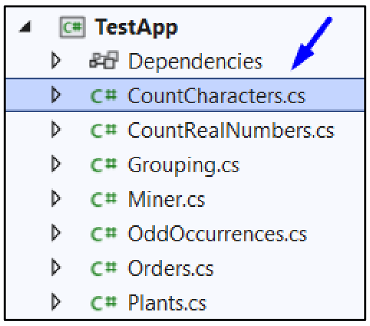
Exercises: Unit Testing Dictionaries

Test your tasks in the Judge system: https://judge.softuni.org/Contests/4474

1. Unit Test: Count Characters

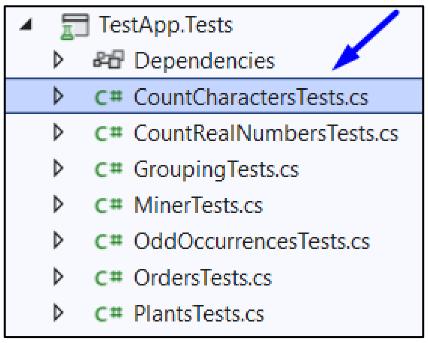
Look at the **provided skeleton** and examine the **`CountCharacters.cs`** class that you will test:



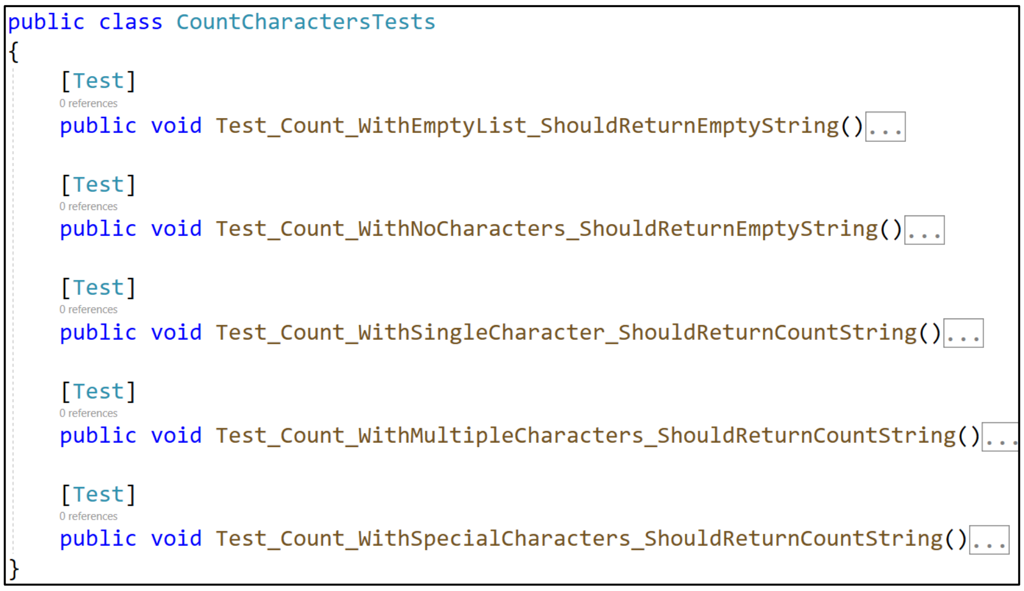
```
TestApp
  Dependencies
  CountCharacters.cs
  CountRealNumbers.cs
  Grouping.cs
  Miner.cs
  OddOccurrences.cs
  Orders.cs
  Plants.cs
```

The method takes in a **list of strings**, and collects the **number of times a character has appeared** and returns a string representing that information.

Then, look at the tests inside the **`CountCharactersTests.cs`** class

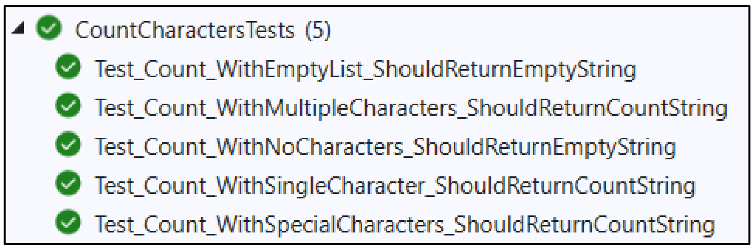


```
TestApp.Tests
  Dependencies
  CountCharactersTests.cs
  CountRealNumbersTests.cs
  GroupingTests.cs
  MinerTests.cs
  OddOccurrencesTests.cs
  OrdersTests.cs
  PlantsTests.cs
```



```
public class CountCharactersTests
{
    [Test]
    public void Test_Count_WithEmptyList_ShouldReturnEmptyString()...

    [Test]
    public void Test_Count_WithNoCharacters_ShouldReturnEmptyString()...

    [Test]
    public void Test_Count_WithSingleCharacter_ShouldReturnCountString()...

    [Test]
    public void Test_Count_WithMultipleCharacters_ShouldReturnCountString()...

    [Test]
    public void Test_Count_WithSpecialCharacters_ShouldReturnCountString()...
}
```

