DAY 18 ASSIGNMENT

-- BY SARATH KASIMSETTY

1) What is the use of XML .

- XML is used for universal data transfer mechanism to send data across to difference platform.
- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to store and arrange the data, which can customize your data handling needs. XML can easily be merged with style sheets to create almost any desired output. Virtually, any type of data can be expressed as an XML document.

2) Write the points discussed about xml in the class

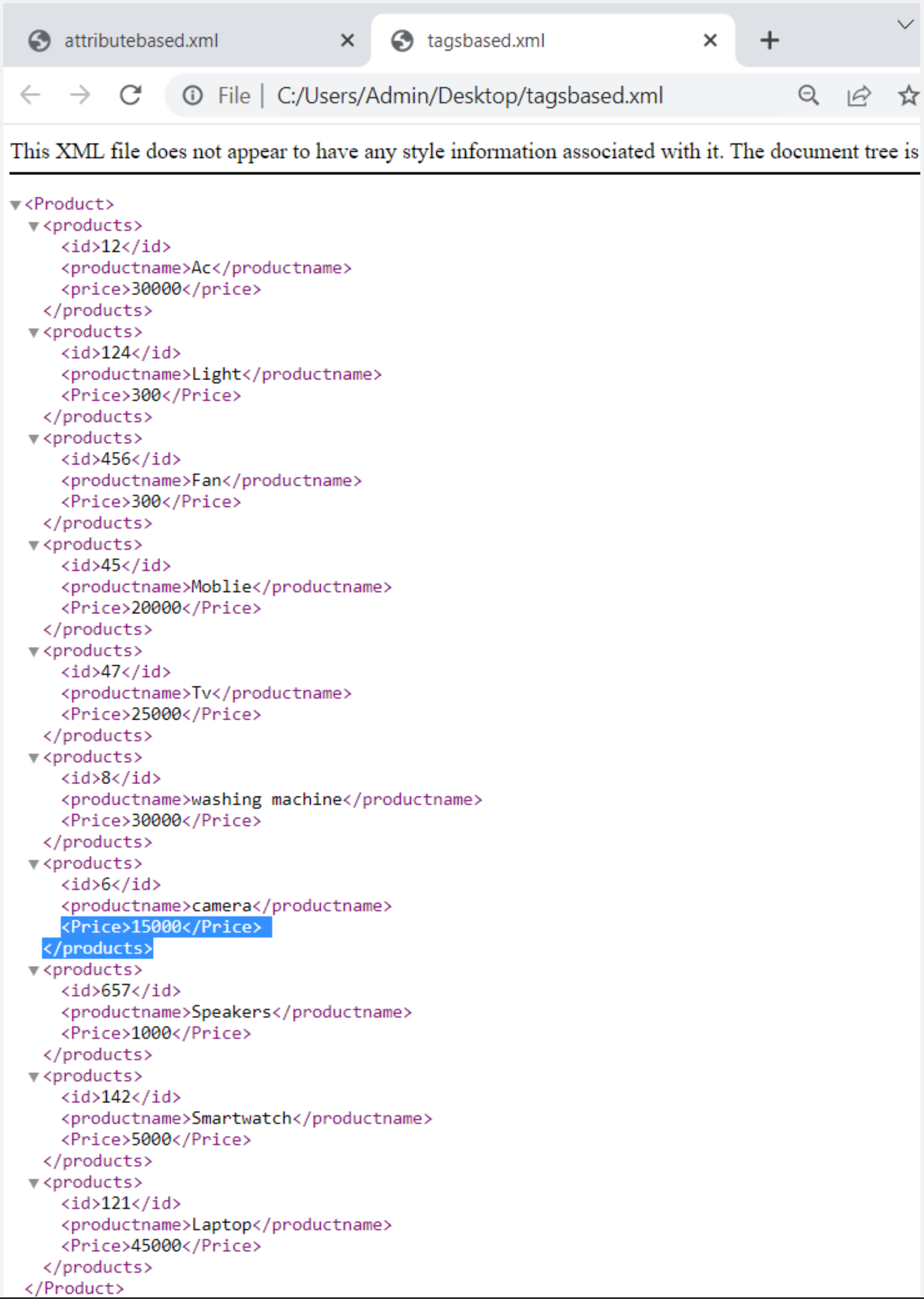
- XML is user for universal data transfer mechanism to send data across to difference platform.
- **XML** full form extensible Markup Language.
- **XML** is user-defined tags.
- **XML** can have only one root tag.
- XML is case sensitive.
- **TWO TYPES OF XML IS:**
  1. Tags based XML.
  2. Attribute based XML.

