

Test case ID	Test case description	Test Data	Expected results	Actual results	Pass/fail status
TC01	Adds two constants, resulting in a legal output.	$3 + 4$	7	7	Pass
TC02	Adds two constants, resulting in legal output.	$12 + 8$	20	20	Pass
TC03	Subtraction inside parenthesis is performed first, then the subtraction outside the parenthesis.	$8 - (5 - 2)$	5	5	Pass
TC04	Subtraction and multiplication without parentheses.	$20 - 4 * 2$	12	12	Pass
TC05	Correctly calculates the equation in PEMDAS order. Calculates division first, then addition.	$30 / 5 + 7$	13	13	Pass
TC06	Multiplication and division are performed from left to right.	$10 * 2 / 5$	4	4	Pass
TC07	Legal expression calculating 2 raised to the power of 3.	$2 ** 3$	8	8	Pass
TC08	Exponentiation with parentheses to alter precedence.	$(2 + 3) ** 2$	25	25	Pass
TC09	Nested parentheses with modulo operation.	$(15 \% 4) + (8 - 3)$	8	8	Pass
TC10	Combination of subtraction, multiplication, and addition.	$6 * 3 - 2 + 4$	20	20	Pass
TC11	Combines parenthesis and multiple operators to perform a legal expression.	$4 * (3 + 2) \% 7 - 1$	5	5	Pass
TC12	Complex nested parentheses with division.	$(40 / (5 + 3)) * 2$	10	10	Pass

TC13	Exponentiation combined with subtraction.	$3 ** 3 - 5$	22	22	Pass
TC14	Multiple operators with parentheses for clarity.	$(5 + 2) * 3 - (8 / 4)$	19	19	Pass
TC15	Expression using floating-point division.	$(10 / 4) + 2$	4.5	4.5	Pass
TC16	Multiple sets of parentheses should not affect the output. Addition is performed correctly.	$((2 + 3)) + ((1 + 2))$	8	8	Pass
TC17	Multiple sets of parentheses should not affect the output, and the operators are used correctly.	$((5 * 2) - ((3 / 1) + ((4 \% 3))))$	6	6	Pass
TC18	Nested parentheses are used to create complexity within the expression, but all syntax is valid, and it will run correctly.	$((2 ** (1 + 1)) + ((3 - 1) ** 2)) / ((4 / 2) \% 3)$	4	4	Pass
TC19	Extraneous parentheses and necessary parentheses are used to clarify order.	$(((((5 - 3))) * ((2 + 1))) + ((2 * 3)))$	12	12	Pass
TC20	Multiple parentheses are used for coherence regarding order. The division, multiplication, and subtraction are calculated correctly.	$((9 + 6)) / ((3 * 1) / (((2 + 2))) - 1)$	-60	-60	Pass
TC21	Expression combines two unary operators to confirm positive/negative with multiplication and division operators.	$+(-2) * (-3) - ((-4) / (+5))$	6.8	6.8	Pass
TC22	Unary operators' subtraction and addition are executed in	$-(+1) + (+2)$	1	1	Pass

	parentheses, concluding with addition.				
TC23	This uses nested unary negations and additions with a varying number of values negated and added.	$-(-(-3)) + (-4) + (+5)$	-2	-2	Pass
TC24	Fractional results are calculated using the unary operators +, - with exponentiation.	$+2 ** (-3)$	0.125	0.125	Pass
TC25	Unary operators are combined with parentheses along with other arithmetic expressions.	$-(+2) * (+3)$ $- (-4) / (-5)$	-6.8	-6.8	Pass
TC26	This expression has an opening parenthesis however, it has no closing parenthesis (unmatched parenthesis) making it invalid.	$2 * (4 + 3 - 1$	Invalid	Unmatched parenthesis	Pass
TC27	Unmatched opening parenthesis.	$(8 + 4 * 3$	Invalid	Unmatched parenthesis	Pass
TC28	Invalid character in the expression.	$6 + 3 @ 2$	Invalid	Unkown operator: @	Pass
TC29	Division by zero.	$10 / 0$	Invalid	Division by zero	Pass
TC30	Missing operator between numbers.	$9 3 + 2$	Invalid	Invalid expression	Pass
TC31	Mismatched parenthesis with an extra closing parenthesis.	$4 * (3 + 2))$	Invalid	Unmatched parenthesis	Pass
TC32	Improper use of exponentiation. Since there is no number after 2 it makes this statement invalid.	$2 ^$	Invalid	Unkown operator: ^	Pass
TC33	Multiple consecutive operators without a number.	$5 + * 3$	Invalid	Invalid operator usage	Pass
TC34	Invalid syntax with unary operator placement.	$- / 2 + 1$	Invalid	Invalid operator usage	

TC35	Incorrect order of operands and operators.	$*4 + 5$	Invalid	Invalid expression	Pass
TC36	The expression is invalid due to lack of operands prior to the * which makes it invalid.	$*5 + 2$	Invalid	Invalid Expression	Pass
TC37	Unsupported operator for exponentiation.	$2 ^ 3$	Invalid	Unkown operator: ^	
TC38	Mathematics does not allow division by 0 making this invalid.	$4 / 0$	Invalid	Division by zero	Pass
TC39	There is no operator between the 5 and (, making the expression invalid.	$5(2 + 3)$	Invalid	Invalid expression	Pass
TC40	The & character is not a valid arithmetic operator, making the expression invalid.	$7 \& 3$	Invalid	Unknown operator: &	Pass
TC41	This expression is invalid because the parentheses are mismatched. There are no closing parentheses after the (1).	$((3 + 4) - 2) + (1)$	Invalid	Unmatched parenthesis	Pass
TC42	Expression tries to divide by 0 which is not mathematically possible. Making this invalid.	$((5 + 2) / (3 * 0))$	Invalid	Division by zero	Pass
TC43	The expression contains a – operator but there is not a second operand to be evaluated, making the expression invalid.	$((2-) 1 + 3)$	Invalid	Invalid operator usage	Pass
TC44	The expression contains a – operator but there is no operand to be evaluated, making the expression invalid.	$((4 * 2) + (-))$	Invalid	Invalid operator usage	Pass

TC45	The ^ operator is not a valid arithmetic operator, making the expression invalid.	$((7 * 3) ^ 2)$	Invalid	Unknown operator: ^	Pass
TC 46	Expression contains an operator at the end making it invalid.	$5 + 3 -$	Invalid	Invalid expression	Pass
TC 47	Nested parentheses with an operator inside but no operands.	$(3+ ()) * 2$	Invalid	Unmatched parenthesis	Pass
TC48	Multiple operators being used together with no numbers in between.	$4 + /2$	Invalid	Invalid operator usage	Pass
TC49	Expression contains an operator at the end making it invalid.	$7 * 2 -$	Invalid	Invalid expression	Pass
TC50	The # operator is an unknown arithmetic operator causing the expression to be invalid.	$10 + 3 \% 2 \#$	Invalid	Unkown operator: #	Pass