The first test if **finished** so you have a **reference**, **one** is finished **partially**, the rest of the tests are **empty,** and your task is to finish them. The tests should run when you're finished:



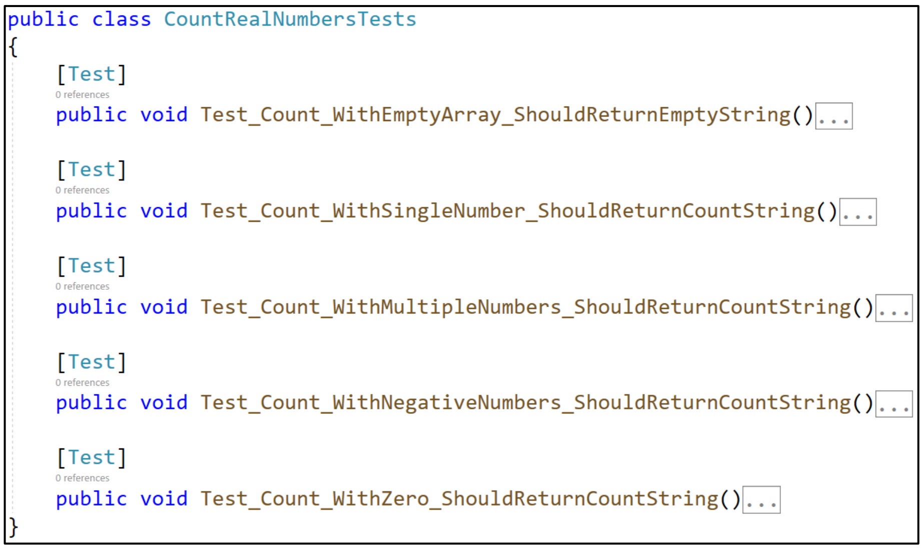
```
◢ ✓ CountCharactersTests (5)
    ✓ Test_Count_WithEmptyList_ShouldReturnEmptyString
    ✓ Test_Count_WithMultipleCharacters_ShouldReturnCountString
    ✓ Test_Count_WithNoCharacters_ShouldReturnEmptyString
    ✓ Test_Count_WithSingleCharacter_ShouldReturnCountString
    ✓ Test_Count_WithSpecialCharacters_ShouldReturnCountString
```

2. Unit Test: Count Real Numbers

Test a given method which takes in an **array of integers** and **counts** how many **times** each **number** was seen.

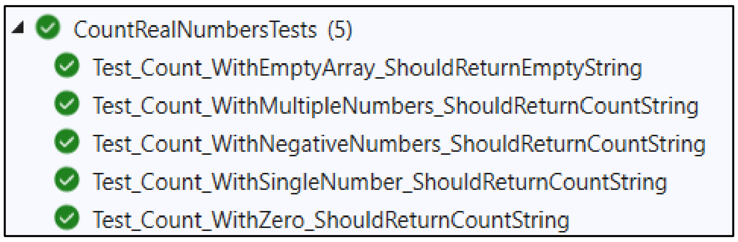
The method is found in the **CountRealNumbers.cs** file:

You are given a **test file CountRealNumbresTests.cs** which contains **5 tests**. **One** of them has been **finished partially**, and **four** are **empty** for you to finish:



```
public class CountRealNumbersTests
{
    [Test]
    0 references
    public void Test_Count_WithEmptyArray_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_Count_WithSingleNumber_ShouldReturnCountString()...

    [Test]
    0 references
    public void Test_Count_WithMultipleNumbers_ShouldReturnCountString()...

    [Test]
    0 references
    public void Test_Count_WithNegativeNumbers_ShouldReturnCountString()...

    [Test]
    0 references
    public void Test_Count_WithZero_ShouldReturnCountString()...
}
```

When you are ready make sure your **tests run:**



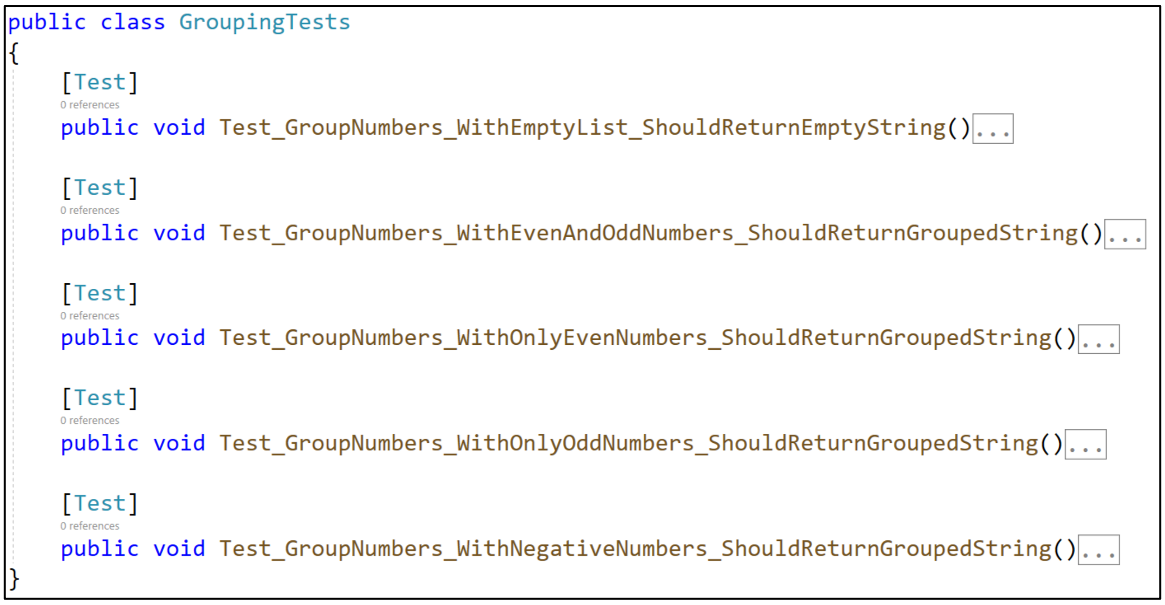
```
◢ ✓ CountRealNumbersTests (5)
    ✓ Test_Count_WithEmptyArray_ShouldReturnEmptyString
    ✓ Test_Count_WithMultipleNumbers_ShouldReturnCountString
    ✓ Test_Count_WithNegativeNumbers_ShouldReturnCountString
    ✓ Test_Count_WithSingleNumber_ShouldReturnCountString
    ✓ Test_Count_WithZero_ShouldReturnCountString
```

3. Unit Test: Grouping

Test a given method which takes in a **list of integers** and **groups** them by **even** and **odd** numbers.

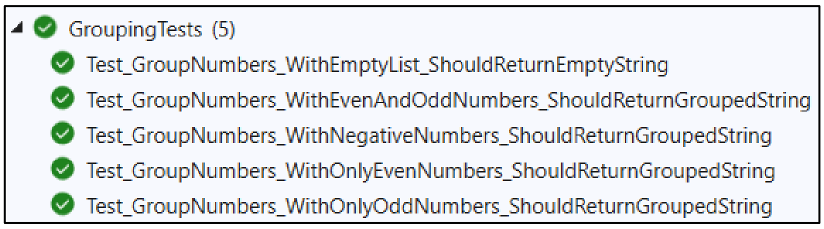
The method is found in the **Grouping.cs** file:

You are given a **test file GroupingTests.cs** which contains **5 tests**. **One** of them has been **finished partially**, and **four** are **empty** for you to finish:



```
public class GroupingTests
{
    [Test]
    0 references
    public void Test_GroupNumbers_WithEmptyList_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_GroupNumbers_WithEvenAndOddNumbers_ShouldReturnGroupedString()...

    [Test]
    0 references
    public void Test_GroupNumbers_WithOnlyEvenNumbers_ShouldReturnGroupedString()...

    [Test]
    0 references
    public void Test_GroupNumbers_WithOnlyOddNumbers_ShouldReturnGroupedString()...

    [Test]
    0 references
    public void Test_GroupNumbers_WithNegativeNumbers_ShouldReturnGroupedString()...
}
```

When you are ready make sure your **tests run:**



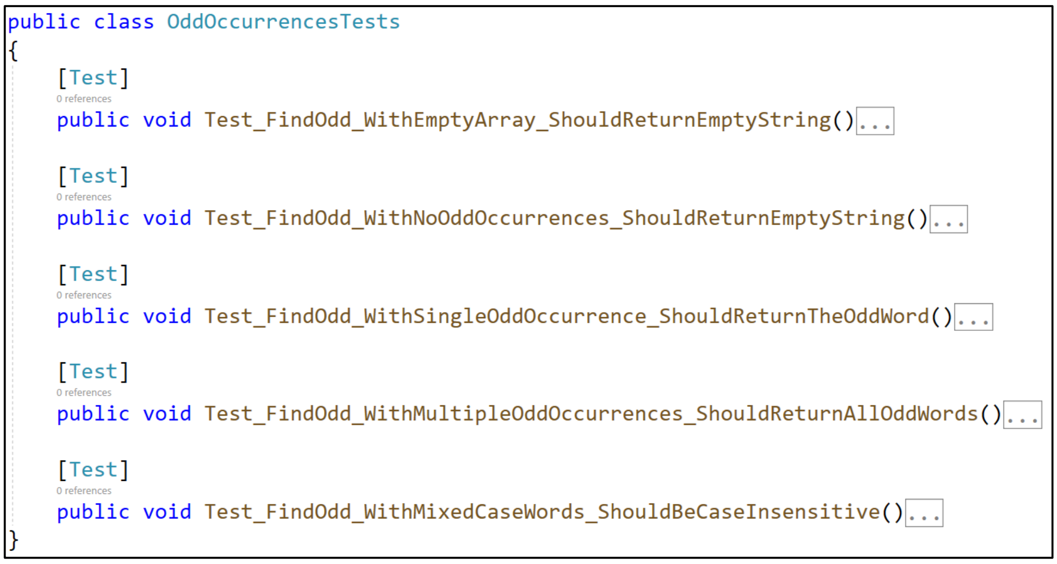
```
◢ ✓ GroupingTests (5)
    ✓ Test_GroupNumbers_WithEmptyList_ShouldReturnEmptyString
    ✓ Test_GroupNumbers_WithEvenAndOddNumbers_ShouldReturnGroupedString
    ✓ Test_GroupNumbers_WithNegativeNumbers_ShouldReturnGroupedString
    ✓ Test_GroupNumbers_WithOnlyEvenNumbers_ShouldReturnGroupedString
    ✓ Test_GroupNumbers_WithOnlyOddNumbers_ShouldReturnGroupedString
```

4. Unit Test: Odd Occurrences

Test a given method which takes in an **array of strings** and finds which **words appear** an **odd number** of times.

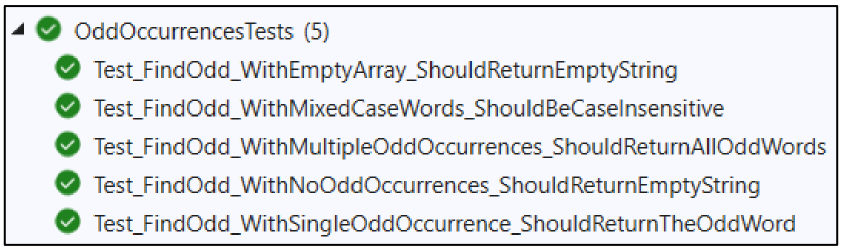
The method is found in the **`OddOccurrences.cs`** file:

You are given a **test file `OddOccurencesTests.cs`** which contains **5 tests**. **One** of them has been **finished partially**, and **four** are **empty** for you to finish:



```
public class OddOccurrencesTests
{
    [Test]
    0 references
    public void Test_FindOdd_WithEmptyArray_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_FindOdd_WithNoOddOccurrences_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_FindOdd_WithSingleOddOccurrence_ShouldReturnTheOddWord()...

    [Test]
    0 references
    public void Test_FindOdd_WithMultipleOddOccurrences_ShouldReturnAllOddWords()...

    [Test]
    0 references
    public void Test_FindOdd_WithMixedCaseWords_ShouldBeCaseInsensitive()...
}
```

When you are ready make sure your **tests run:**



```
◢ ✓ OddOccurrencesTests (5)
    ✓ Test_FindOdd_WithEmptyArray_ShouldReturnEmptyString
    ✓ Test_FindOdd_WithMixedCaseWords_ShouldBeCaseInsensitive
    ✓ Test_FindOdd_WithMultipleOddOccurrences_ShouldReturnAllOddWords
    ✓ Test_FindOdd_WithNoOddOccurrences_ShouldReturnEmptyString
    ✓ Test_FindOdd_WithSingleOddOccurrence_ShouldReturnTheOddWord
```

5. Unit Test: Miner

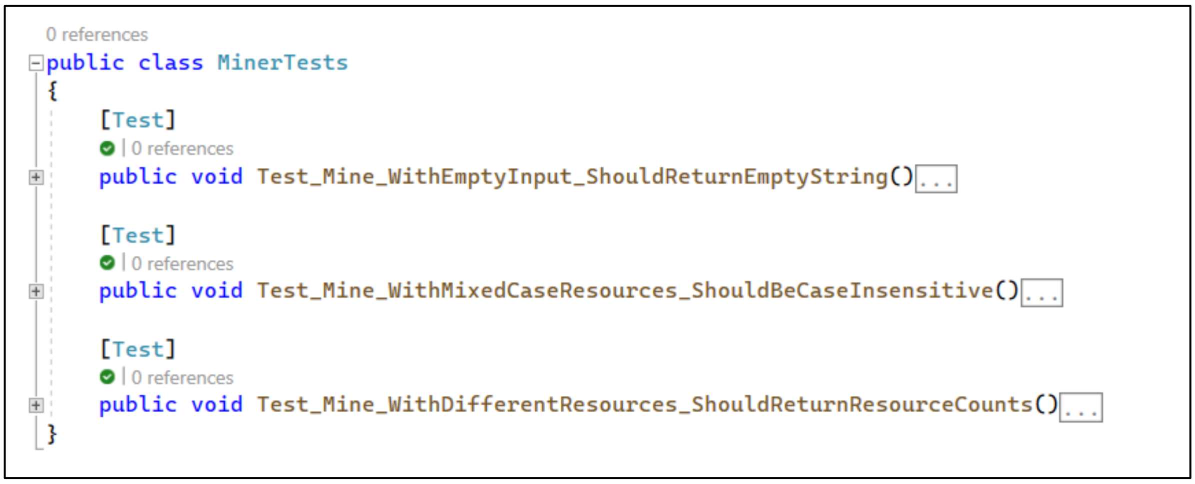
Test a given method which takes in **N number of strings** in the form of:

"**`{mineral} {quantity}`**"

Then it counts the **total quantity of a given mineral** and returns a string showing that.

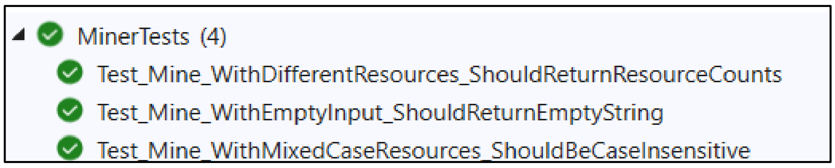
The method is found in the **`Miner.cs`** file

You are given a **test file `MinerTests.cs`** which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:



```
0 references
public class MinerTests
{
    [Test]
    ✓ | 0 references
    public void Test_Mine_WithEmptyInput_ShouldReturnEmptyString()...

    [Test]
    ✓ | 0 references
    public void Test_Mine_WithMixedCaseResources_ShouldBeCaseInsensitive()...

    [Test]
    ✓ | 0 references
    public void Test_Mine_WithDifferentResources_ShouldReturnResourceCounts()...
}
```

When you are ready make sure your **tests run:**



```
◢ ✓ MinerTests (4)
    ✓ Test_Mine_WithDifferentResources_ShouldReturnResourceCounts
    ✓ Test_Mine_WithEmptyInput_ShouldReturnEmptyString
    ✓ Test_Mine_WithMixedCaseResources_ShouldBeCaseInsensitive
```

6. Unit Test: Orders

Test a given method which takes in **N number of strings** in the form of:

"**`{product} {price} {quantity}`**"

It saves **each product** and **quantity** and **updates** the **price** each time it **changes**, and finally **calculates** the **total price** for each **product**.

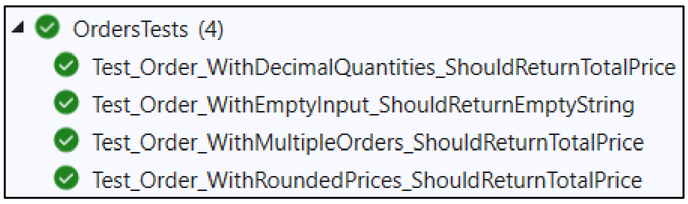
The method is found in the **`Orders.cs`** file

```
        StringBuilder sb = new();
        foreach (KeyValuePair<string, decimal[]> pair in products)
        {
            decimal totalPrice = pair.Value[1] * pair.Value[0];
            sb.AppendLine($"{pair.Key} -> {totalPrice:f2}");
        }

        return sb.ToString().Trim();
    }
}
```

You are given a **test file `OrdersTests.cs`** which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:

```
public class OrdersTests
{
    [Test]
    0 references
    public void Test_Order_WithEmptyInput_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_Order_WithMultipleOrders_ShouldReturnTotalPrice()...

    [Test]
    0 references
    public void Test_Order_WithRoundedPrices_ShouldReturnTotalPrice()...

    [Test]
    0 references
    public void Test_Order_WithDecimalQuantities_ShouldReturnTotalPrice()...
}
```

When you are ready make sure your **tests run:**



```
◢ ✓ OrdersTests (4)
    ✓ Test_Order_WithDecimalQuantities_ShouldReturnTotalPrice
    ✓ Test_Order_WithEmptyInput_ShouldReturnEmptyString
    ✓ Test_Order_WithMultipleOrders_ShouldReturnTotalPrice
    ✓ Test_Order_WithRoundedPrices_ShouldReturnTotalPrice
```

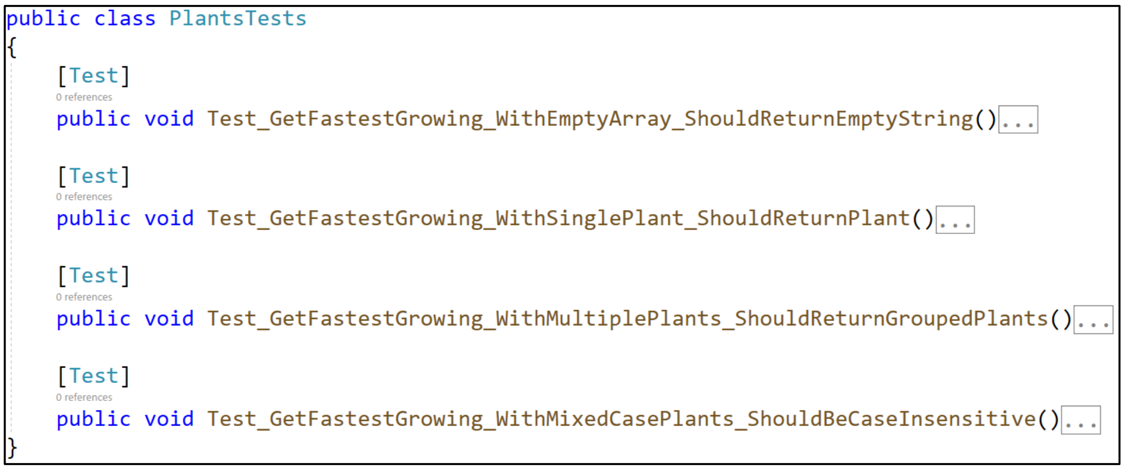
7. Unit Test: Plants

Test a given method which takes in an **array of strings** which saves and groups plants based on their number of letters, the shortest named plants will grow the **fastest**.

The method is found in the **`Plants.cs`** file

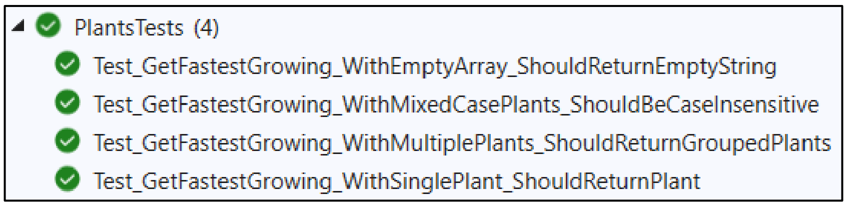
```
        StringBuilder sb = new();
        foreach (KeyValuePair<int, List<string>> kvp in groupedPlants.OrderBy(kv:KeyValuePair<int,List<…>> => kv.Key))
        {
            sb.AppendLine($"Plants with {kvp.Key} letters:");
            foreach (string plant in kvp.Value)
            {
                sb.AppendLine(plant);
            }
        }

        return sb.ToString().Trim();
    }
}
```

You are given a **test file `PlantsTests.cs`** which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:



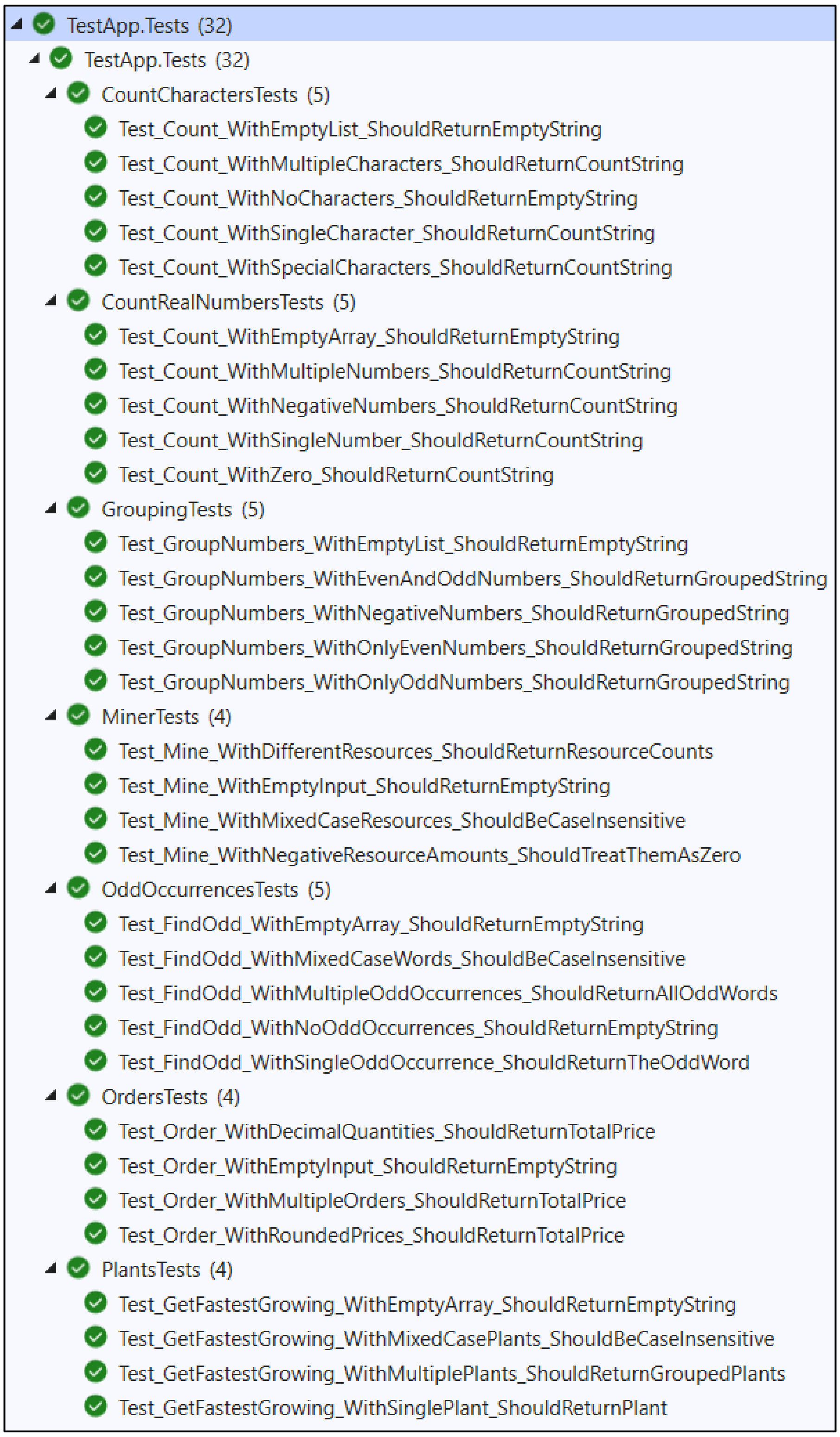
```
public class PlantsTests
{
    [Test]
    0 references
    public void Test_GetFastestGrowing_WithEmptyArray_ShouldReturnEmptyString()...

    [Test]
    0 references
    public void Test_GetFastestGrowing_WithSinglePlant_ShouldReturnPlant()...

    [Test]
    0 references
    public void Test_GetFastestGrowing_WithMultiplePlants_ShouldReturnGroupedPlants()...

    [Test]
    0 references
    public void Test_GetFastestGrowing_WithMixedCasePlants_ShouldBeCaseInsensitive()...
}
```

