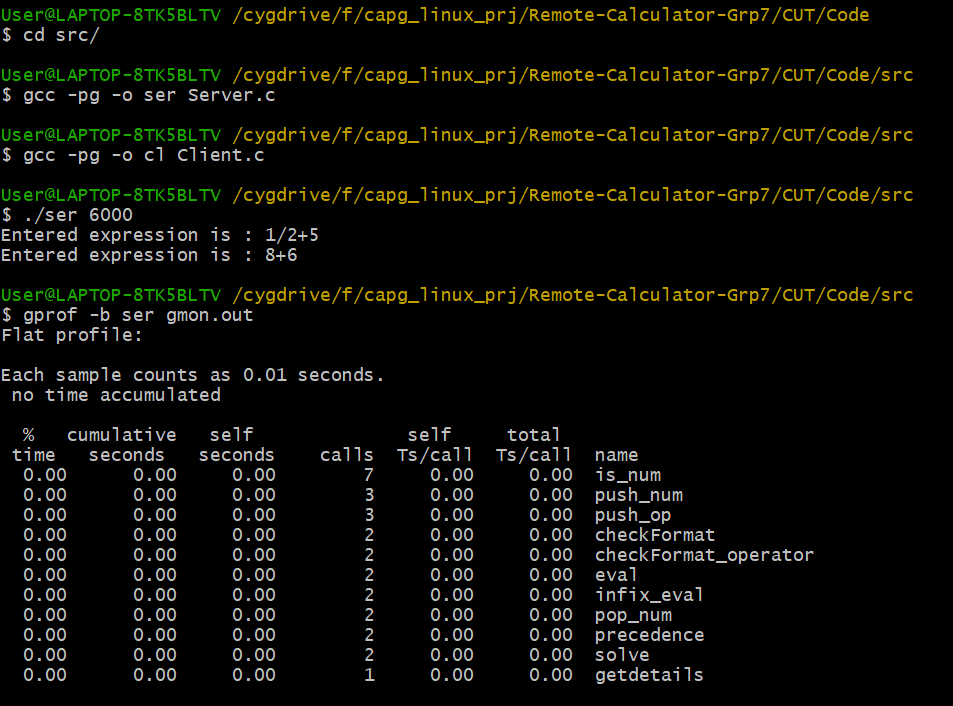


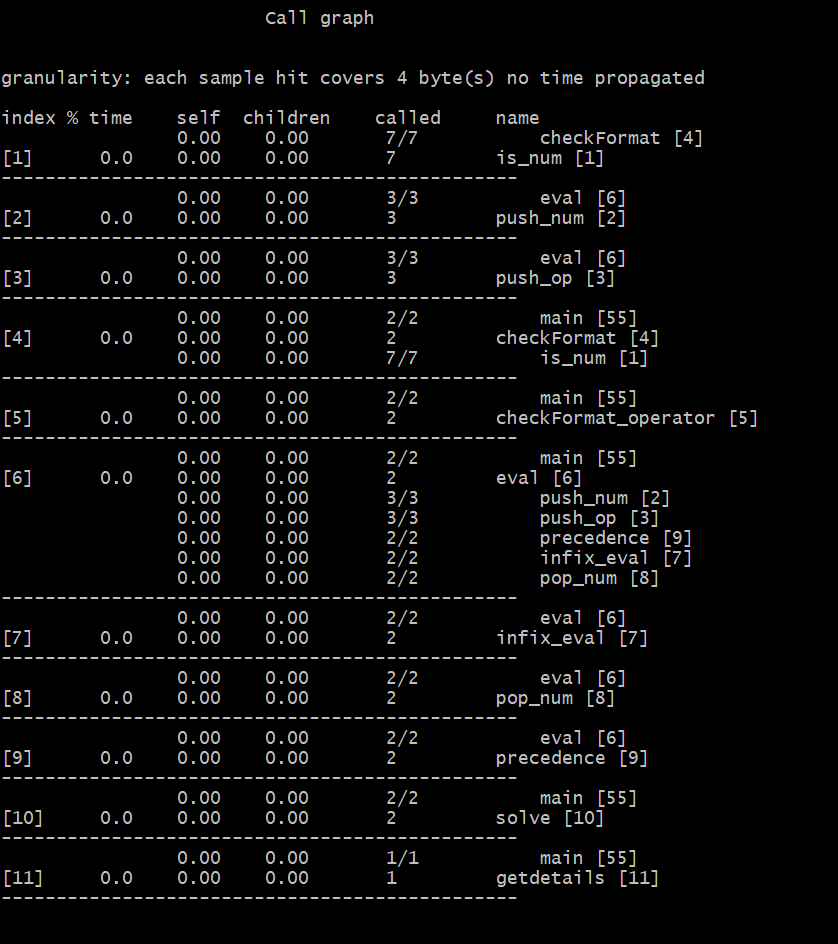


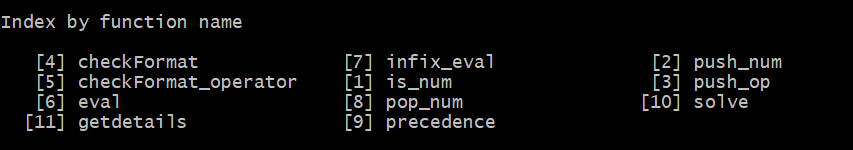


