**Explanation of Test Types:**

* **describe Blocks:** Group related tests using describe blocks. This provides better organization and readability to your test suite.
* **it Blocks:** Each individual test case is defined within an it block.
* **Assertions with expect:** Chai's expect assertion library is used to assert the expected outcomes of the functions.
  + **expect(actualValue).to.equal(expectedValue):** Basic equality check.
  + **expect(actualValue).to.be.above(expectedValue):** Check if actualValue is greater than expectedValue.
  + **expect(actualValue).to.be.below(expectedValue):** Check if actualValue is less than expectedValue.
  + **expect(actualValue).to.throw(Error, 'error message'):** Check if a function throws an error with a specific message.
  + **expect(actualValue).to.be.an('object'):** Check if actualValue is an object.
  + **expect(actualValue).to.have.property('propertyName'):** Check if an object has a specific property.
  + **expect(actualValue).to.be.a('string'):** Check if actualValue is a string.

**Running the Tests:**

1. **Install Mocha and Chai:**

>>> npm install mocha chai --save-dev

1. **Create a mocha.opts File (Optional):**
   * In your project root directory, create a file named mocha.opts and add the following line:

>>> --require chai/register

1. This registers Chai's assertion library with Mocha.
2. **Run Tests:**

>>> mocha

This will execute the tests defined in calculator.test.js.

**Key Points:**

* **Test Organization:** The use of describe blocks helps organize tests into logical groups.
* **Assertions:** Chai's assertions provide a clear and expressive way to define expected outcomes for your tests.
* **Error Handling:** Testing for potential errors (like division by zero) ensures your code is robust.
* **Test Coverage:** Aim to cover various scenarios and edge cases within your tests to ensure the reliability of your functions.