<Company Name>

KI-69 Calculator

Test Case

Version <1.2>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>
12/02/2023	1.0	Added test cases and purpose	Jenna Luong
12/03/2023	1.1	Worked on more test cases	Nikka Vuong
12/03/2023	1.2	Completed test cases	Jenna Luong

Table of Contents

Test Case

1. Purpose

This Test Case Specification document for the KI-69 defines test cases for items that should be tested. The test cases are identified in section 2and the items and features are described in section 3. The input and output specifications will be defined in sections 4 and 5 respectively.

2. Test case identifier

Case	Expression
TC01	3 + 4
TC02	8 - (5 – 2)
TC03	10 * 2 / 5
TC04	2^3
TC05	4 * (3 + 2) % 7 - 1
TC06	(((2 + 3))) + (((1 + 2)))
TC07	((5 * 2) - ((3 / 1) + ((4 % 3))))
TC08	(((2 ^ (1 + 1)) + ((3 - 1) ^ 2)) / ((4 / 2) % 3))
TC09	(((((5 - 3))) * (((2 + 1))) + ((2 *3))))
TC10	((9 + 6)) / ((3 * 1) / (((2 + 2))) - 1)
TC11	+(-2) * (-3) - ((-4) / (+5))
TC12	-(+1) + (+2)
TC13	2 * (4 + 3 - 1
TC14	* 5 + 2
TC15	4 / 0
TC16	7 & 3
TC17	((7 * 3) @ 2)
TC18	-3 + 2
TC19	(((3 + 4) - 2) + (1)
TC20	((5 + 2) / (3 * 0))

3. Test item

Case	Items & Features

TC01	Testing a valid addition expression. It takes two numeric constants and adds them together.
TC02	Testing subtraction with parentheses expression. It subtracts a grouped subtraction expression of numeric constants from a numeric constant.
TC03	Testing multiplication and division. Demonstrates the calculator's use of PEMDAS on numeric constants. Multiplies then divides.
TC04	Testing exponentiation. Uses two numeric constants and raises one numeric constant by another.
TC05	Testing mixed operators. Demonstrates the calculator's use of PEMDAS. Solves expression in parentheses, then multiplies, then uses modulo, then subtracts.
TC06	Testing complex addition with extraneous parentheses. Makes sure that multiple sets of extraneous parentheses don't affect validity of expression. Addition is performed correctly.
TC07	Testing mixed operators with extraneous parentheses. This expression combines various operators with multiple sets of extraneous parentheses, but they do not change the order of operations or the final result.
TC08	Testing nested parentheses with exponents. This expression includes nested parentheses and exponentiation, creating complexity, but it adheres to the correct order of operations.
TC09	Testing combination of extraneous and necessary parentheses. This expression includes both extraneous parentheses and necessary parentheses to clarify the order of operations. It evaluates correctly.
TC10	Testing extraneous parentheses with division. Extraneous parentheses are added for clarity, but they do not affect the validity of the expression. The division, multiplication, and subtraction are performed correctly.
TC11	Testing combined unary operators with arithmetic operations. This expression combines unary + and - operators with multiplication, division, and addition.

TC12	Testing unary negation and addition in parentheses. Unary negation and addition operators are used within parentheses, followed by addition.
TC13	Testing unmatched parentheses. This expression has unmatched opening and closing parentheses, making it invalid.
TC14	Testing operators without operands. The * operator lacks operands on the left, making the expression invalid.
TC15	Testing incorrect operator usage. Division by zero is undefined in mathematics, so this expression is invalid.
TC16	Testing invalid characters. The & character is not a valid arithmetic operator, so this expression is invalid in the context of arithmetic operations.
TC17	Testing invalid characters. The @ character is not a valid arithmetic operator in this context, causing the expression to be invalid.
TC18	Testing negation. It negates an operand and adds it to another operand.
TC19	Testing mismatched parentheses. The parentheses are not properly matched, with one closing parenthesis missing, making the expression invalid.
TC20	Testing invalid operator usage. This expression attempts to divide by zero, which is mathematically undefined, rendering the expression invalid.

4. Input specifications

Case	Input	Values	Relationships
TC01	Expression	3, +, 4	+: adds 3 to 4
TC02	Expression	8, -, (, 5, -, 2,)	-: subtracts (5-2) from 8 (): gives precedence to the expression/value inside
TC03	Expression	10, *, 2, /, 5	*: multiplies 10 and 2 because of PEMDAS

			/: divides (10*2) by 5 since multiplication and division have the same priority, evaluate from left to right
TC04	Expression	2, ^, 3	^: brings 2 to the power of 3. multiplies 2 by itself 3 times
TC05	Expression	4, *, (, 3, +, 2,), %, 7, -, 1	(): gives precedence to expression inside *: multiplies 4 by 5 %: mods 20 by 7 -: subtracts 6 by 1
TC06	Expression	(, (, (, 2, +, 3,),) ,), +, (, (, (, 1, +, 2,),),)	(): gives precedence to expression inside (((2+3))): This is simply the sum of 2 and 3, which is 5. (((1+2))): This is the sum of 1 and 2, which is 3.
TC07	Expression	(, (, 5, *, 2,), -, (, (, 3, /, 1,), +, (, (, 4, %, 3,),),),)	(): gives precedence to expression inside. Solves inner most parentheses to the outer most parentheses 5*2: multiplies 5 by 2 3/1: divide 3 by 1 4%3: mods 4 by 3 +: adds the results from previous steps -: subtracts the results from last step to first step
TC08	Expression	(, (, 2, ^, (, 1, +, 1,),), +, (, (, 3, -, 1,), ^, 2,),), /, (, (, 4, /, 2,), %, 3,),)	(): gives precedence to expression inside. Solves inner most parentheses to the outer most parentheses 1+1: adds 1 and 1 2^(1+1): raises 2 to the power of 1+1. 3−1: subtracts 1 from 3 (3−1)^2: squares the result of 3-1

