**Test Case 11**

* **Test Case ID**: TC\_CALC\_11
* **Test Description**: Verify addition of two large numbers
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as 999999999
  2. Enter the second number as 1
  3. Select the + operation
  4. Click the = button
* **Expected Result**: The result displayed is 1000000000

**Test Case 12**

* **Test Case ID**: TC\_CALC\_12
* **Test Description**: Verify subtraction of two large numbers resulting in a negative output
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as 500000
  2. Enter the second number as 700000
  3. Select the - operation
  4. Click the = button
* **Expected Result**: The result displayed is -200000

**Test Case 13**

* **Test Case ID**: TC\_CALC\_13
* **Test Description**: Verify multiplication resulting in zero
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as 0
  2. Enter the second number as 4567
  3. Select the \* operation
  4. Click the = button
* **Expected Result**: The result displayed is 0

**Test Case 14**

* **Test Case ID**: TC\_CALC\_14
* **Test Description**: Verify division result with decimal output
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as 7
  2. Enter the second number as 2
  3. Select the / operation
  4. Click the = button
* **Expected Result**: The result displayed is 3.5

**Test Case 15**

* **Test Case ID**: TC\_CALC\_15
* **Test Description**: Verify handling of special characters as input
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first input as @#%
  2. Enter the second number as 5
  3. Select the + operation
  4. Click the = button
* **Expected Result**: An error message is displayed saying "Invalid input"

**Test Case 16**

* **Test Case ID**: TC\_CALC\_16
* **Test Description**: Verify operation with both operands as zero
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as 0
  2. Enter the second number as 0
  3. Select the + operation
  4. Click the = button
* **Expected Result**: The result displayed is 0

**Test Case 17**

* **Test Case ID**: TC\_CALC\_17
* **Test Description**: Verify operation with negative decimal numbers
* **Preconditions**: Calculator application is running
* **Test Steps**:
  1. Enter the first number as -2.5
  2. Enter the second number as 4
  3. Select the \* operation
  4. Click the = button
* **Expected Result**: The result displayed is -10.0

**Test Case 18**

* **Test Case ID**: TC\_CALC\_18
* **Test Description**: Verify multiple operations sequence without clearing
* **Preconditions**: Calculator is running and retains state between operations
* **Test Steps**:
  1. Enter the first number as 2
  2. Enter the second number as 3
  3. Select the + operation
  4. Click the = button
  5. Enter 4
  6. Select the \* operation
  7. Click the = button
* **Expected Result**: First result is 5, second result is 20

**Test Case 19**

* **Test Case ID**: TC\_CALC\_19
* **Test Description**: Verify application behavior after invalid input correction
* **Preconditions**: Calculator is running
* **Test Steps**:
  1. Enter invalid input abc
  2. Application shows error "Invalid input"
  3. Enter valid number 5
  4. Enter second number 3
  5. Select + operation
  6. Click = button
* **Expected Result**: The result displayed is 8 after correcting input

**Test Case 20**

* **Test Case ID**: TC\_CALC\_20
* **Test Description**: Verify calculator clears input after result displayed
* **Preconditions**: Calculator is running and has a clear/reset functionality
* **Test Steps**:
  1. Enter 5
  2. Enter 3
  3. Select + operation
  4. Click = button
  5. Click Clear button
  6. Check input fields
* **Expected Result**: All input fields are empty after clearing