Lab-6

Name: Gondrala Mani Sai

Id Num: 2100031545

IN-LAB:

Task1:Develop a MyExtension **class, which declares the following extension methods:**

- the **SummaDigit** method, which extends the Int32 type and returns the sum of the digits of an arbitrary integer.

```
Example 1: n = 1274   result = 14 (14 = 1 + 2 + 7 + 4)
```

- the **SummaWithReverse** method, which extends the UInt32 type and returns the sum of the original positive integer with the number obtained from the original by rearranging all digits in reverse order

```
Example 2: n = 132    result = 363 (363 = 132 + 231)
```

- the **CountNotLetter** method, which extends the String type and returns the number of characters in the string that are not Latin letters.

```
Example 3: s = "I like C#"   result = 3 (there are two spaces and a "sharp"
character in the line)
```

- the **IsDayOff** method, which extends the DayOfWeek type and returns the boolean value true if it is a weekend (Saturday or Sunday) or the boolean value false if it is a weekday.

```
Example 4: day = DayOfWeek.Sunday  result = true
```

- the **EvenPositiveElements** method, which extends the IEnumerable <int> type and returns only even positive numbers from a set of integers

```
Example 5: int[] mas = { 2, -2, 3, 4, 0, 6, 1, 9 }    result =  2 , 4, 6
Example 6: for List<int> list = new List<int>{ 2, 3, -4, 8, 5, 4 }   result
= 2, 8, 4
```

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;

public static class MyExtension
{
    public static int SummaDigit(this int number)
    {
        int sum = 0;
        while (number != 0)
        {
            sum += number % 10;
            number /= 10;
```

```
        }
        return sum;
    }

    public static uint SummaWithReverse(this uint number)
    {
        uint reverse = 0;
        uint original = number;
        while (number > 0)
        {
            reverse = reverse * 10 + number % 10;
            number /= 10;
        }
        return original + reverse;
    }

    public static int CountNotLetter(this string str)
    {
        return str.Count(c => !char.IsLetter(c));
    }

    public static bool IsDayOff(this DayOfWeek day)
    {
        return day == DayOfWeek.Saturday || day == DayOfWeek.Sunday;
    }

    public static IEnumerable<int> EvenPositiveElements(this IEnumerable<int>
numbers)
    {
        return numbers.Where(n => n > 0 && n % 2 == 0);
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Extention_Methods
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n1 =Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"Sum of digits of {n1}: {n1.SummaDigit()}");

            uint n2 = Convert.ToUInt32(Console.ReadLine());
            Console.WriteLine($"Sum of {n2} with its reverse:
{n2.SummaWithReverse()}");

            string v = Console.ReadLine();
            string s = v;
            Console.WriteLine($"Number of non-letter characters in \"{s}\":
{s.CountNotLetter()}");

            DayOfWeek day = DayOfWeek.Sunday;
            Console.WriteLine($"Is {day} a day off? {day.IsDayOff()}");

            int[] mas = { 2, -2, 3, 4, 0, 6, 1, 9 };
```

```
            Console.WriteLine($"Even positive elements in array: {string.Join(",
", mas.EvenPositiveElements())}");

            var list = new List<int> { 2, 3, -4, 8, 5, 4 };
            Console.WriteLine($"Even positive elements in list: {string.Join(",
", list.EvenPositiveElements())}");
        }
    }
}
```

Output: