Arithmetic Expression Evaluator

Document

Version 1.0.3

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/28/2023 | 1.0 | Creation of document | Charlie Gillund |
| 11/30/2023 | 1.01 | Updated parts 2,3,4,5 | Charlie Gillund |
| 12/3/2023 | 1.02 | Updated Output Specification Test Cases | Drew Meyer |
| 12/3/2023 | 1.03 | Added results of test cases and fixed errors within document | Charlie Gillund |

Table of Contents

1. Purpose 4

2. Test case identifier 4

3. Test item 4

4. Input specifications 4

5. Output specifications 4

6. Environmental needs 4

6.1.1 Hardware 4

6.1.2 Software 4

6.1.3 Other 4

7. Special procedural requirements 5

8. Intercase dependencies 5

# Purpose

The purpose of the Test Case Document is to provide an overview of the test cases used to evaluate the effectiveness of the Arithmetic Expression Evaluator. These Test cases are designed to evaluate the functionality of each facet of the program.

# Test case identifier

There are three different test case identifiers for the following tests. The first identifier (BF) stands for basic function and tests a basic function based on the requirements document. These are all tested using the eval function. The second identifier is (ER) which stands for an error test case to ensure that errors are handled correctly and produce the right output according to the requirements document. The third identifier(M) is the menu functions such as help, history, and exit.

# Test items

The items we are testing are the basic functions specified in the requirements document. (addition, subtraction, multiplication, division, modulo, etc.) and the basic error cases also in the requirements document (Invalid operator, invalid parentheses, Invalid expression, etc.). We also test the menu functions which hare the help function, exit function, and history function.

# Input specifications

There are two different inputs. The first type is the basic menu functions which include help, exit, and history which brings up the help menu, exits the program, and brings up the history respectfully. The second type is the expressions which requires the keyword eval followed by a valid expression.

# Output specifications

The expected outputs and the test outputs are in the table below. The outputs for the menu functions is the menu or screen they output or the function of actually closing the program. The output for expressions should be a number or an explanation of an error.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Input | Expected Output | Actual Output | Pass/Fail |
| M-01 | Tests the help menu funciton | help | “The Help Menu” | “The Help Menu | Pass |
| M-02 | Tests the history menu function | history | “The History Menu” | “The History Menu” | Pass |
| M-03 | Tests the exit menu function | exit | “Exits Program” | “Exits Program” | Pass |
| BF-01 | Test Addition | eval 3+4 | 7.00 | 7.00 | Pass |
| BF-02 | Test Subtraction | eval 4-1 | 3.00 | 3.00 | Pass |
| BF-03 | Test Multiplication | eval 5\*3 | 15.00 | 15.00 | Pass |
| BF-04 | Test Exponentiation | eval 2^4 | 16.00 | 16.00 | Pass |
| BF-05 | Test Exponentiation(alt) | eval 3\*\*3 | 27.00 | 27.00 | Pass |
| BF-06 | Test Modulo | eval 12%10 | 2.00 | 2.00 | Pass |
| BF-07 | Test Negative Numbers | eval -3+9 | 6.00 | 6.00 | Pass |
| BF-08 | Test Division | eval 9/3 | 3.00 | 3.00 | Pass |
| BF-09 | Test Parentheses / Implicit Multiplication | eval 9(3\*3) | 81.00 | 81.00 | Pass |
| BF-10 | Test Order of Operation | eval 3/3\*(2+1) | 3.00 | 3.00 | Pass |
| BF-11 | Precision and Rounding | eval 10 / 3 | 3.33 | 3.33 | Pass |
| BF-12 | Associativity | eval 2^3^2 | 512.00 | 512.00 | Pass |
| BF-13 | Zero Exponent | eval 2^0 | 1.00 | 1.00 | Pass |
| BF-14 | Negative Exponent | eval 2^-2 | 0.25 | 0.25 | Pass |
| BF-15 | Exponentiation with parentheses | eval 2^(3+2) | 32.00 | 32.00 | Pass |
| BF-16 | Modulo with Negative Numbers | eval -2%3 | -2.00 | -2.00 | Pass |
| BF-17 | Nested Parentheses | eval ((2+3)\*2) | 10.00 | 10.00 | Pass |
| BF-18 | Fractional Exponent | eval 4^(1/2) | 2.00 | 2.00 | Pass |
| ER-01 | Division by Zero | eval 9/0 | Error: Division by zero encountered in expression | Error: Division by zero encountered in expression | Pass |
| ER-02 | Modulo by Zero | eval 3%0 | Error: Modulo by zero encountered in expression | Error: Modulo by zero encountered in expression | Pass |
| ER-03 | Invalid Operator | eval 3$9 | Error: Unusable character '$' at position 1 | Error: Unusable character '$' at position 1 | Pass |
| ER-04 | Unmatched Parentheses (Left) | eval (9+3 | Error: Parentheses in expression are not balanced | Error: Parentheses in expression are not balanced | Pass |
| ER-05 | Unmatched Parenthesis (Right) | eval 9+3) | Error: Mismatched parentheses at position 3 | Error: Mismatched parentheses at position 3 | Pass |
| ER-06 | Fractional Exponent (Negative Base) | eval -4^(1/2) | Error: Expression generated NaN during evaluation | Error: Expression generated NaN during evaluation | Pass |
| ER-07 | Large Numbers and Overflow | eval 10^100 | Error: Exceeds floating point size limit (generated infinity during evaluation) | Error: Exceeds floating point size limit (generated infinity during evaluation) | Pass |
| ER-08 | Invalid Menu Function | string | Input was not understood, please try again | Input was not understood, please try again | Pass |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Environmental needs

### Hardware No special hardware is required.

### Software No outside software is needed to run the program.

### Other

# Special procedural requirements

No special procedural requirements.

# Intercase dependencies

There are no test cases which need another test case to be executed before the current test case.