

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1 | 7/12/2020 | THIRLUKAVENKATA SAI YAMINI |  |  |  |
| 2 | 7/12/2020 | BONAGIRI GREESHMA |  |  |  |
| 3 | 7/12/2020 | SANA MD |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

# 

Contents

|  |  |
| --- | --- |
| 1. Requirements   * 1. High Level Requirements   2. Low level Requirements | 5 |
| 2. Description  2.1 Research – Costing  2.2 Aging  2.3 4W1H Analysis  2.4 SWOT Analysis | 6  6  7  7 |
| 3. Design | 8 −11 |
| 4. Implementation  4.1 Git Hub Link | 12 |
| 5. Test Plan | 12 −13 |
| 6. Agile Aspects | 13 |
| 7. Challenges faced | 13 |
| 8. Summary | 14−15 |
| 9. References | 15 |
|  |  |
|  |  |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S. No** | **Figure Name** | **Page No.** |
| 1 | Calculator | 6 |
| 2 | Block Diagram | 8 |
| 3 | Activity Diagram | 9 |
| 4 | Component Diagram -1 | 9 |
| 5 | Sequence Diagram -1 | 10 |
| 6 | Class Diagram -2 | 10 |
| 7 | Sequence Diagram -2 | 11 |
| 8 | Git hub Dashboard | 12 |
| 9 | Unit Testing | 15 |
| 10 | Our Contributions | 15 |

# 

# 

# Mini project on “Scientific Calculator”

# Objectives :

# 1. To be familiar with C programming language.

# 2. To implement various mathematical operations in C language.

# 3. To build a simple project on C.

**1. Requirements:**

An **electronic calculator** is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics. The calculator has the following keys: 0 to 9 numbers and the following symbols +, -, \*, /, ±. On encountering a division by 0 the display should read "Cannot divide by 0" and typing the key “C” should reset the calculator. • On calculating the square root value of a negative operand the display should read "Wrong operand". • On erroneous operand or operation keys the display should read “Reset (C) to continue” or “Clear (CE) to continue” as appropriate. Of course, any situation can be cleared using the main reset key “C”.

**1.1 High Level Requirements**:

* Ability to perform all operations like arithmetic
* Binary functions.
* Scientific Calculations
* Digital operations

**1.2 Low Level Requirements:**

* Calculation the square root
* Ability to perform Basic functions and exponents
* On the calculation of Inverse and Matrix operations
* Trigonometric

**2. Description:**

The project will be a simple Scientific Calculator implementation in C. Here We use various calculations to calculate different mathematical operations. The output of the project will be a console application without any graphics. You should implement at least following mathematical operations. operations like Addition, Subtraction, Multiplication. Calculators also have the ability to store numbers into computer memory. Basic calculators usually store only one number at a time; more specific types are able to store many numbers represented in variables. The variables can also be used for constructing formulas. Some models have the ability to extend memory capacity to store more numbers; the extended memory address is termed an array index.

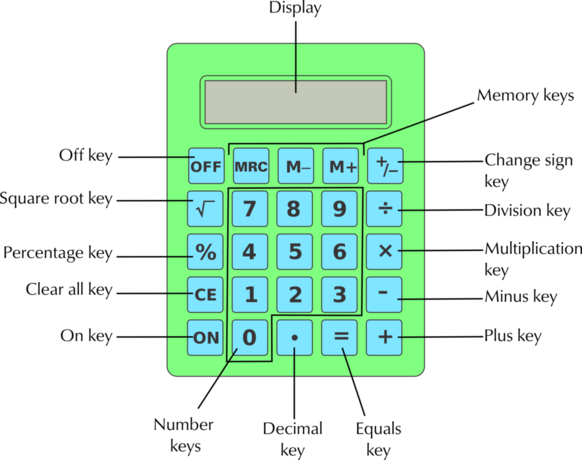


Figure 1. Calculator

**2.1 Research – Costing**

|  |  |  |
| --- | --- | --- |
| **Type** | **Features** | **Price** |
| Abacus | It holds vertically arranged rods with beads. Each rod has unique place value and each bead has a number | $27 |
| Basic | Has 8 – 12 digits display physical keys for input. Works on both battery and solar energy. | $2 |
| Scientific | Include statistical and trigonometric calculations, and some even perform computer algebra. | $17 |
| Online | Performs basic operations and even more. No need of any hardware integration. | $0 |

**2.2 Aging:**

Around 2,500 B.C the invention of abacus calculations that had been considered extremely difficult became routine. In 1617 Napier published Rabdology (calculation with rods) inscribed with multiplication tables. In 1642 Pascal invented calculator that operates by means of clockwork-type mechanism. From 1948 Curta calculators ruled out and ended in 1970 with the advent of electronic calculator.

