

LAB-6

Q1)Task1,2,3

2100032064

```
using System;
```

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

```
using System.Linq;
```

```
public static class MyExtension
```

```
{
```

```
    // Extension method to calculate the sum of the digits of an integer
```

```
    public static int SummaDigit(this int number)
```

```
    {
```

```
        int sum = 0;
```

```
        while (number != 0)
```

```
        {
```

```
            sum += number % 10;
```

```
            number /= 10;
```

```
        }
```

```
        return sum;
```

```
    }
```

```
    // Extension method to calculate the sum of a number with its reverse
```

```
    public static uint SummaWithReverse(this uint number)
```

```
    {
```

```
        uint reverseNumber = 0;
```

```
        uint n = number;
```

```
        while (n > 0)
```

```

{
    reverseNumber = reverseNumber * 10 + n % 10;
    n /= 10;
}

return number + reverseNumber;
}

```

// Extension method to count non-letter characters in a string

```
public static int CountNotLetter(this string str)
```

```

{
    int count = 0;
    foreach (char c in str)
    {
        if (!char.IsLetter(c))
        {
            count++;
        }
    }
    return count;
}

```

// Extension method to check if a day is a weekend

```
public static bool IsDayOff(this DayOfWeek day)
```

```

{
    return day == DayOfWeek.Saturday || day == DayOfWeek.Sunday;
}

```

// Extension method to filter even positive elements from a sequence of integers

```

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

```

```

class Program
{
    static void Main(string[] args)
    {
        // Example usage of extension methods

        int n1 = 1274;

        Console.WriteLine($"Sum of digits of {n1}: {n1.SummaDigit()}");

        uint n2 = 132;

        Console.WriteLine($"Sum of {n2} with its reverse: {n2.SummaWithReverse()}");

        string s = "I like C#";

        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[] arr = { 2, -2, 3, 4, 0, 6, 1, 9 };

        Console.WriteLine("Even positive elements in array: " + string.Join(", ", arr.EvenPositiveElements()));

        List<int> list = new List<int> { 2, 3, -4, 8, 5, 4 };
    }
}

```

```

        Console.WriteLine("Even positive elements in list: " + string.Join(", ",
list.EvenPositiveElements()));

    }

}

```

OUTPUT:

