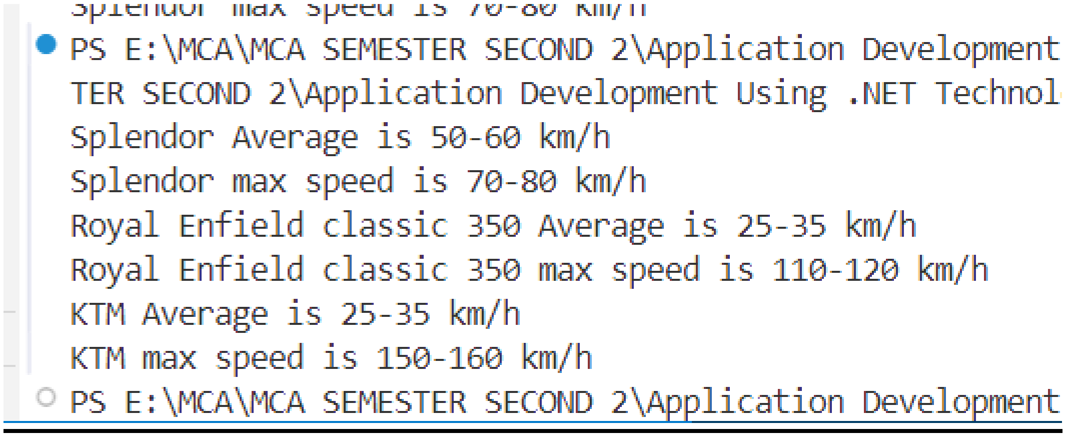
Program 31: Create an abstract class named as "Bike" that contains two abstract methods named as maxspeed () and average (). Create three classes for bike splendor, Royal Enfield class 350 and KTM and these classes are inheriting the "Bike" class. Write a program to display bike details with maximum speed and average of a bike.

Code:

```
using System;

namespace fourth_practical
{

    public abstract class Bike
    {
        public abstract void Average();
        public abstract void maxSpeed();
    }

    public class Splendor : Bike
    {
        public override void Average()
        {
            Console.WriteLine("Splendor Average is 50-60 km/h");
        }
        public override void maxSpeed()
        {
            Console.WriteLine("Splendor max speed is 70-80 km/h");
        }
    }
    public class RoyalEnfield350 : Bike
    {
        public override void Average()
        {
            Console.WriteLine("Royal Enfield classic 350 Average is 25-35 km/h");
        }
        public override void maxSpeed()
        {
            Console.WriteLine("Royal Enfield classic 350 max speed is 110-120 km/h");
        }
    }
    public class KTM : Bike
    {
        public override void Average()
        {
            Console.WriteLine("KTM Average is 25-35 km/h");
        }
        public override void maxSpeed()
        {
            Console.WriteLine("KTM max speed is 150-160 km/h");
        }
    }
    public class Program
    {
        public static void Main(string[] args)
        {
            // Splendor
            Splendor splendor = new Splendor();
            splendor.Average();
            splendor.maxSpeed();

            // Royal Enfield 350
            RoyalEnfield350 royal = new RoyalEnfield350();
            royal.Average();
            royal.maxSpeed();

            //KTM
            KTM ktm = new KTM();
            ktm.Average();
            ktm.maxSpeed();
        }

    }

}
```

Output:



```
Splendor max speed is 70-80 km/h
● PS E:\MCA\MCA SEMESTER SECOND 2\Application Development
TER SECOND 2\Application Development Using .NET Technol
Splendor Average is 50-60 km/h
Splendor max speed is 70-80 km/h
Royal Enfield classic 350 Average is 25-35 km/h
Royal Enfield classic 350 max speed is 110-120 km/h
KTM Average is 25-35 km/h
KTM max speed is 150-160 km/h
○ PS E:\MCA\MCA SEMESTER SECOND 2\Application Development
```

