QuickMath.io

QuickMath.io

Test Case

Version 1.0

Revision History

Date	Version	Description	Author
24/11/2023	0.1	Added test cases for stack implementation	Owen, David
26/11/2023	0.2	Added test cases for parser and inputting	Jamie, Vinny, Tatum
30/11/2023	0.3	Added other test cases	Omar, Tatum
01/12/2023	1.0	Finalized	Omar, David, Jamie, Vinny, Tatum, Omar

Table of Contents

1. Purpose 4

2. Test Cases 4

3. Environmental needs 6

Test Case

1. Purpose

The purpose of this Test Case Specification Document for the project QuickMath.io is to specify all of the testing that the QuickMath.io team used to check their product for correctness. Each test case has an ID, description, input, expected output, actual output, and whether or not the test passes.

Instead of multiple disconnected sections, we have opted for one, *Test Cases*, section.

2. Test Cases

Test Case ID	Description	Input	Expected Output	Actual Output	Pass?
SL-1	Testing single operator	"2 9 +"	2 9 +	2 9 +	yes
SL-2	Testing single operator (different from SL-1)	"3 4 *"	3 4 *	3 4 *	yes
SL-3	Testing single operator (different from SL-1, SL-2)	"9 3 %"	9 3 %	9 3 %	yes
SL-4	Testing single operator (different from SL-1, SL-2, SL-3)	"1 5 /"	1 5 /	1 5 /	yes
SL-5	Testing two operators at once	"2 3 4 + ^"	2 3 4 + ^	2 3 4 + ^	yes
SL-6	Testing three operators at once	"9 8 2 % * 6 +"	9 8 2 % * 6 +	9 8 2 % * 6 +	yes
SL-7	Testing multiple operators and operands.	"2 4 5 / 5 3 - 5 ^ 4 ^ * +"	2 4 5 / 5 3 - 5 ^ 4 ^ * +	2 4 5 / 5 3 - 5 ^ 4 ^ * +	yes
SL-8	Testing multiple operators and operands (different from SL-7)	"1 9 + 7 4 2 / * -"	1 9 + 7 4 2 / * -	1 9 + 7 4 2 / * -	yes
SL-9	Basic Arithmetic Operations	“3 + 4 * 2 / (1 – 5)”	3 4 2 * 1 5 - / +	3 4 2 * 1 5 - / +	yes
SL-10	Exponentiation and Modulus Operators	“2 ^ 3 % 3”	2 3 ^ 3 %	2 3 ^ 3 %	yes
SL-11	Floating Point Numbers	“3.5 + 2.4 * 7.2”	3.5 2.4 7.2 * +	3.5 2.4 7.2 * +	yes
SL-12	Multiple Parentheses	“(3 + (4 * 5))”	3 4 5 * +	3 4 5 * +	yes
SL-13	Complex Expression with Multiple Operators	“12 + 7 * (5 + 3 ^ 9 – 4) * 6 – 100"	12 7 5 3 9 ^ + 4 - * 6 * + 100 -	12 7 5 3 9 ^ + 4 - * 6 * + 100 -	yes
SL-14	No Spaces Between Numbers and Operators	“8+2*(3-1)”	8 2 3 1 - * +	8 2 3 1 - * +	yes
SL-15	Empty Expression	“ ”			yes
SL-16	Single Number	“42”	42	42	yes

SL-17	Testing for simple incomplete parentheses	“(5+2”	Error	Error	yes
SL-18	Testing for ** to ^ conversion	"5**3"	5^3	5^3	yes
SL-19	Testing for more complex incomplete parentheses	“((5+3) *6(5+4)”	Error	Error	yes
SL-20	Testing for valid parentheses	"2*(3+5)"	2*(3+5)"	2*(3+5)"	yes
SL-21	Testing for dangling minus sign	"-5 + 3"	0-5 + 3	0-5 +3	yes
SL-22	Testing two operations at once	"2*(3**2-4)-1"	2*(3^2-4)-1	2*(3^2-4)-1	yes
SL-23	Testing dangling minus with decimal	" -5.2 + 3"	0-5.2 + 3	0-5.2 +3	yes
SL-24	Testing for parentheses completion, conversion, and dangling minus	"2*-1(3**2-4)”	2*0-1(3^2-4)	2*0-1(3^2-4)	yes
SL-25	Testing for floating point division	7 5 /	1.4	1.4	yes
SL-26	Testing multiple operators	4 4 * 2 5 * -	6	6	yes
SL-27	Testing modulus operator	4 4 * 2 5 * %	6	6	yes
SL-28	Testing ^ operator	2 3 ^	8	8	yes
SL-29	Testing floating point input (stod)	0.25 5 *	1.25	1.25	yes
SL-30	Testing the calculator with addition	3 + 4	7	7	yes
SL-31	Testing the calculator with subtraction and parentheses	8 - (5 – 2)	5	5	yes
SL-32	Testing the calculator with multiplication and division	10 * 2 / 5	4	4	yes
SL-33	Testing the calculator with exponentiation	2 ^ 3	8	8	yes
SL-34	Testing the calculator with mixed operators	4 * (3 + 2) % 7 - 1	5	5	yes
SL-35	Testing the calculator with complex addition and extraneous parentheses	(((2 + 3))) + (((1 + 2)))	8	8	yes
SL-36	Testing the calculator with mixed operators with extraneous	((5 * 2) - ((3 / 1) + ((4 % 3))))	6	6	yes

	parentheses				
SL-37	Testing the calculator with nested parentheses with exponents	(((2 ^ (1 + 1)) + ((3 – 1) ^ 2)) / ((4 / 2) % 3))	4	4	yes
SL-38	Testing the calculator with with a combination of extraneous and necessary parentheses	(((((5 – 3))) * (((2 + 1))) + ((2 * 3))))	12	12	yes
SL-39	Testing the calculator with extraneous parentheses with division	((9 + 6)) / ((3 * 1) / (((2 + 2))) - 1)	-60	-60	yes
SL-40	Testing the calculator with combining unary operators with arithmetic operators	+(-2) * (-3) - ((-4) / (+5)	6.8	6.8	yes
SL-41	Testing the calculator with Unary negation and addition in parenthese	-(+1) + (+2)	1	1	yes
SL-42	Testing the calculator with negation and addition with negated parentheses	-(-(-3)) + (-4) + (+5)	-2	-2	yes
SL-43	Testing the calculator with unary negation and exponentiation	+2 ^ (-3)	0.125	0.125	yes
SL-44	Testing the calculator with combining unary operators with paenthesses	-(+2) * (+3) - (-4) / (-5)	-6.8	-6.8	yes

3. Environmental needs

The user needs to have git installed on their computer, as well as a command line interface. The application can be obtained from the, https://github.com/realVinayak/EECS348_Project/tree/main, repository.