

Variables / Estado inicial		result
TC-C-01		
Type:	Functional	
Title:	Validate initial value of result	
Objective:	Verify the contract's initial state	
Precondition:	Contract deployed	
Test Data:	N/A	
Steps:	Read the result variable	
Expected Result:	result = 10	
Priority:	High	
Tags:	#State #Initialization	
Function:		addition
Equivalence Partitions:		
EP1:	Both values > 0	
EP2:	One value = 0	
EP3:	Large valid values	
EP4:	Overflow (invalid)	
TC-C-02		(EP1)
Type:	EP	
Title:	addition with valid values	
Test Data:	num1 = 5, num2 = 5	
Steps:	Execute addition(5, 5)	
Returns:	Returns: 10	
Expected Result:	Event emitted: Addition(5, 5, 10)	
TC-C-03		(EP2)
Type:	EP	
Title:	addition with zero value	
Test Data:	num1 = 0, num2 = 7	
Steps:	Execute addition(0, 7)	
Returns:	Returns: 7	
Expected Result:		
TC-C-04		Positive
Type:	BVA	
Title:	Boundary Value Analysis (BVA)	
Test Data:	num1 = MAX_UINT256, num2 = 0	
Steps:	Execute addition(MAX_UINT256, 0)	
Expected Result:	Retorna: MAX_UINT256	
TC-C-05		Negative
Type:	Boundary Value Analysis (BVA)	
Title:	addition overflow	
Test Data:	num1 = MAX_UINT256, num2 = 1	
Steps:	Execute addition(MAX_UINT256, 1)	
Expected Result:	Transaction reverts automatically due to arithmetic overflow	
Function:		subtraction
Equivalence Partitions:		
EP1:	num1 > num2	
EP2:	num1 == num2	
EP3:	num1 < num2 (underflow)	
TC-C-06		(EP1)
Type:	EP	
Title:	Valid subtraction	
Test Data:	10, 3	
Steps:	Execute subtraction	
Returns:	Returns: 7	
Expected Result:	Event emitted: Subtraction(10, 3, 7)	
TC-C-07		(EP2)
Type:	EP	
Title:	subtraction with zero	
Test Data:	5, 5	
Expected Result:	Returns: 0	
TC-C-08		(EP3) Negative
Type:	EP	
Title:	Subtraction underflow	
Test Data:	3, 5	
Steps:	Execute subtraction(3, 5)	
Expected Result:	Transaction reverts automatically due to arithmetic underflow	

TC-C-09															
Type:	EP														
Title:	subtraction2 with negative values														
Test Data:	num1 = -10, num2 = 3														
Steps:	Execute subtraction2(-10, 3)														
Expected Result:	Returns: -13														
TC-C-10															
Type:	(BVA)														
Title:	BVA														
Test Data:	subtraction2 at lower int256 boundary														
Test Data:	num1 = MIN_INT256, num2 = 0														
Steps:	Execute subtraction2(MIN_INT256, 0)														
Expected Result:	Returns: MIN_INT256														
Function multiply(uint256)															
TC-C-11															
Type:	Functional / State														
Title:	multiply modifies contract state														
Test Data:	multiplier = 2														
Steps:	Execute multiply(2)														
Expected Result:	result = 20														
TC-C-12															
Type:	(BVA)														
Title:	Boundary Value Analysis (BVA)														
Test Data:	multiply by zero														
Test Data:	multiplier = 0														
Steps:	Execute multiply(0)														
Expected Result:	State updated: result = 0														
Function multiply2(uint256) – Modifier															
Equivalence Partitions:															
EP1:	num1 == 10 (valid)														
EP2:	num1 != 10 (invalid)														
EP3:	num1 == 0 (invalid)														
TC-C-13															
Type:	EP														
Title:	multiply2 with allowed value														
Test Data:	num1 = 10														
Steps:	Execute multiply2(10)														
Expected Result:	State updated: result = 100														
TC-C-14															
Type:	EP NEGATIVO														
Title:	multiply2 blocked by modifier														
Test Data:	num1 = 9														
Steps:	Execute multiply2(9)														
Expected Result:	Transaction reverts due to modifier restriction														