			((2(1+1))+((3−1)2)): adds the two results together 4/2: divides 4 by 2 (4/2)%3: mods 2 by 3 /: divides the two numeric constants left
TC09	Expression	(, (, (, (, (, 5, -, 3,),),), *, (, (, (, 2, +, 1,),),), +, (, (, 2, *, 3,),),),)	(): gives precedence to expression inside. Solves inner most parentheses to the outer most parentheses (5-3): subtracts the two numeric constants (2+1): adds the two numeric constants (2*3): multiplies the two numeric constants *: multiplies the 2 times 3 +: adds the two numeric constants left
TC10	Expression	(, (, 9, +, 6,),), /, (, (, 3, *, 1,), /, (, (, (, 2, +, 2,),),), -, 1,),	(): gives precedence to expression inside. Solves inner most parentheses to the outer most parentheses (9+6): adds the two numeric constants (2+2): adds the two numeric constants (3*1): multiplies the two numeric constants /: divides the two results from addition and multiplication -: subtracts 1 from the divided result by /: divides the two numeric constants left
TC11	Expression	+, (, -2,), *, (, -3,), -, (, (, -4,), /, (, +5,),)	+(-2): the positive of -2 is -2 -3: no change, as it's already negative

			(-2) * (-3): multiplies -2 and -3 resulting in a positive 6 (-4) / (+5): divides -4 by 5 resulting in a negative fraction of -4/5
TC12	Expression	-, (, +1,), +, (, +2,)	-(+1)-(+1): the negative of a positive number is negative, so it equals -1-1 +2+2: no change, as it's already positive
TC13	Invalid Expression	2, *, (, 4, +, 3, -, 1	4 + 3 – 1: valid expression 2 * (): valid expression (): unmatched opening and closing parentheses, therefore invalid
TC14	Invalid Expression	*, 5, +, 2	*5: lacks operands on the left, therefore invalid
TC15	Invalid Expression	4, /, 0,	4 / 0: division by zero is undefined, therefore invalid
TC16	Invalid Expression	7, &, 3	7 & 3: & not a valid arithmetic operator, therefore invalid
TC17	Invalid Expression	(, (, 7, *, 3,), @, 2,)	(): all parentheses are matching 7 * 3: valid expression @ 2: @ not a valid arithmetic operator, therefore invalid
TC18	Expression	-3, +, 2	-3: negates operand -3 + 2: adds operands together
TC19	Invalid Expression	(, (, (, 3, +, 4,), -, 2,), +, (, 1,)	(3+4): adds 3 and 4 (7-2): subtracts 7 and 2 (5 + (1): unbalanced parentheses, therefore invalid
TC20	Invalid Expression	(, (, 5, +, 2,), /, (, 3, *, 0,),)	(5+2): adds 5 and 2 (3*0): multiplies 3 and 0 (7/0): division by zero is undefined, therefore invalid

5. Output specifications

Case	Expected Output	Actual Output
TC01	7	7
TC02	5	5
TC03	4	4
TC04	8	8
TC05	5	5
TC06	8	8
TC07	6	6
TC08	4	4
TC09	12	12
TC10	-60	-60
TC11	6.8	Terminates program
TC12	1	Terminates program
TC13	Error, Invalid Expression	Terminates program
TC13 (fixed)	Error, Invalid Expression	Error, Invalid Expression
TC14	Error, Invalid Expression	Terminates program
TC14 (fixed)	Error, Invalid Expression	Error, Invalid Expression
TC15	Error, Invalid Expression	Error, Invalid Expression
TC16	Error, Invalid Expression	Error, Invalid Expression
TC17	Error, Invalid Expression	Error, Invalid Expression
TC18	-1	-1
TC19	Error, Invalid Expression	Error, Invalid Expression
TC20	Error, Invalid Expression	Error, Invalid Expression

6. Environmental needs

6.1.1 Hardware (nothing particular for the arithmetic expression project)

Specify the characteristics and configurations of the hardware required to execute this test

6.1.2 Software (nothing particular for the arithmetic expression project)

Specify the system and application software required to execute this test case. This may include system software such as operating systems, compilers, simulators, and test tools. In addition, the test item may interact with application software.

6.1.3 Other

N/A

7. Special procedural requirements

Describe any special constraints on the test procedures that execute this test case. These constraints may involve special set up, operator intervention, output determination procedures, and special wrap up.

8. Intercase dependencies

List the identifiers of test cases that must be executed prior to this test case. Summarize the nature of the dependencies.