**2.3** **4W 1H:**

|  |  |
| --- | --- |
| WHAT | The main aim of this project is it to implement a basic calculator with all configurations in c language |
| WHY | To perform operations like add, sub, divide, multiply. |
| WHERE | It is used in anywhere for all scientific purpose |
| WHEN | It is used to store the values also and produce the outputs |
| HOW | By using software algorithm. |

**2.4 SWOT Analysis:**

**Strengths:**

* To help students understand when replicas and how to apply mathematical methods
* To help allow students to make quick ,accurate mathematical calculations
* To expand students the scope of their mental abilities
* To teach logical thinking and problem solving skills
* To check their results for mathematical accuracy

**Weaknesses:**

* Time consumption
* Unclear value proposition
* Inefficient process

**Opportunities:**

* Increasing interest
* virtual Visits

**Threats:**

* Low Profit
* Outdated Technology

**3. Design**

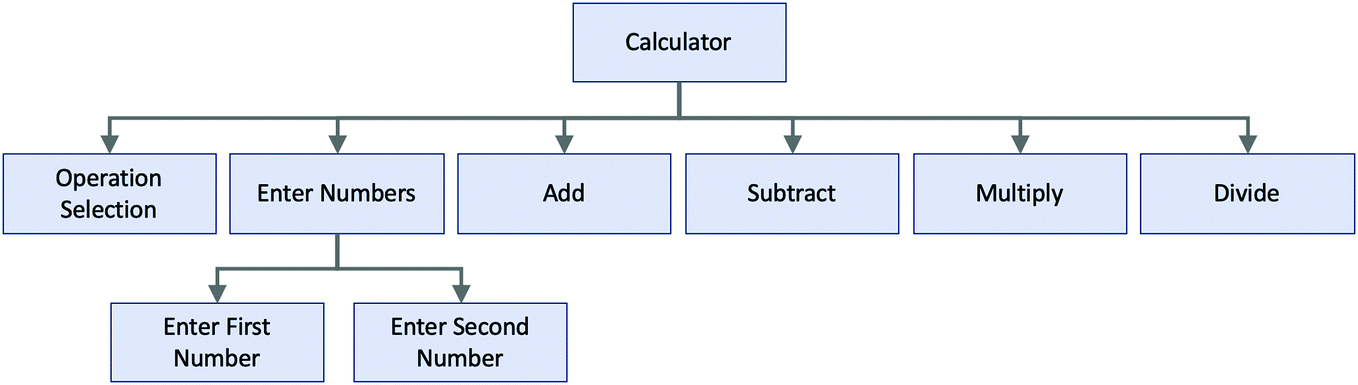


Figure.2 Block Diagram

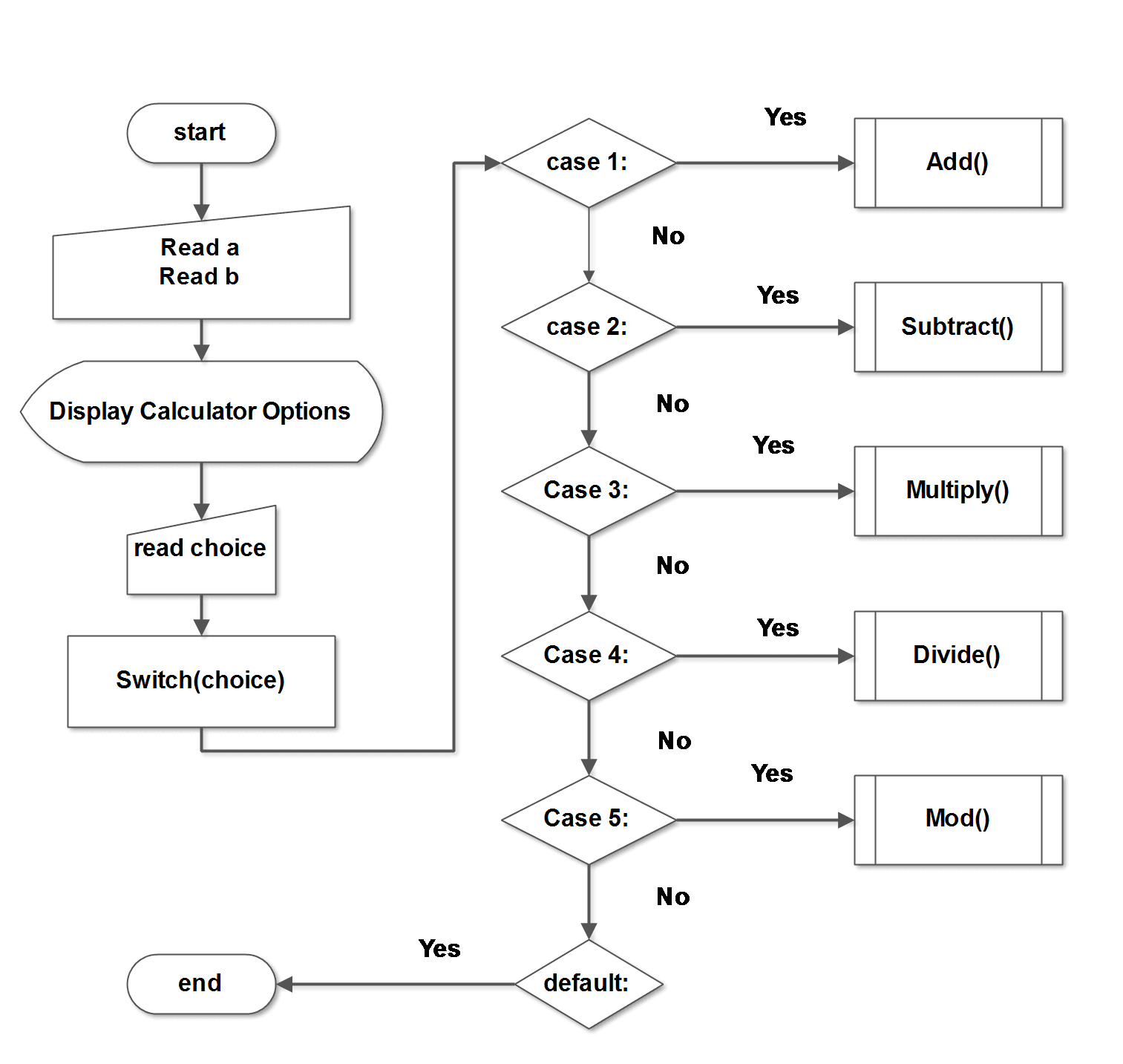


Figure 2. Activity Diagram

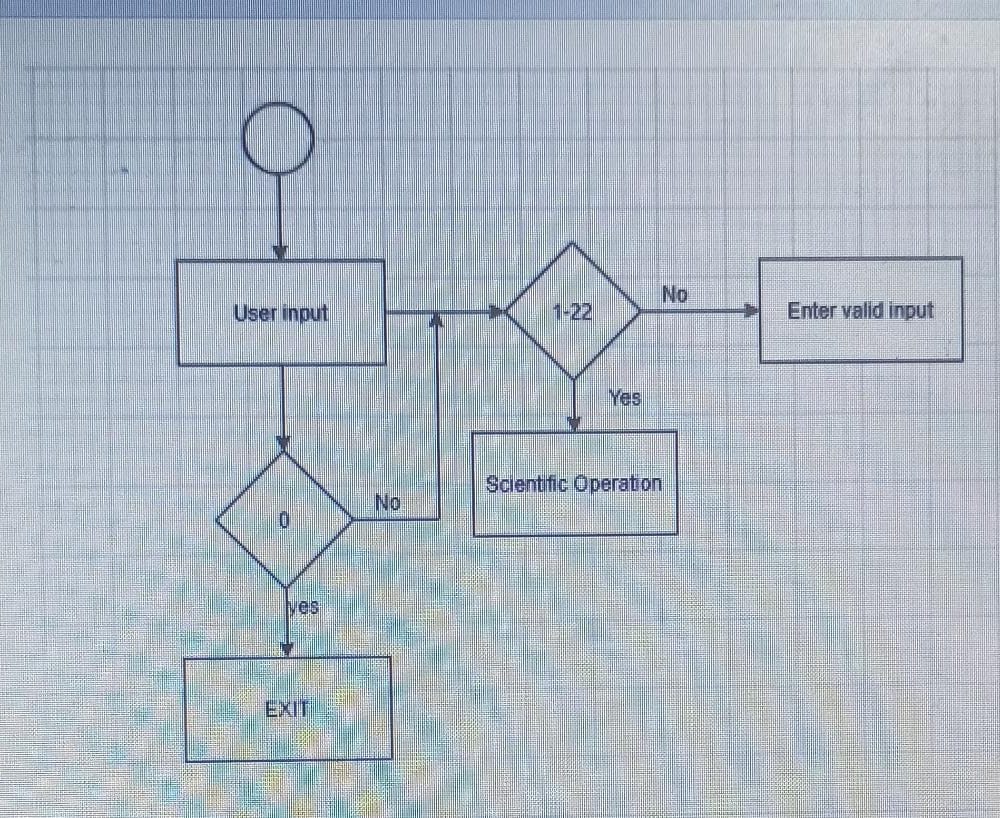
****

Figure.4 Component Diagram 1

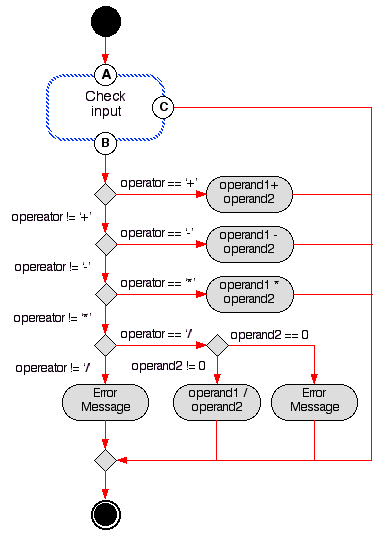


Figure.5 Sequence Diagram 1

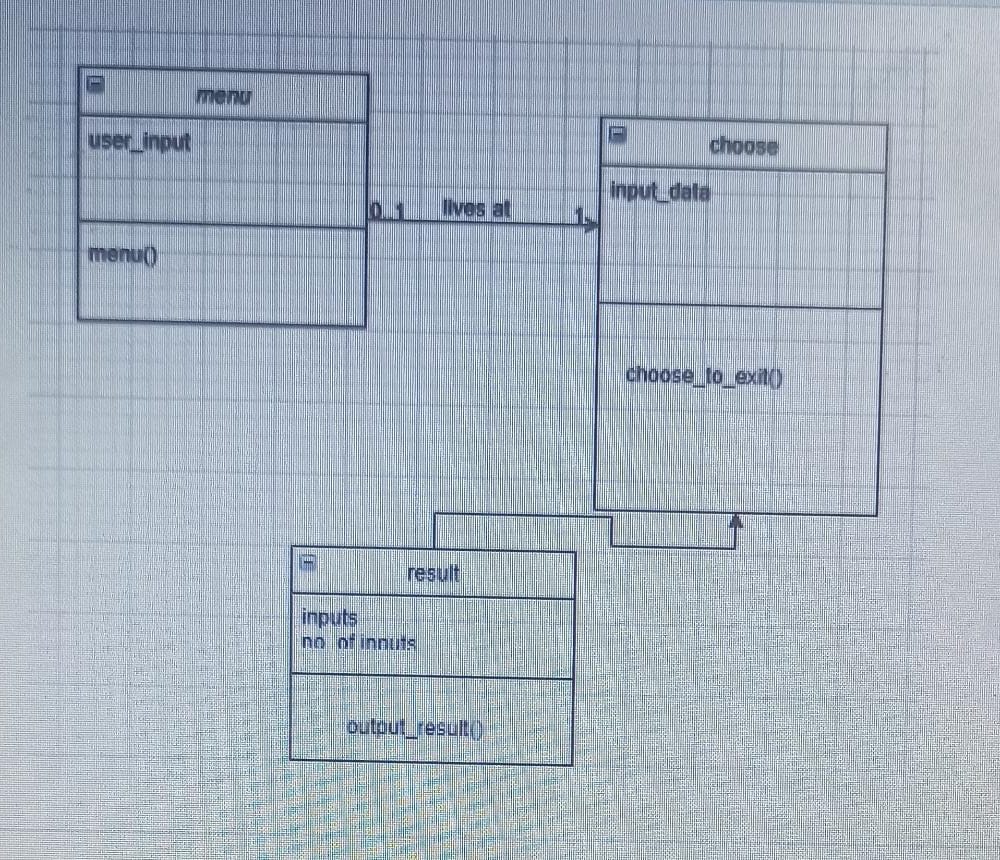
****

Figure.6 Class Diagram 1

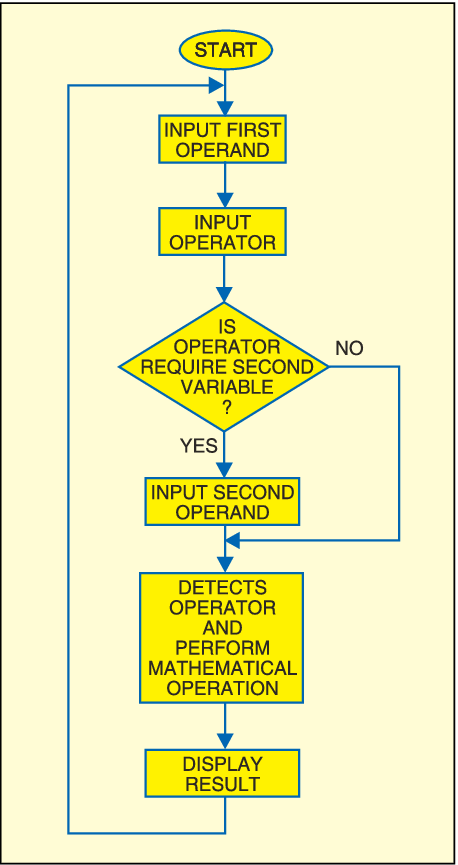


Figure.7 Sequence Diagram 2

**4. GIT HUB link:**

**URL:** [**https://github.com/99003180/calculator.git**](https://github.com/99003180/calculator.git)

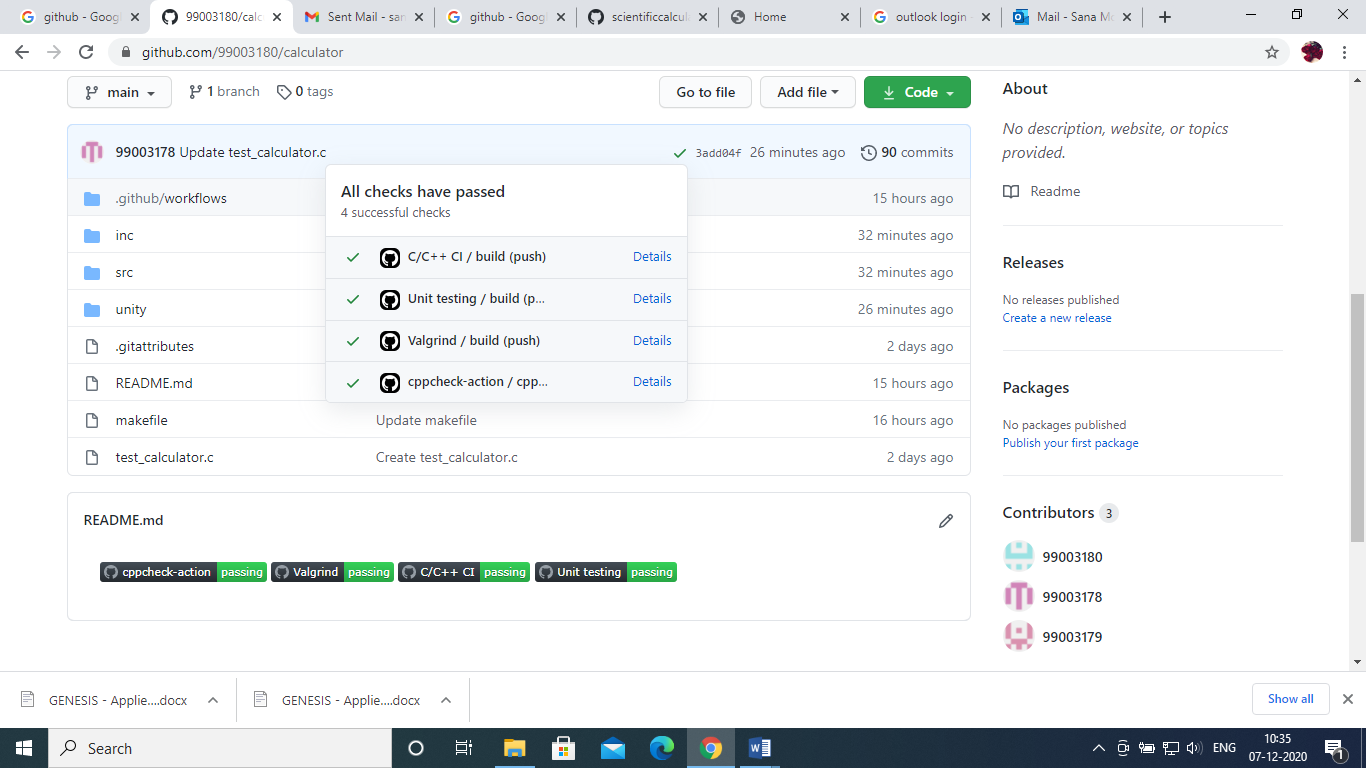
****

Figure.8 Git hub dashboard

**5. Test plan**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | DESCRIPTION | INPUTS | EXCEPTED OUTPUTS | TEST RESULTS |
| T1 | Addition | (4,6)  (65535,65535)  (-50,-50)  (50,-50) | 10  131070  -100  0 | PASS |