**6.3. GPROF**

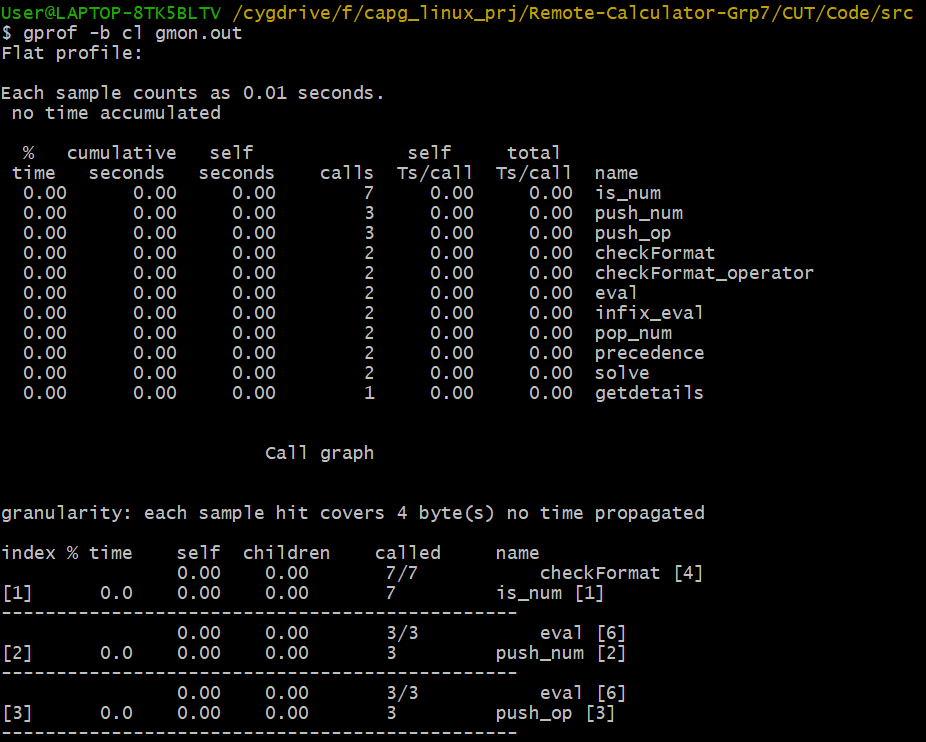
**Server side Gprof Result**

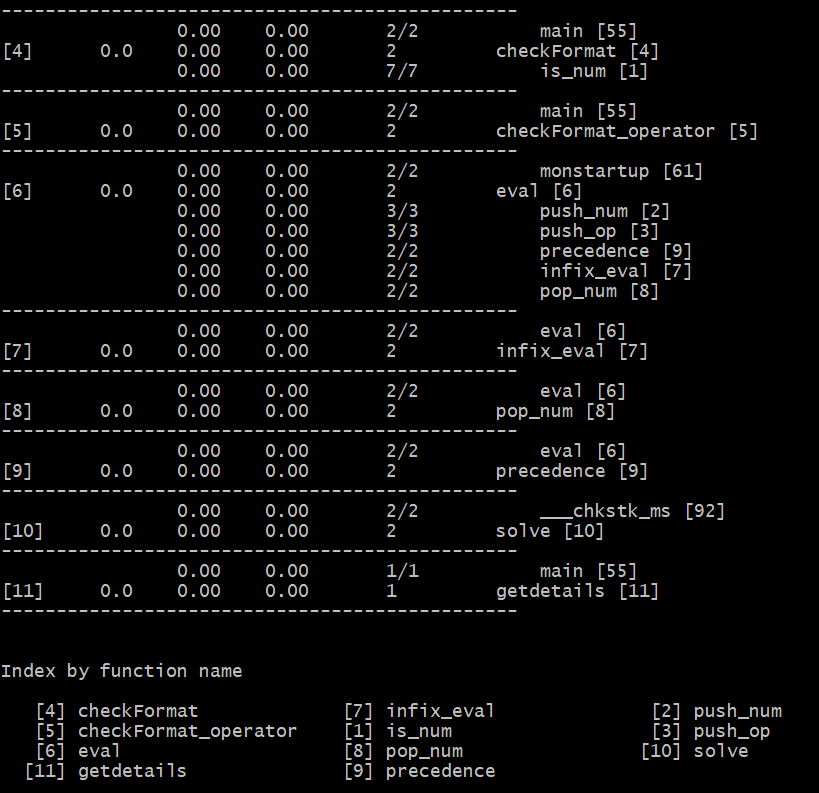