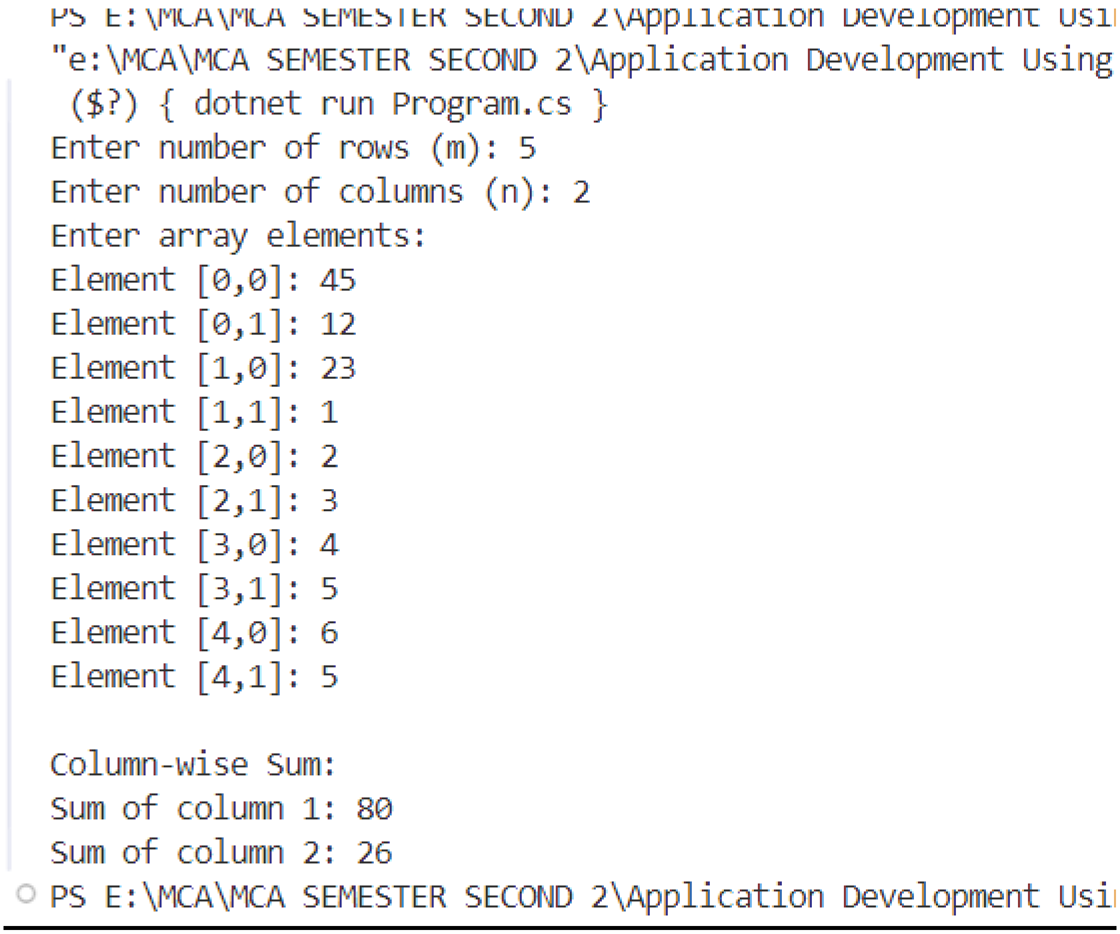
Program 32: Take mth row and nth column from user and declare a 2-D array with elements through user input. Write a program:

- **To display array in such a way that 1st column become 1st row, 2nd column become 2nd row and as follows if m is equivalent to n.**
- **To display the addition of each column of an array if m is greater than n.**
- **To display the addition of each row of an array if mis less than n.**

Code:

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter number of rows (m): ");
        int m = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter number of columns (n): ");
        int n = Convert.ToInt32(Console.ReadLine());

        int[,] array = new int[m, n];

