# .NET PROGRAMMING LAB 6

**Name: CH BALA GOWTHAM**

**ID:** 2000032067

**Section: S-**13

## 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
```

**Solution:**

```csharp
int summaDigit(int n)
    {
        int sum = 0;
while (n > 1)
        {
            sum += n % 10;
n /= 10;
        }
return sum;
    }
    int summaInverser(int n)
    {
        int rev = 0, num = n;
while (num) {            int d
= n% 10;          n = n %
10;           rev = rev * 10
+ d;
        }
        return rev+num;
    }
    int CountLetter(string s)
    {
int c=0;
        foreach(char ch in s)
        {
            if (!char.IsLetter(ch) || CharUnicodeInfo.GetUnicodeCategory(c) !=
UnicodeCategory.LowercaseLetter && CharUnicodeInfo.GetUnicodeCategory(c) !=
UnicodeCategory.UppercaseLetter) c++;
        }
return c;
    }
    bool isDayoff(DateTime d)
    {
        return d.DayOfWeek = DayOfWeek.Saturday || d.DayOfWeek=DayOfWeek.Sunday
    }
    public static class Enumerable
    {
        public static IEnumerable<int> EvenPositiveElements(this IEnumerable<int>
source) {
            foreach(int element in source)
            {
                if(element>0 && element%2==0) yeild return element;
            }
        }


    }
```
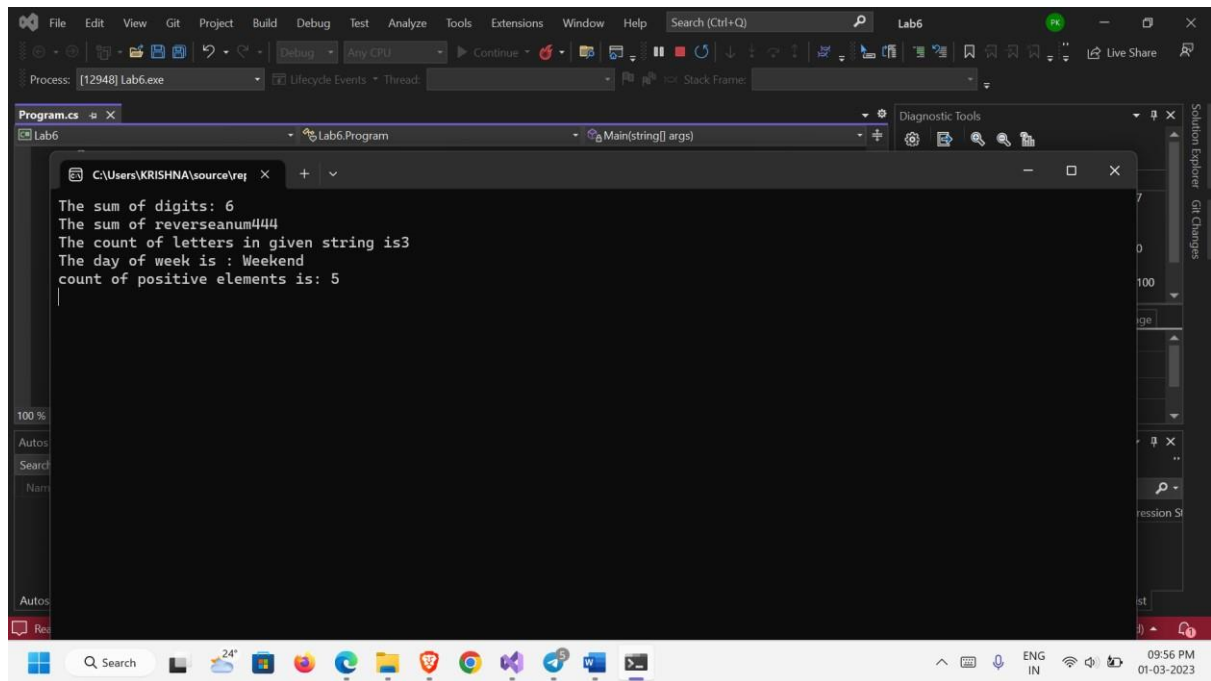
**Output:**