**Client side Gprof results**

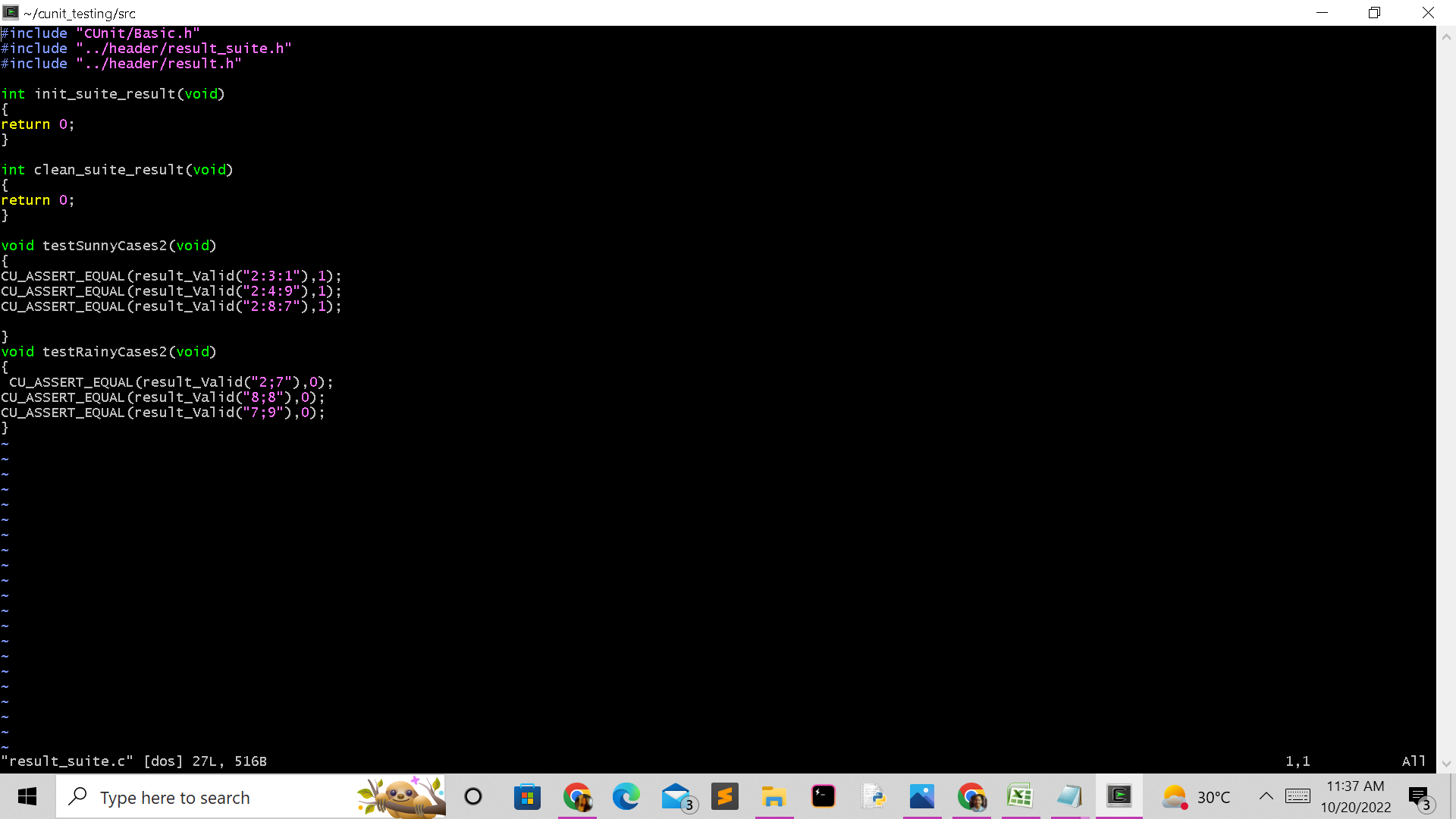




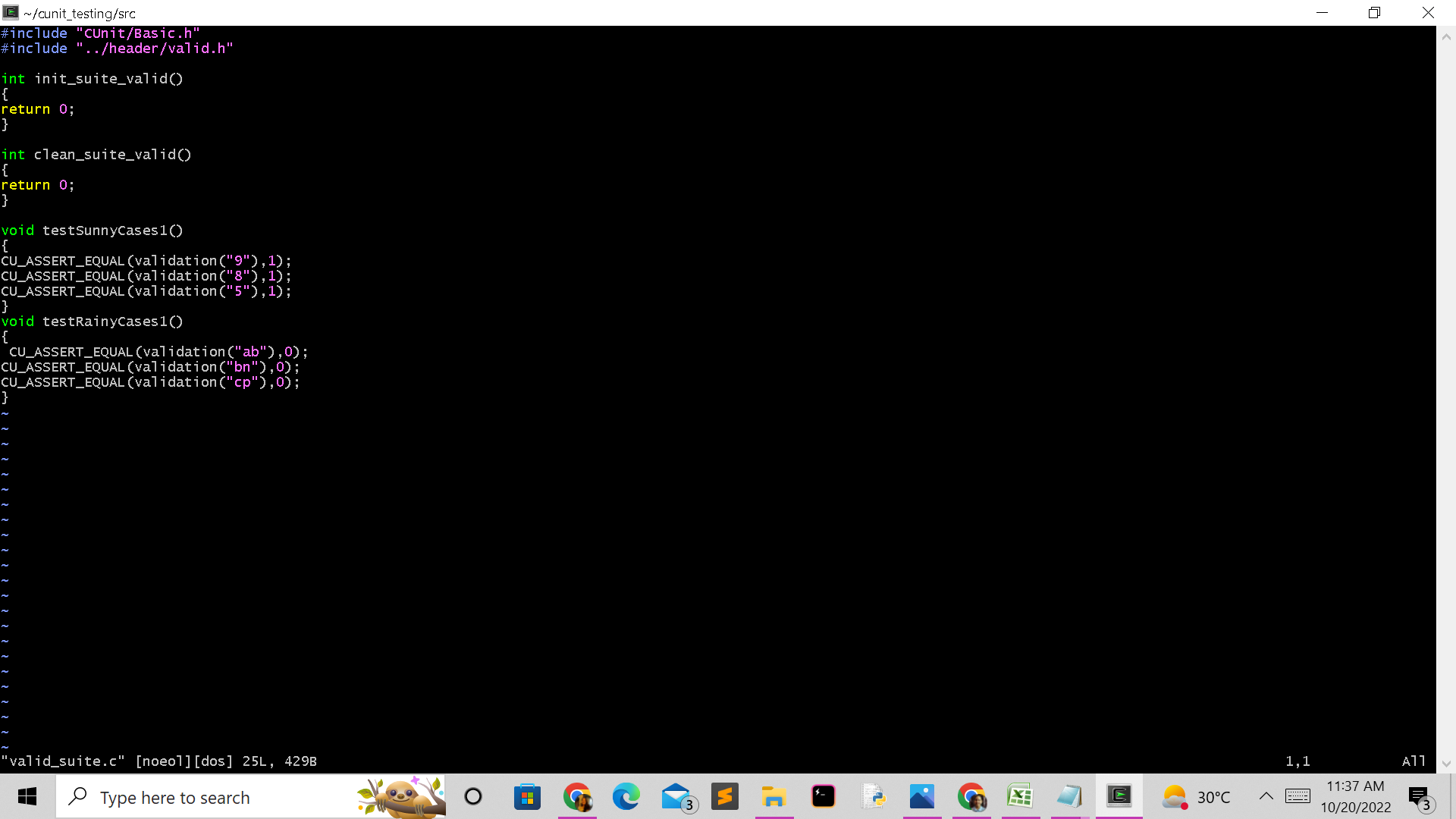
**7. Testing**

**7.1 Unit Testing**

1. **UT\_Case 1**