        Console.WriteLine("Enter array elements:");
        for (int i = 0; i < m; i++)
        {
            for (int j = 0; j < n; j++)
            {
                Console.Write("Element [{0},{1}]: ", i, j);
                array[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }

        if (m == n)
        {
            Console.WriteLine("\nTransposed Matrix:");
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {
                    Console.Write(array[j, i] + " ");
                }
                Console.WriteLine();
            }
        }
        else if (m > n)
        {
            Console.WriteLine("\nColumn-wise Sum:");
            for (int j = 0; j < n; j++)
            {
                int sum = 0;
                for (int i = 0; i < m; i++)
                {
                    sum += array[i, j];
                }
                Console.WriteLine("Sum of column {0}: {1}", j + 1, sum);
            }
        }
        else
        {
            Console.WriteLine("\nRow-wise Sum:");
            for (int i = 0; i < m; i++)
            {
                int sum = 0;
                for (int j = 0; j < n; j++)
                {
                    sum += array[i, j];
                }
                Console.WriteLine("Sum of row {0}: {1}", i + 1, sum);
            }
        }
    }
}
```

Output:



```
PS E:\MCA\MCA SEMESTER SECOND 2\Application Development Usi
"e:\MCA\MCA SEMESTER SECOND 2\Application Development Using
 ($?) { dotnet run Program.cs }
Enter number of rows (m): 5
Enter number of columns (n): 2
Enter array elements:
Element [0,0]: 45
Element [0,1]: 12
Element [1,0]: 23
Element [1,1]: 1
Element [2,0]: 2
Element [2,1]: 3
Element [3,0]: 4
Element [3,1]: 5
Element [4,0]: 6
Element [4,1]: 5

Column-wise Sum:
Sum of column 1: 80
Sum of column 2: 26
PS E:\MCA\MCA SEMESTER SECOND 2\Application Development Usi
```

Program 33: Write a program to create a class name as "Convert" that contains data members as follows:

- **name: String**
- **show, (string): Display the string in such a way that if a character is a vowel, show the second previous character, and if the character is not a vowel, show the second next character.**

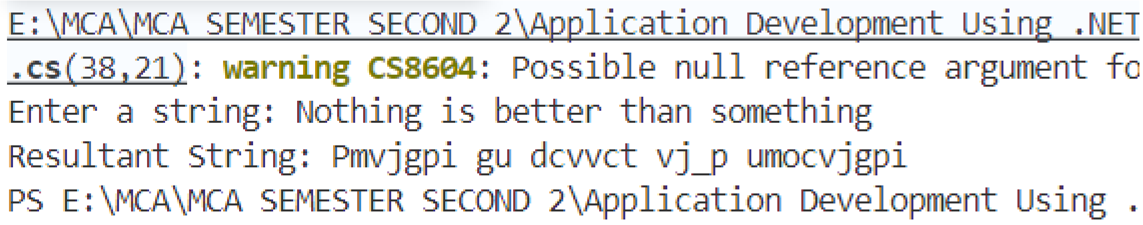
Note: Use of this keyword is compulsory, and you cannot create object in this program.

Example: name "Kavita danu" Resultant string- Myxgyy fyps

Code:

```
using System;

class Convert
{
    private string name;

    private Convert(string name)
    {
        this.name = name;
        this.show();
    }

    private void show()
    {
        string result = "";
        foreach (char ch in this.name)
        {
            if ("AEIOUaeiou".Contains(ch))
            {
                result += (char)(ch - 2);
            }
            else if (char.IsLetter(ch))
            {
                result += (char)(ch + 2);
            }
            else
            {
                result += ch;
            }
        }
        Console.WriteLine("Resultant String: " + result);
    }

    static void Main()
    {
        Console.Write("Enter a string: ");
        string input = Console.ReadLine();
        new Convert(input);
    }
}
```

Output:



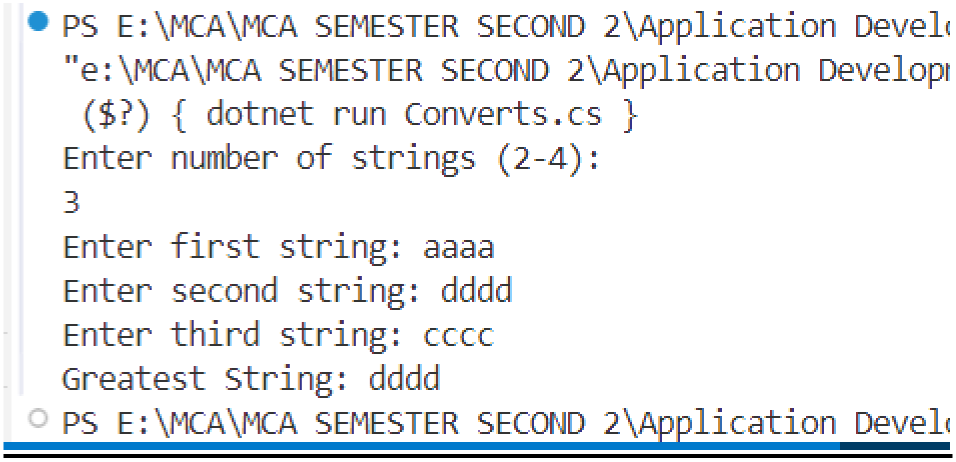
```
E:\MCA\MCA SEMESTER SECOND 2\Application Development Using .NET
.cs(38,21): warning CS8604: Possible null reference argument fo
Enter a string: Nothing is better than something
Resultant String: Pmvjgpi gu dcvvct vj_p umocvjgpi
PS E:\MCA\MCA SEMESTER SECOND 2\Application Development Using .
```

Program 34: Write a program to check which string is greatest out of two, three, and four string by using function overloading.

Code:

```
using System;

class StringCompare
{
    private static void Compare(string str1, string str2)
    {
        string greatest = str1.CompareTo(str2) > 0 ? str1 : str2;
        Console.WriteLine("Greatest String: " + greatest);
    }

    private static void Compare(string str1, string str2, string str3)
    {
        string greatest = str1;
        if (str2.CompareTo(greatest) > 0) greatest = str2;
        if (str3.CompareTo(greatest) > 0) greatest = str3;
        Console.WriteLine("Greatest String: " + greatest);
    }

    private static void Compare(string str1, string str2, string str3, string str4)
    {
        string greatest = str1;
        if (str2.CompareTo(greatest) > 0) greatest = str2;
        if (str3.CompareTo(greatest) > 0) greatest = str3;
        if (str4.CompareTo(greatest) > 0) greatest = str4;
        Console.WriteLine("Greatest String: " + greatest);
    }

    static void Main()
    {
        Console.WriteLine("Enter number of strings (2-4): ");
        int count = Convert.ToInt32(Console.ReadLine());

        if (count == 2)
        {
            Console.Write("Enter first string: ");
            string str1 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter second string: ");
            string str2 = Console.ReadLine() ?? string.Empty;
            Compare(str1, str2);
        }
        else if (count == 3)
        {
            Console.Write("Enter first string: ");
            string str1 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter second string: ");
            string str2 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter third string: ");
            string str3 = Console.ReadLine() ?? string.Empty;
            Compare(str1, str2, str3);
        }
        else if (count == 4)
        {
            Console.Write("Enter first string: ");
            string str1 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter second string: ");
            string str2 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter third string: ");
            string str3 = Console.ReadLine() ?? string.Empty;
            Console.Write("Enter fourth string: ");
            string str4 = Console.ReadLine() ?? string.Empty;
            Compare(str1, str2, str3, str4);
        }
        else
        {
            Console.WriteLine("Invalid input! Enter 2, 3, or 4 strings.");
        }
    }
}
```

<u>Output:</u>



```
PS E:\MCA\MCA SEMESTER SECOND 2\Application Devel
"e:\MCA\MCA SEMESTER SECOND 2\Application Develop
 ($?) { dotnet run Converts.cs }
Enter number of strings (2-4):
3
Enter first string: aaaa
Enter second string: dddd
Enter third string: cccc
Greatest String: dddd
PS E:\MCA\MCA SEMESTER SECOND 2\Application Devel
```

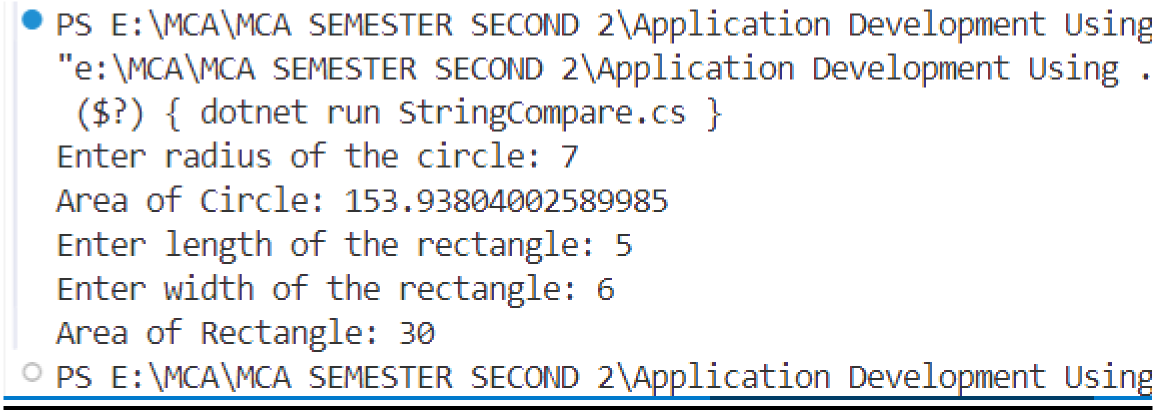
Program 35: Create a C# program where you have a base class 'Shape' with a virtual method called Calculate Area () that returns the area of a shape. Then, create derived classes 'Circle' and 'Rectangle' that inherit from 'Shape' and override the CalculateArea () method to calculate the area of a circle and rectangle respectively, based on their specific properties like radius and length/width.

Code:

```
// Shape.cs File
using System;

abstract class Shape
{
    public abstract double CalculateArea();
}

class Circle : Shape
{
    private double radius;

    public Circle(double radius)
    {
        this.radius = radius;
    }

    public override double CalculateArea()
    {
        return Math.PI * radius * radius;
    }
}

class Rectangle : Shape
{
    private double length, width;

    public Rectangle(double length, double width)
    {
        this.length = length;
        this.width = width;
    }

    public override double CalculateArea()
    {
        return length * width;
    }
}

// Program.cs file
class Program
{
    static void Main()
```

```
        {
            Console.Write("Enter radius of the circle: ");
            double radius = Convert.ToDouble(Console.ReadLine());
            Shape circle = new Circle(radius);
            Console.WriteLine("Area of Circle: " + circle.CalculateArea());

            Console.Write("Enter length of the rectangle: ");
            double length = Convert.ToDouble(Console.ReadLine());
            Console.Write("Enter width of the rectangle: ");
            double width = Convert.ToDouble(Console.ReadLine());
            Shape rectangle = new Rectangle(length, width);
            Console.WriteLine("Area of Rectangle: " + rectangle.CalculateArea());
        }
    }
```

Output:



```
● PS E:\MCA\MCA SEMESTER SECOND 2\Application Development Using
  "e:\MCA\MCA SEMESTER SECOND 2\Application Development Using .
   ($?) { dotnet run StringCompare.cs }
  Enter radius of the circle: 7
  Area of Circle: 153.93804002589985
  Enter length of the rectangle: 5
  Enter width of the rectangle: 6
  Area of Rectangle: 30
○ PS E:\MCA\MCA SEMESTER SECOND 2\Application Development Using
```