When you are ready make sure your **tests run:**



```
◢ ✓ PlantsTests (4)
    ✓ Test_GetFastestGrowing_WithEmptyArray_ShouldReturnEmptyString
    ✓ Test_GetFastestGrowing_WithMixedCasePlants_ShouldBeCaseInsensitive
    ✓ Test_GetFastestGrowing_WithMultiplePlants_ShouldReturnGroupedPlants
    ✓ Test_GetFastestGrowing_WithSinglePlant_ShouldReturnPlant
```

At the end make sure all tests pass:



```
◢ ✓ TestApp.Tests (32)
  ◢ ✓ TestApp.Tests (32)
    ◢ ✓ CountCharactersTests (5)
        ✓ Test_Count_WithEmptyList_ShouldReturnEmptyString
        ✓ Test_Count_WithMultipleCharacters_ShouldReturnCountString
        ✓ Test_Count_WithNoCharacters_ShouldReturnEmptyString
        ✓ Test_Count_WithSingleCharacter_ShouldReturnCountString
        ✓ Test_Count_WithSpecialCharacters_ShouldReturnCountString
    ◢ ✓ CountRealNumbersTests (5)
        ✓ Test_Count_WithEmptyArray_ShouldReturnEmptyString
        ✓ Test_Count_WithMultipleNumbers_ShouldReturnCountString
        ✓ Test_Count_WithNegativeNumbers_ShouldReturnCountString
        ✓ Test_Count_WithSingleNumber_ShouldReturnCountString
        ✓ Test_Count_WithZero_ShouldReturnCountString
    ◢ ✓ GroupingTests (5)
        ✓ Test_GroupNumbers_WithEmptyList_ShouldReturnEmptyString
        ✓ Test_GroupNumbers_WithEvenAndOddNumbers_ShouldReturnGroupedString
        ✓ Test_GroupNumbers_WithNegativeNumbers_ShouldReturnGroupedString
        ✓ Test_GroupNumbers_WithOnlyEvenNumbers_ShouldReturnGroupedString
        ✓ Test_GroupNumbers_WithOnlyOddNumbers_ShouldReturnGroupedString
    ◢ ✓ MinerTests (4)
        ✓ Test_Mine_WithDifferentResources_ShouldReturnResourceCounts
        ✓ Test_Mine_WithEmptyInput_ShouldReturnEmptyString
        ✓ Test_Mine_WithMixedCaseResources_ShouldBeCaseInsensitive
        ✓ Test_Mine_WithNegativeResourceAmounts_ShouldTreatThemAsZero
    ◢ ✓ OddOccurrencesTests (5)
        ✓ Test_FindOdd_WithEmptyArray_ShouldReturnEmptyString
        ✓ Test_FindOdd_WithMixedCaseWords_ShouldBeCaseInsensitive
        ✓ Test_FindOdd_WithMultipleOddOccurrences_ShouldReturnAllOddWords
        ✓ Test_FindOdd_WithNoOddOccurrences_ShouldReturnEmptyString
        ✓ Test_FindOdd_WithSingleOddOccurrence_ShouldReturnTheOddWord
    ◢ ✓ OrdersTests (4)
        ✓ Test_Order_WithDecimalQuantities_ShouldReturnTotalPrice
        ✓ Test_Order_WithEmptyInput_ShouldReturnEmptyString
        ✓ Test_Order_WithMultipleOrders_ShouldReturnTotalPrice
        ✓ Test_Order_WithRoundedPrices_ShouldReturnTotalPrice
    ◢ ✓ PlantsTests (4)
        ✓ Test_GetFastestGrowing_WithEmptyArray_ShouldReturnEmptyString
        ✓ Test_GetFastestGrowing_WithMixedCasePlants_ShouldBeCaseInsensitive
        ✓ Test_GetFastestGrowing_WithMultiplePlants_ShouldReturnGroupedPlants
        ✓ Test_GetFastestGrowing_WithSinglePlant_ShouldReturnPlant
```