3) Create a simple xml to illustrate:

A) Tag based xml with 10 products

```
<Product>
  <products>
    <id>12</id>
    <productname>Ac</productname>
    <price>30000</price>
  </products>
  <products>
    <id>124</id>
    <productname>Light</productname>
    <Price>300</Price>
  </products>
  <products>
    <id>456</id>
    <productname>Fan</productname>
    <Price>300</Price>
  </products>
  <products>
    <id>45</id>
    <productname>Moblie</productname>
    <Price>20000</Price>
  </products>
  <products>
    <id>47</id>
    <productname>Tv</productname>
    <Price>25000</Price>
  </products>
  <products>
    <id>8</id>
    <productname>washing machine</productname>
    <Price>30000</Price>
  </products>
  <products>
    <id>6</id>
    <productname>camera</productname>
    <Price>15000</Price>
  </products>
  <products>
    <id>657</id>
    <productname>Speakers</productname>
    <Price>1000</Price>
  </products>
  <products>
    <id>142</id>
    <productname>Smartwatch</productname>
    <Price>5000</Price>
  </products>
  <products>
    <id>121</id>
    <productname>Laptop</productname>
    <Price>45000</Price>
  </products>
</Product>
```

OUTPUT:

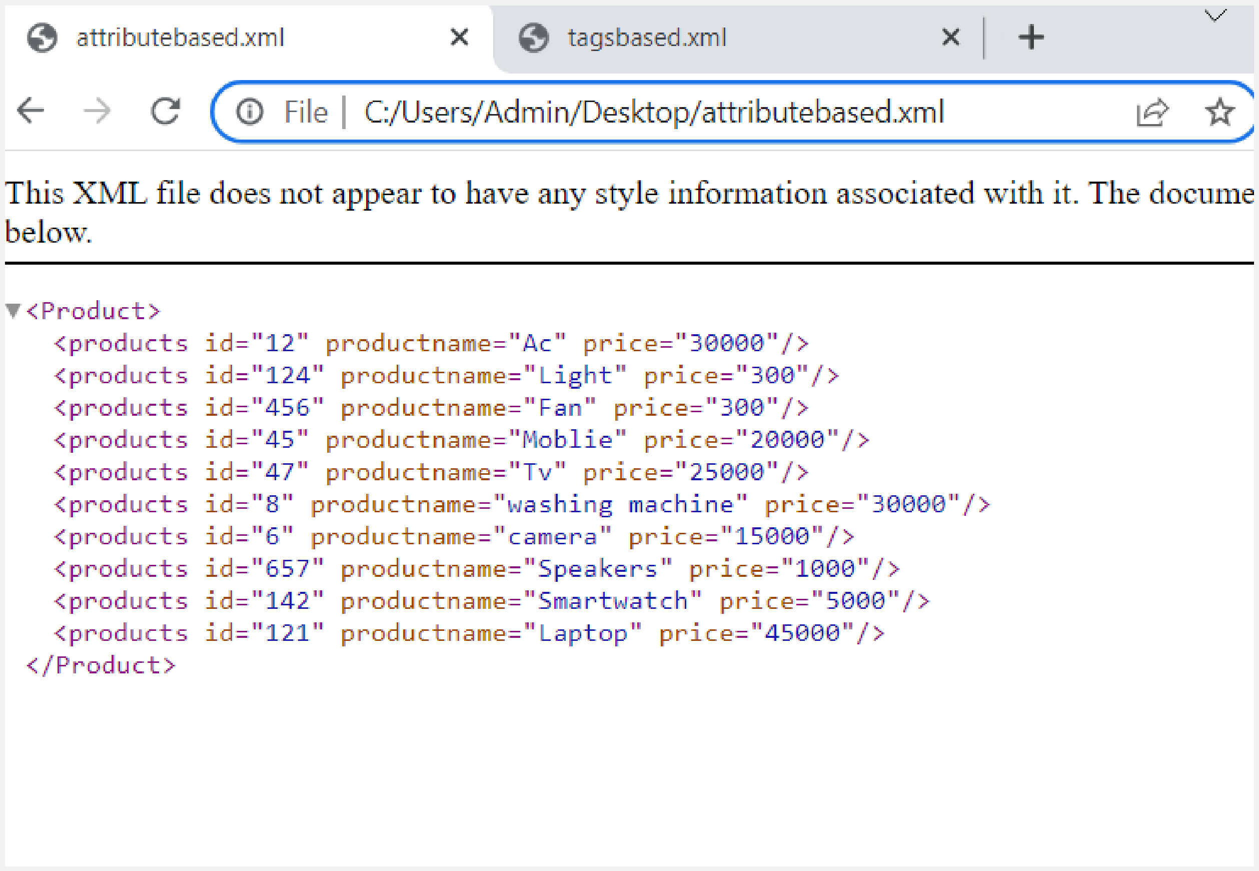


This XML file does not appear to have any style information associated with it. The document tree is

```
▼<Product>
  ▼<products>
     <id>12</id>
     <productname>Ac</productname>
     <price>30000</price>
   </products>
  ▼<products>
     <id>124</id>
     <productname>Light</productname>
     <Price>300</Price>
   </products>
  ▼<products>
     <id>456</id>
     <productname>Fan</productname>
     <Price>300</Price>
   </products>
  ▼<products>
     <id>45</id>
     <productname>Moblie</productname>
     <Price>20000</Price>
   </products>
  ▼<products>
     <id>47</id>
     <productname>Tv</productname>
     <Price>25000</Price>
   </products>
  ▼<products>
     <id>8</id>
     <productname>washing machine</productname>
     <Price>30000</Price>
   </products>
  ▼<products>
     <id>6</id>
     <productname>camera</productname>
     <Price>15000</Price>
   </products>
  ▼<products>
     <id>657</id>
     <productname>Speakers</productname>
     <Price>1000</Price>
   </products>
  ▼<products>
     <id>142</id>
     <productname>Smartwatch</productname>
     <Price>5000</Price>
   </products>
  ▼<products>
     <id>121</id>
     <productname>Laptop</productname>
     <Price>45000</Price>
   </products>
 </Product>
```

B) Attribute based xml with 10 products

```
<Product>
  <products id ="12" productname="Ac" price ="30000"></products>
  <products id ="124" productname="Light" price ="300"></products>
  <products id ="456" productname="Fan" price ="300"></products>
  <products id ="45" productname="Moblie" price ="20000"></products>
  <products id ="47" productname="Tv" price ="25000"></products>
  <products id ="8" productname="washing machine" price ="30000"></products>
  <products id ="6" productname="camera" price ="15000"></products>
  <products id ="657" productname="Speakers" price ="1000"></products>
  <products id ="142" productname="Smartwatch" price ="5000"></products>
  <products id ="121" productname="Laptop" price ="45000"></products>
</Product>
```



This XML file does not appear to have any style information associated with it. The docume
below.

```
▼<Product>
   <products id="12" productname="Ac" price="30000"/>
   <products id="124" productname="Light" price="300"/>
   <products id="456" productname="Fan" price="300"/>
   <products id="45" productname="Moblie" price="20000"/>
   <products id="47" productname="Tv" price="25000"/>
   <products id="8" productname="washing machine" price="30000"/>
   <products id="6" productname="camera" price="15000"/>
   <products id="657" productname="Speakers" price="1000"/>
   <products id="142" productname="Smartwatch" price="5000"/>
   <products id="121" productname="Laptop" price="45000"/>
 </Product>
```

4) Convert the above xml to JSON and display the JSON data

XML

```
<Product>
 <products id ="12" productname="Ac" price ="30000"></products>
 <products id ="124" productname="Light" price ="300"></products>
 <products id ="456" productname="Fan" price ="300"></products>
 <products id ="45" productname="Moblie" price ="20000"></products>
 <products id ="47" productname="Tv" price ="25000"></products>
 <products id ="8" productname="washing machine" price ="30000"></products>
 <products id ="6" productname="camera" price ="15000"></products>
 <products id ="657" productname="Speakers" price ="1000"></products>
 <products id ="142" productname="Smartwatch" price ="5000"></products>
 <products id ="121" productname="Laptop" price ="45000"></products>
</Product>
```

OUTPUT OF JSON

```
[
   {
      "@id": "12",
      "@productname": "Ac",
      "@price": "30000"
   },
   {
      "@id": "124",
      "@productname": "Light",
      "@price": "300"
   },
   {
      "@id": "456",
      "@productname": "Fan",
      "@price": "300"
   },
   {
      "@id": "45",
      "@productname": "Moblie",
      "@price": "20000"
   },
   {
      "@id": "47",
      "@productname": "Tv",
      "@price": "25000"
   },
   {
        "@id": "8",
        "@productname": "washing machine",
        "@price": "30000"
    },
    {
        "@id": "6",
        "@productname": "camera",
        "@price": "15000"
    },
    {
        "@id": "657",
        "@productname": "Speakers",
        "@price": "1000"
    },
    {
        "@id": "142",
        "@productname": "Smartwatch",
        "@price": "5000"
    },
    {
        "@id": "121",
        "@productname": "Laptop",
        "@price": "45000"
    }
]
```

5) Research and write the benefits of JSON over XML(2 or 3 points)

- Less Verbose, yet easy to debug (readable). Readability of data over the wire is very much important to debug with ease.

- More Speed & Less Bandwidth Utilization. Since the packet size is smaller in JSON compared to XML to convey the same...

- Less Memory Footprint, faster generation & processing. The data structure for JSON is smaller resulting in less memory.

6) For the below requirement, create a layered architecture project with seperate class library for Business logic.

- *** create console application**
- *** create windows(or desktop) application**

<u>Business Requirement:</u>

FIND FACTORIAL OF A NUMBER:

0 = 1

positive number (upto 7) = factorial answer

> 7 = -999 (as answer)

< 0 = -9999 (as answer)

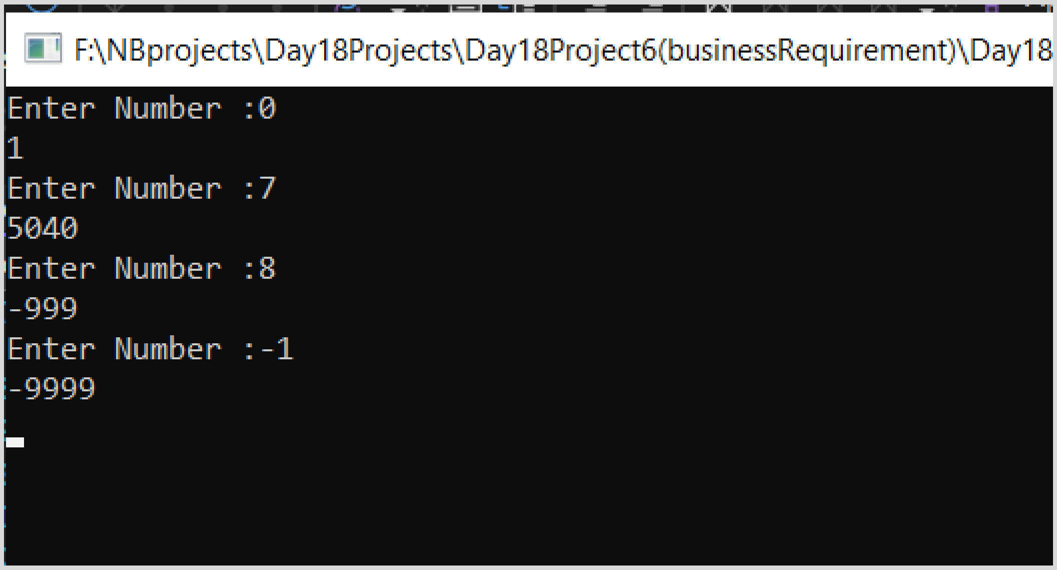
put the screen shots of the output and project (solution explorer) screenshot

Library of business logic :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//create a layered architecture
//project with seperate class library for Business logic.

namespace MathematicsLibrary
{
    public  class Algebra
    {
        /// <summary>
        /// Business Requriment
        /// </summary>
        /// <param name="n">Factorial Number</param>
        /// <returns>Factorial</returns>
        public static int Factorial(int n)
        {
            int fact = 1;
            if (n == 0)
                return 1;

            else if (n > 7)
                return -999;
            else if (n < 0)
                return -9999;
            else
            {
                for (int i = 1; i <= n; i++)
                     fact = fact * i;
                return fact;
            }
        }
    }
}
```

Client Application(console)

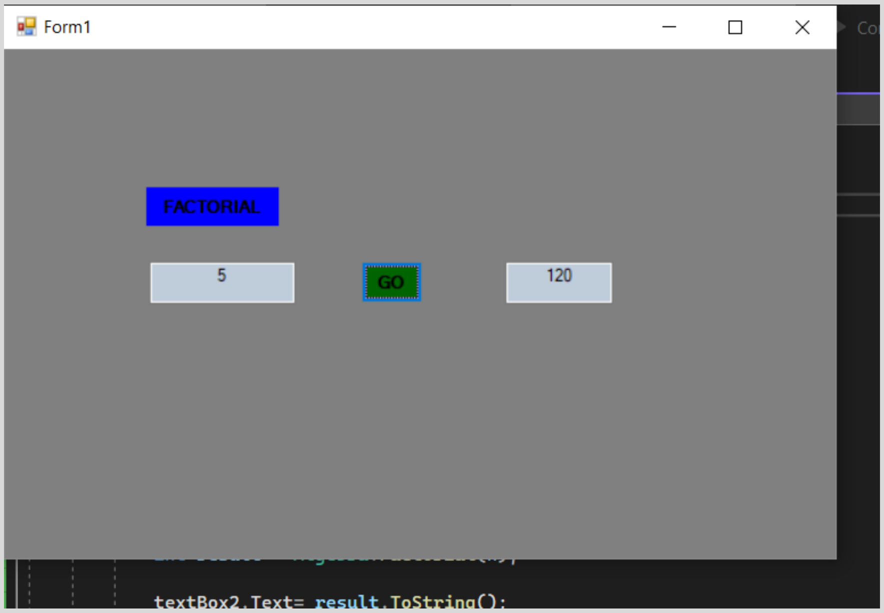
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathematicsLibrary;
/// <summary>
/// layered architectureproject with seperate class library for Business logic.
/// </summary>
namespace Day18Project6_businessRequirement_
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 4; i++)
            {
                //Read input from user
                Console.Write("Enter Number :");
                int n = Convert.ToInt32(Console.ReadLine());
                // Library class of business Requriment to print
                Console.WriteLine(Algebra.Factorial(n));
            }
            Console.ReadLine();
        }
    }
}
```


F:\NBprojects\Day18Projects\Day18Project6(businessRequirement)\Day18

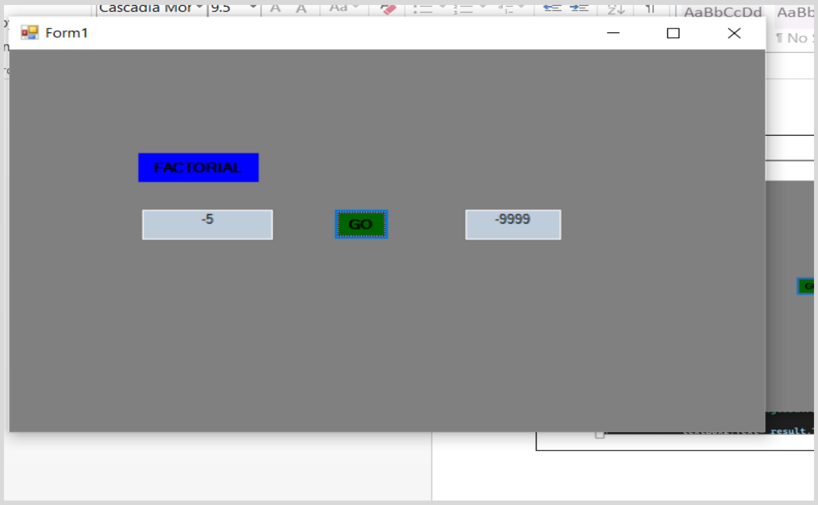
```
Enter Number :0
1
Enter Number :7
5040
Enter Number :8
-999
Enter Number :-1
-9999
```

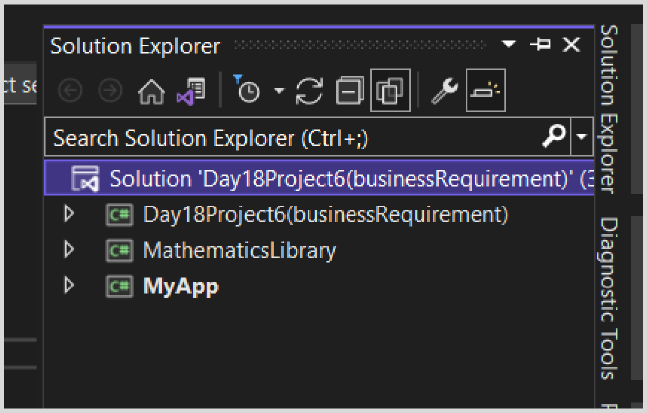
Windows(or desktop) application(client app)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MathematicsLibrary;
//ssarth kasimsetty
//Window applicatiom

namespace MyApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            int n = Convert.ToInt32(textBox1.Text);

            int result = Algebra.Factorial(n);

            textBox2.Text= result.ToString();
            Console.ReadLine();

        }
    }
}
```



```
textBox2.Text= result.ToString();
```



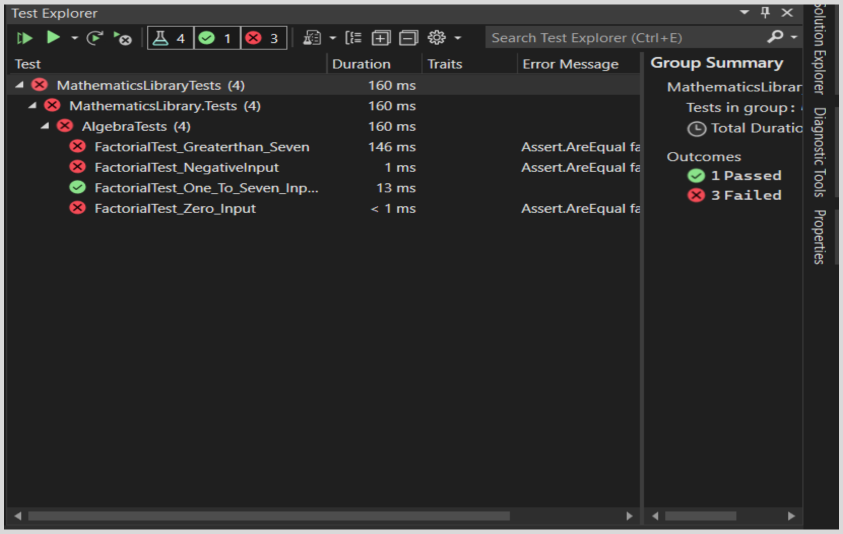

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Day18Project6(businessRequirement)' (3
Day18Project6(businessRequirement)
MathematicsLibrary
MyApp
Solution Explorer
Diagnostic Tools

7) For the above method, Implement TDD and write 4 test cases

CODE: FactorialTests.cs

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using MathematicsLibrary;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MathematicsLibrary.Tests
{
    [TestClass()]
    public class AlgebraTests
    {
        [TestMethod()]
        public void FactorialTest_Zero_Input()
        {
            //arrange
            int n = 0;
            int expected = 1;
            //act
            int actual = Algebra.Factorial(n);
            //assert
            Assert.AreEqual(expected, actual);
        }
        [TestMethod()]
        public void FactorialTest_Greaterthan_Seven()
        {
            //arrange
            int n = 8;
            int expected = -999;

            //act
            int actual = Algebra.Factorial(n);
            //assert
            Assert.AreEqual(expected,actual);
        }
        [TestMethod()]
        public void FactorialTest_NegativeInput()
        {
            //arrange
            int n = -1;
            int expected = -9999;

            //act
            int actual = Algebra.Factorial(n);
            //assert
            Assert.AreEqual(expected, actual);
        }
        [TestMethod()]
        public void FactorialTest_One_To_Seven_Input()
        {
            //arrange
            int n = 5;
            int expected = 120;
            //act
            int actual = Algebra.Factorial(n);
            //assert
            Assert.AreEqual(expected, actual);
        }

    }
}
```

The screen shot of all test cases failing.



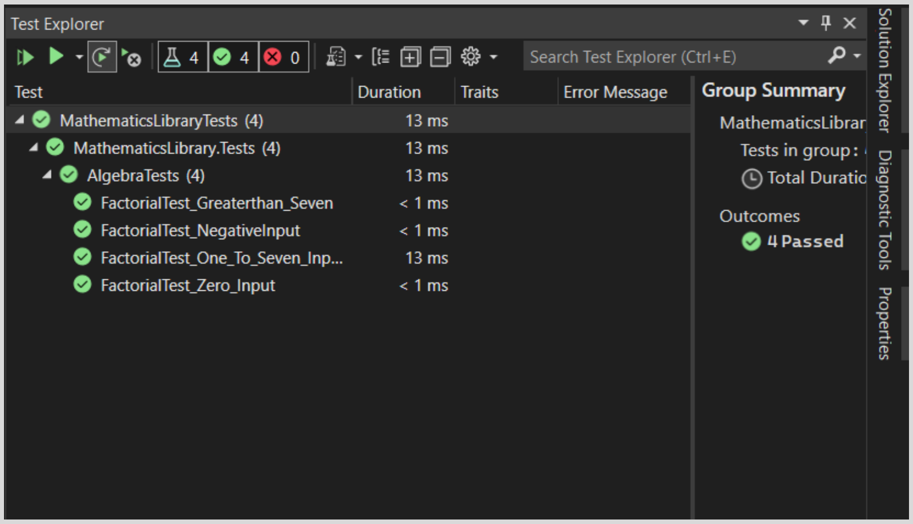
Test Explorer

Search Test Explorer (Ctrl+E)

```
Test                                      Duration  Traits  Error Message
MathematicsLibraryTests (4)               160 ms
  MathematicsLibrary.Tests (4)            160 ms
    AlgebraTests (4)                      160 ms
      FactorialTest_Greaterthan_Seven     146 ms            Assert.AreEqual fa
      FactorialTest_NegativeInput           1 ms            Assert.AreEqual fa
      FactorialTest_One_To_Seven_Inp...    13 ms
      FactorialTest_Zero_Input            < 1 ms            Assert.AreEqual fa
```

Group Summary
MathematicsLibrar
Tests in group:
Total Duratic
Outcomes
1 Passed
3 Failed

Test cases Pass



Test Explorer

Search Test Explorer (Ctrl+E)

```
Test                                      Duration  Traits  Error Message
MathematicsLibraryTests (4)                13 ms
  MathematicsLibrary.Tests (4)             13 ms
    AlgebraTests (4)                       13 ms
      FactorialTest_Greaterthan_Seven     < 1 ms
      FactorialTest_NegativeInput         < 1 ms
      FactorialTest_One_To_Seven_Inp...    13 ms
      FactorialTest_Zero_Input            < 1 ms
```

Group Summary
MathematicsLibrar
Tests in group:
Total Duratic
Outcomes
4 Passed

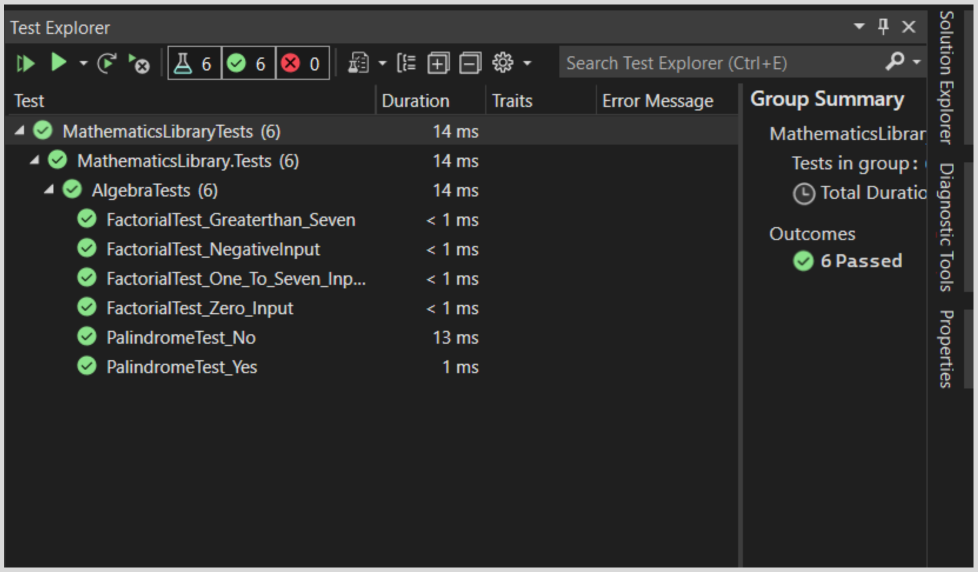
8) Add one more method to check if the number is palindrome or not in the above Algebra class and write test case for the same.

ALGEBRA CLASS (Palindrome method)

```
sing System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//create a layered architecture
//project with seperate class library for Business logic.

namespace MathematicsLibrary
{
    public  class Algebra
    {
        /// <summary>
        /// Business Requriment
        /// </summary>
        /// <param name="n">Factorial Number</param>
        /// <returns>Factorial</returns>
        public static int Factorial(int n)
        {
            int fact = 1;
            if (n == 0)
                return 1;

            else if (n > 7)
                return -999;
            else if (n < 0)
                return -9999;
            else
            {
                for (int i = 1; i <= n; i++)
                    fact = fact * i;
                return fact;
            }

        }
        public static int Palindrome(int n)
        {
            int m, rem, rev = 0;
            m = n;
            while (m > 0)
            {
                rem = m % 10;
                m = m / 10;
                rev = rev * 10 + rem;
            }
            if (rev == n)
                return rev;
            else
                return rev;

        }
    }
}
```

Palindrome Test Case

```
[TestMethod()]
        public void PalindromeTest_Yes()
        {
            //arrange
            int n = 232;
            int expected = 232;
            //act
            int actual = Algebra.Palindrome(n);
            //assert
            Assert.AreEqual(expected, actual);

        }
        [TestMethod()]
        public void PalindromeTest_No()
        {
            //arrange
            int n = 145;
            int expected = 541;
            //act
            int actual= Algebra.Palindrome(n);
            //Assert
            Assert.AreNotSame(expected, actual);

        }
```



```
Test                                   Duration   Traits   Error Message
◢ MathematicsLibraryTests (6)              14 ms
  ◢ MathematicsLibrary.Tests (6)           14 ms
    ◢ AlgebraTests (6)                     14 ms
        FactorialTest_Greaterthan_Seven   < 1 ms
        FactorialTest_NegativeInput       < 1 ms
        FactorialTest_One_To_Seven_Inp... < 1 ms
        FactorialTest_Zero_Input          < 1 ms
        PalindromeTest_No                  13 ms
        PalindromeTest_Yes                  1 ms

Group Summary
  MathematicsLibrar
    Tests in group:
    Total Duratic
  Outcomes
    6 Passed
```