LAB-9

Q) Task1,2,3,4,5 2100030408

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

class Program
{
    static void Main(string[] args)
    {
        // Task 1
        char targetChar1 = 'C';
        List<string> stringList1 = new List<string> { "Cat", "Dog", "Computer", "Car", "Cup" };
        List<string> result1 = GetStringsStartingAndEndingWithChar(stringList1, targetChar1);
        Console.WriteLine("Task 1:");
        Console.WriteLine(string.Join(", ", result1));

        // Task 2
        List<string> stringList2 = new List<string> { "Apple", "Banana", "Orange", "Grape" };
        List<int> result2 = GetSortedLengths(stringList2);
        Console.WriteLine("\nTask 2:");
        Console.WriteLine(string.Join(", ", result2));

        // Task 3
        List<string> stringList3 = new List<string> { "Hello", "World", "CSharp" };
        List<string> result3 = GetFirstAndLastChars(stringList3);
        Console.WriteLine("\nTask 3:");
        Console.WriteLine(string.Join(", ", result3));
```

```
    // Task 4
    int k = 3;
    List<string> stringList4 = new List<string> { "ABC1", "123", "CDE4", "WXYZ5", "456" };
    List<string> result4 = GetStringsOfLengthKEndingWithDigit(stringList4, k);
    Console.WriteLine("\nTask 4:");
    Console.WriteLine(string.Join(", ", result4));

    // Task 5
    List<int> integerList = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    List<string> result5 = GetOddIntegerStringRepresentations(integerList);
    Console.WriteLine("\nTask 5:");
    Console.WriteLine(string.Join(", ", result5));
}

static List<string> GetStringsStartingAndEndingWithChar(List<string> stringList, char targetChar)
{
    return stringList.Where(s => s.Length > 1 && s[0] == targetChar && s[^1] == targetChar).ToList();
}

static List<int> GetSortedLengths(List<string> stringList)
{
    return stringList.Select(s => s.Length).OrderBy(n => n).ToList();
}

static List<string> GetFirstAndLastChars(List<string> stringList)
{
    return stringList.Select(s => s[0].ToString() + s[^1]).ToList();
}

static List<string> GetStringsOfLengthKEndingWithDigit(List<string> stringList, int k)
```

```
    {
        return stringList.Where(s => s.Length == k && char.IsDigit(s[^1])).OrderBy(s => s).ToList();
    }

    static List<string> GetOddIntegerStringRepresentations(List<int> integerList)
    {
        return integerList.Where(n => n % 2 != 0).Select(n => n.ToString()).OrderBy(s => s).ToList();
    }
}
```

OUTPUT :