**Team 9**

**Quick Calc**

**Test Procedure**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Authors** | **Description of Change** | **Sections** | **Rev** | **Date** |
| Brody Whelan | Added test cases arithmetic operators, history, binary, and hexadecimal | 1,2,3 | 1 | 12/06/2023 |

**Table of Contents**

[1 Team Description 4](#_Toc509916186)

[2 Introduction 4](#_Toc509916187)

[2.1 Identification 4](#_Toc509916188)

[3 Test Procedures 4](#_Toc509916189)

[3.1 Test Case 1 5](#_Toc509916190)

[3.2 Test Case 2 6](#_Toc509916191)

[3.3 Test Case 3 7](#_Toc509916192)

[4 Verification Cross Reference Matrix 8](#_Toc509916193)

# Team Description

|  |  |
| --- | --- |
| **Team Member Name** | **Email Address** |
| Victor Prieto | victorprie@csu.fullerton.edu |
| Lyla Traylor | lylatraylor@csu.fullerton.edu |
| Brody Whelan | bwhelan212@csu.fullerton.edu |

# Introduction

Quick Calc is a calculator program that allows the user to complete a variety of calculations depending on their needs. This application does so by allowing users to perform arithmetic operations like addition, subtraction, modulo, multiplication and division. You can also convert a number to its binary or hexadecimal form.

## Identification

|  |  |
| --- | --- |
| **Requirement Document Tested:** | 12-06-2023 |
| **Requirement Document Revision:** | 12-05-2023 |
| **Revision Release Date:** | 12-06-2023 |

# Test Procedures

## Test Case 1

Description: Testing arithmetic operators and history function

Precondition:

1) Entering more than one operator between number inputs will result in an error.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step Number** | **Action** | **System Response** | **Requirement Tested**  **(if applicable)** |
| 1 | User enters the first digit 500+7059.3 | 50+7059.3 is outputted to the display bar |  |
| 2 | User enters equals button | The result 7559.3 is outputted to the display bar and the calculation is sent to the history log |  |
| 3 | User takes the result % 25 | 7559.3%25 is displayed |  |
| 4 | User enters the equals button | The result outputted is 9.300000000000182 and the calculation is sent to log | Showing if addition works with integers |
| 5 | User hits the clear button | The display bar has not output |  |
| 6 | User enters 70.3+= by mistake | Error is displayed and not sent to history log | Invalid operations do not work |
| 7 | User hits clear then enters -10000 + 50600 = | 40600 is displayed |  |
| 8 | User enters %17= | 4 is displayed and the calculation and result are sent to the log |  |
| 9 | User hits the history button | A new window with the previous calculations is shown:  500+7059.3 = 7559.3  7559.3%25 = 9.300000000000182  -10000+50600 = 40600  40600%17 = 4 | History log should display all successful calculations |

## Test Case 2

Description: Testing hexadecimal conversion, binary conversion, and clear history

Precondition:

1) To use a bin or hex button the user must enter the number they want to convert first.

2) If the history log is open and the user wants to clear the log, they must close the window and re-open the history.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step Number** | **Action** | **System Response** | **Requirement Tested**  **(if applicable)** |
| 1 | User enter 453 | 453 is outputted to the display bar |  |
| 2 | User hits hex button | 1c5 is outputted to the display bar and sent to the history |  |
| 3 | User clears the input and enters 5011 | 5011 is displayed |  |
| 4 | User selects the Bin button | 1001110010011 is displayed and appended to the history log |  |
| 5 | User hits the history button to look at their past calculations | The history opens in a new window with the following:  453 in hexadecimal: 1c5  5011 in binary: 1001110010011 |  |
| 6 | User closes the history log and hits the C-H button to clear the history | The history log is cleared |  |

## Test Case 3

Description: Testing trigonometric and log functions

Precondition:

1. To use a trig or log button the user must enter the number they want to convert first.
2. Trig functions only work for numbers in radians.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step Number** | **Action** | **System Response** | **Requirement Tested**  **(if applicable)** |
| 1 | User enters 50000 | 50000 is displayed |  |
| 2 | The user hits the sin button to convert the sin | -0.9998401890897896 is displayed and the calculation is sent to the history log |  |
| 3 | The user hits clear and enters 210 | 210 is displayed |  |
| 4 | The user hits the cos button | -0.8838 is displayed and the calculation is sent to the history log |  |
| 5 | The user enters 150 | 150 is displayed to the log |  |
| 6 | The user hits the log button | 2.1760912590556813 is displayed and sent to the history |  |
| 7 | The user hits the clear button and enters 9000 | The display is cleared and 9000 is displayed |  |
| 8 | The user hits the ln button to take the nautral log | 9.104979856318357 is outputted and sent to history |  |

# Verification Cross Reference Matrix

|  |  |
| --- | --- |
| **Requirement Identifier** | **Where Tested** |
| Arithmetic operators | History log |
| Binary and Hexadecimal conversions | History log |
| Updating and clearing the history log | History log |