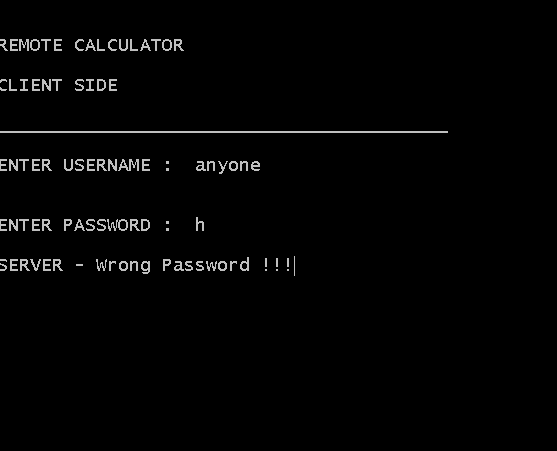
****

1. **UT\_Case 2**

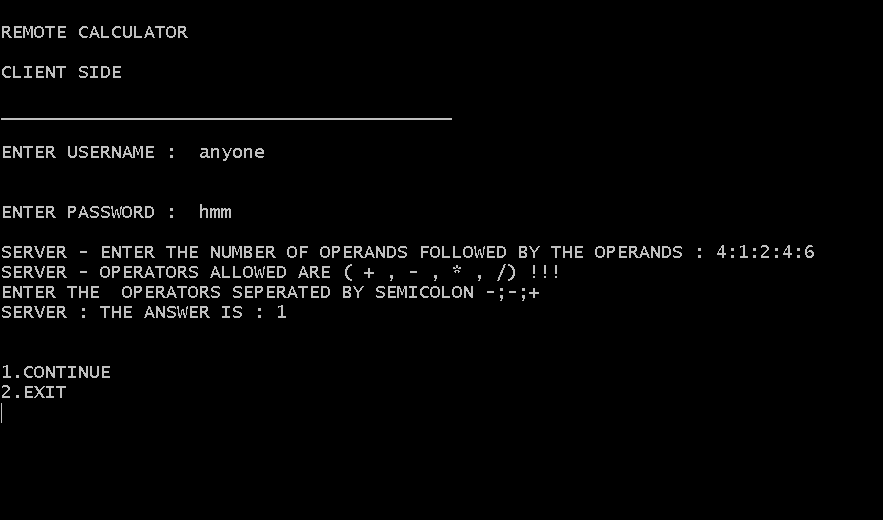
****