| T2 | Subtraction | (6,3)  (3,-2)  (10,5) | 3  1  5 | PASS |
| T3 | Multiplication | (6,2)  (7,3)  (5,15)  (100,190) | 12  21  75  19000 | PASS |
| T4 | Division | (100,10)  (7,0)  (6,6) | 10  0  1 | PASS |
| T5 | Reverse | 321  12367 | 123  76321 | PASS |
| T6 | Factorial | 1  5 | 1  120 | PASS |
| T7 | Modulus | (100,10)  (10,8) | 0  2 | PASS |

**6. Agile Aspects**

**6.1** To add, subtract, multiply and divide the given integers

Test Case:

* For given integers, the results should be accommodated in larger memory
* Result should be available in no time

User Stories:

During calculation of large numbers, I got results within no time. For example, adding 65535 and 65535, I could get the result of that numbers.

**6.2** To multiply two integers

Test Case:

* The result should be accommodated in larger memory

User stories:

Usually when two integers are large, the result will be some garbage value

But this calculator handles efficiently.

**6.3** To Divide two integers

Test Case:

* The result should be available in no time
* Error message when denominator is 0

User stories:

Whenever denominator was given as 0, it will give output as 0 and indicates a message to the user that they are trying to divide numerator with 0 i.e., Divide by zero error.

**7. Challenges faced**

* Setting up Git Bash, Visual Studio
* Committing individual contributions into Git
* Performing Unit testing for each function
* Faced issue with Valgrind i.e., Dynamic analysis

**8. Summary**

**99003178:**

My contribution in the project was to write modulus operation, Sum of two matrix and decimal to Binary converter.

In modulus operation, two operands were taken as input, remainder was sent as output.

In matrix\_sum operation, for loops were used to take user inputs and calculated Sum.

In Decimal to binary conversion, decimal inputs were converted to binary.

**99003179:**

In the project calculator my contribution is to perform Power, factorial and reversing of given number.

Factorial of a non-negative integer, is multiplication of all integers smaller than or equal to n. For factorial one number is taken as user input and the input value is less than zero then factorial is not possible. If the input value is greater than zero then it returns the result.

For reverse of a number, one number is given as input then the modulo operator (%) returns the remainder of a division. Last Digit to **reverse**. Remove last digit from **number**. **number** = **number** / 10, then it returns the result.

For power operation, two inputs should be given then the power operations will perform.

**99003180:**

My contribution in the project Calculator was designing a calculator with arithmetic functions like Addition, Subtraction, Multiplication, Division and Modulus.

For addition, two numbers were taken as user input. The result was stored in larger data type so that we don’t get wrong results when adding large numbers. For example, adding 65535 with 65535 gave result as 131070 and it was declared as double. Boundary conditions were taken care of.

For subtraction, two numbers were taken as user input and result was published. All boundary conditions were taken care of.

For Multiplication also, result was stored in larger data type in order not to get wrong results.

For Division, whenever user gave 0 as denominator, Divide by 0 error was handled by sending a message to user.

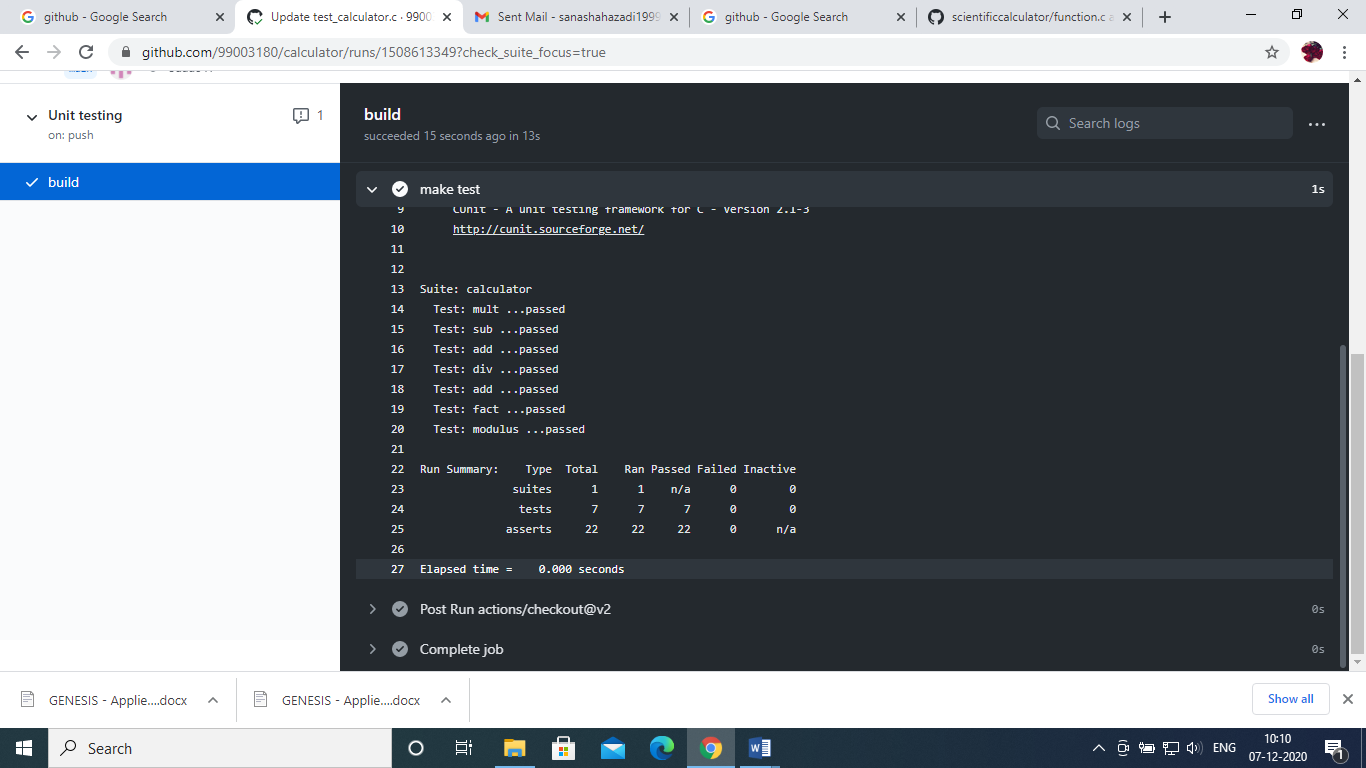


Figure. 9 Unit Testing

Above figure shows the Unit testing performed. Unit testing was performed by considering all boundary conditions. All the test cases passed.

