**TestFixture & Test**  
  
Follow the steps listed below to write the NUnit test cases for the application.

* Create a Unit test project(.Net Framework) in the solution provided.
* Add the CalcLibrary project as reference
* Create a class “CalculatorTests” to write all the test cases for the methods in the solution
* Use the ‘TestFixture’, ‘SetUp’ and ‘TearDown’ attributes, to declare, initialize and cleanup activities respectively
* Create a Test method to check the addition functionality
* Use the ‘TestCase’ attribute to send the inputs and the expected result
* Use Assert.That to check the actual and expected result match

**ANSWER :**

**Calculator.cs :**

namespace CalcLibrary {

public class Calculator {

public int Add(int a, int b) {

return a + b;

}

}

}

**CalculatorTests.cs :**

using NUnit.Framework;

using CalcLibrary;

namespace CalcLibraryTests {

[TestFixture]

public class CalculatorTests {

private Calculator calculator;

[SetUp]

public void SetUp() {

calculator = new Calculator();

}

[TearDown]

public void TearDown() {

calculator = null;

}

[Test]

[TestCase(2, 3, 5)]

[TestCase(10, 20, 30)]

[TestCase(-1, 1, 0)]

public void Add\_WhenCalled\_ReturnsCorrectSum(int a, int b, int expected) {

int result = calculator.Add(a, b);

Assert.That(result, Is.EqualTo(expected));

}

[Test]

[Ignore("This test is currently ignored")]

public void IgnoredTest() {

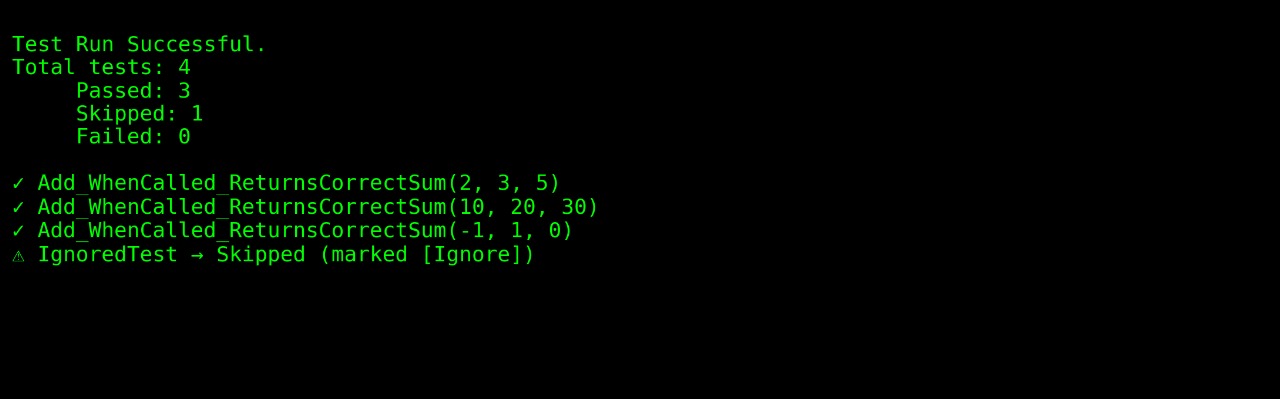
Assert.Fail("This test should be ignored.");

}

}

}

**OUTPUT**

****