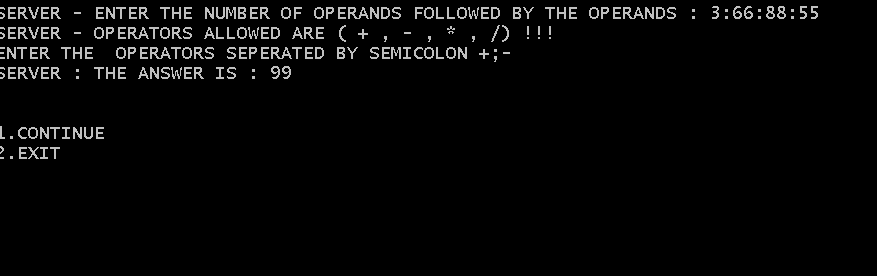
**7.2 Integration Testing**

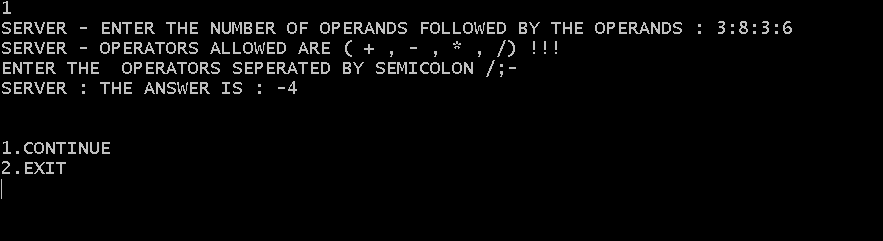
**IT\_CASE 1:- CHECKING FOR WRONG PASSWORD**