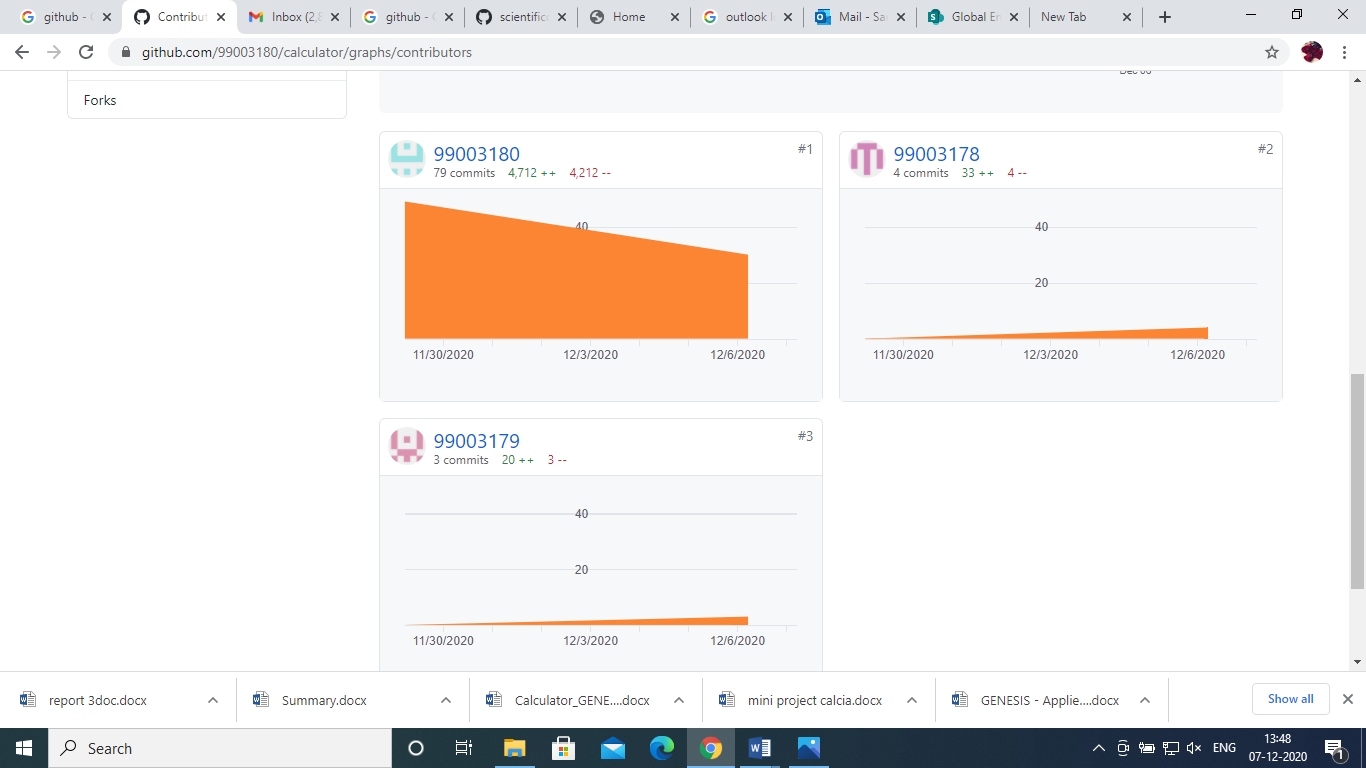


Figure 10. Our Contributions

**REFERENCES:**

[1] Orange HRM Open Source, Retrieved: November 4, 2013. From: http://www.orangehrm.com/open-sourceproduct-features-pim.shtml

[2] A.S. Syed Navaz, A.S.SyedFiaz, C.Prabhadevi, V.Sangeetha, S.Gopalakrishnan, “Human Resource Management System”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 8, Issue 4 (Jan. – Feb. 2013) Page 62-71. [

[3] Julie Bulmash, “Human Resource Management and Technology”, Chapter 3 [4] TECH HRM (Human Resource Management System), Retrieved: November 4, 2013.

 From: http://www.techjetsolutions.com/brochure/TECHHRM. pdf [5] Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”,