****

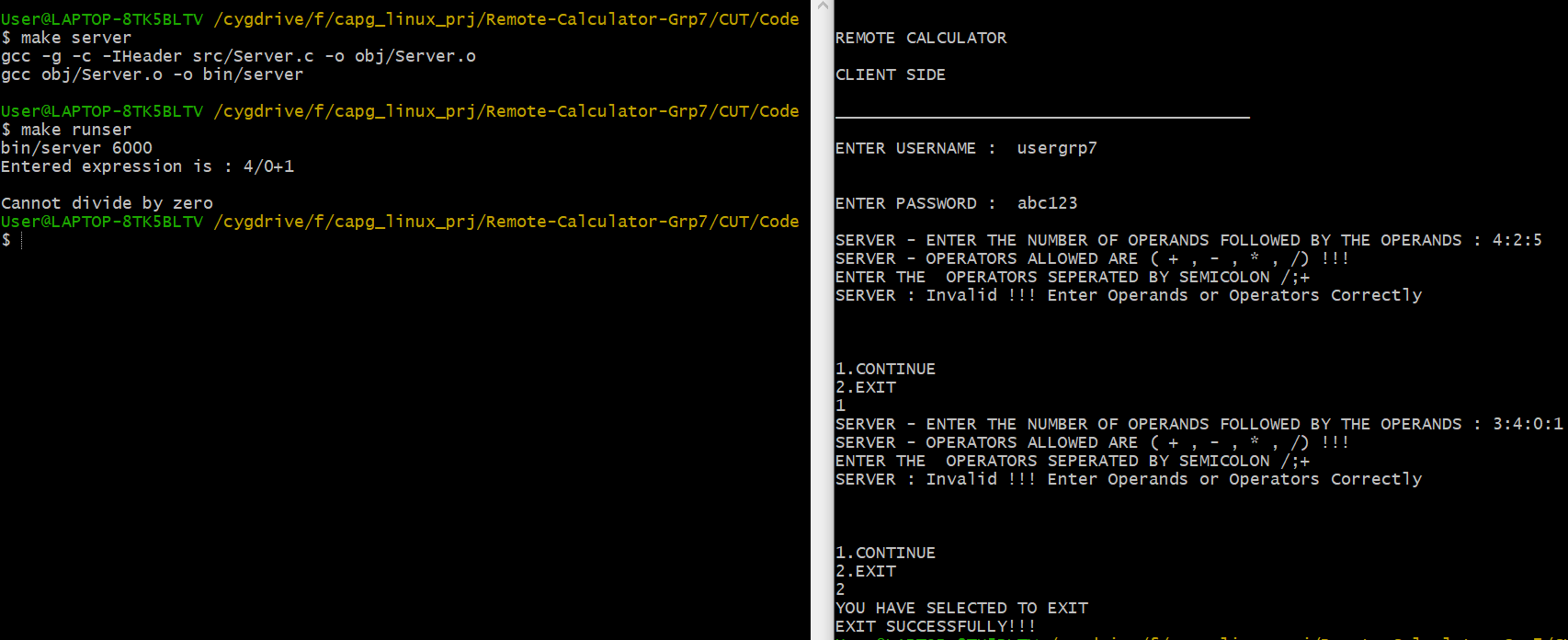
**IT\_CASE 2:- CHECKING FOR VARIOUS ARITHMATICAL OPERATIONS:**

****

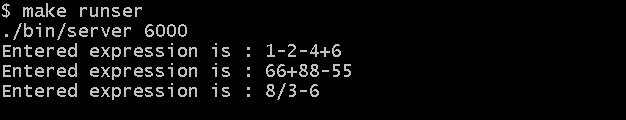
****

****

**IT\_CASE 3:- CHECKING FOR WRONG FORMAT**



**IT\_CASE 4:- OUTPUT ON SERVER SIDE**

****

**8. Requirements Traceability Matrix(RTM)**

