100. Count the occurrences of a character in a String

The **`CountOccurrences`** function counts the occurrences of a character in a String.

```
using System;
using System.Linq;

class Program100
{
    static void Main()
    {
        Console.WriteLine(CountOccurrences("hello world", 'l'));
        // Output: 3
    }

    static int CountOccurrences(string str, char ch)
    {
        return str.Count(c => c == ch);
    }
}
```

101. Convert Video Length to Seconds

The **`MinutesToSeconds`** function takes the length of a video in the format "mm:ss" and converts it to seconds. If the input format is correct, it returns the length of the video in seconds; otherwise, it returns 0.

```
using System;

class Program {
    static void Main() {
        Console.WriteLine(MinutesToSeconds("02:54"));
        // Output: 174
    }

    static int MinutesToSeconds(string time) {
        string[] arr = time.Split(':');
        if (int.Parse(arr[1]) < 60) {
            return int.Parse(arr[0]) * 60 + int.Parse(arr[1]);
        }
        return 0;
    }
}
```

102. Remove Duplicates from a String

The **RemoveDuplicatesFromString** function removes duplicates from a String.

```
using System;
using System.Linq;

class Program102
{
    static void Main()
    {
        Console.WriteLine(RemoveDuplicatesFromString("hello"));
        // helo
    }

    static string RemoveDuplicatesFromString(string str)
    {
        return new string(str.Distinct().ToArray());
    }
}
```

103. Find the Mode of an Array of Numbers

The **Mode** function finds the mode of an array of numbers.

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

class Program103
{
    static void Main()
    {
        var numbers = new int[] { 1, 2, 2, 3, 3, 3, 4, 4, 4, 4 };
        var mode = Mode(numbers);
        Console.WriteLine($"Mode: {string.Join(", ", mode)}");
        // Output: Mode: 4
    }

    static IEnumerable<int> Mode(IEnumerable<int> arr) => arr.GroupBy(x =>
x).ToDictionary(g => g.Key, g => g.Count()).Where(kv => kv.Value ==
arr.GroupBy(x => x).ToDictionary(g => g.Key, g =>
g.Count()).Values.Max()).Select(kv => kv.Key);
}
```

104. Check for Repdigit

The **`IsRepdigit`** function checks if a given integer is a repdigit, meaning it is composed of the same digit repeated. If the integer is a repdigit, it returns true; otherwise, it returns false.

```
using System;
using System.Linq;

class Program104 {
    static void Main() {
        Console.WriteLine(IsRepdigit(333));
        // True
    }

    static bool IsRepdigit(int num) => num.ToString().Distinct().Count() == 1;
}
```

105. Convert Binary Number to Decimal

The **`BinaryToDecimal`** function converts a binary number to decimal.

```
using System;
using System.Linq;

class Program105 {
  static void Main() {
    Console.WriteLine(BinaryToDecimal("1101"));
    // Output: 13
  }

  static int BinaryToDecimal(string binary) {
    return binary.Reverse()
      .Select((bit, index) => (bit - '0') * (int) Math.Pow(2, index))
      .Sum();
  }
}
```

106. Check if an Array is Sorted in Descending Order

The **`SortedDescending`** function checks if an array is sorted in descending order.

```
using System;
using System.Linq;

class Program106
{
    static void Main()
    {
        Console.WriteLine(SortedDescending(new int[] { 5, 4, 3, 2, 1 }));
        // Output: True
        Console.WriteLine(SortedDescending(new int[] { 1, 5, 3, 8, 2 }));
        // Output: False
    }

    static bool SortedDescending(int[] arr)
    {
        return arr.Zip(arr.Skip(1), (a, b) => a >= b).All(x => x);
    }
}
```

107. Find the Average of Even Numbers in an Array

The **`AverageOfEvenNumbers`** function finds the average of even numbers in an array.

```
using System;
using System.Linq;

class Program107
{
    static void Main()
    {
        Console.WriteLine(AverageOfEvenNumbers(new int[] { 1, 2, 3, 4, 5, 6, 7,
8, 9, 10 }));
        // Output: 6.0
    }

    static double AverageOfEvenNumbers(int[] arr)
    {
        var evenNumbers = arr.Where(x => x % 2 == 0);
        return evenNumbers.Average();
    }
}
```

108. Capitalize the First Letter of Each Word in a String

The **`CapitalizeWords`** capitalizes the first letter of each word in a String.

```
using System;
using System.Globalization;

class Program108
{
    static void Main()
    {
        Console.WriteLine(CapitalizeWords("hello world"));
        // Output: "Hello World"
    }

    static string CapitalizeWords(string str)
    {
        return CultureInfo.CurrentCulture.TextInfo.ToTitleCase(str);
    }
}
```

109. Check if an Array is a Subset of Another Array

The **`IsSubset`** function checks if an array is a subset of another array.

```
using System;
using System.Linq;

class Program109
{
    static void Main()
    {
        Console.WriteLine(IsSubset(new int[] { 1, 2, 3 }, new int[] { 2, 3, 4,
5, 6 }));
        // Output: False
        Console.WriteLine(IsSubset(new int[] { 1, 2, 3 }, new int[] { 2, 3, 1,
5, 6 }));
        // Output: True
    }

    static bool IsSubset(int[] arr1, int[] arr2)
    {
        return arr1.All(item => arr2.Contains(item));
    }
}
```

110. Find the Minimum and Maximum Numbers in an Array

The **MinMax** function finds the minimum and maximum numbers in an array.

```
using System;
using System.Linq;

class Program110
{
    static void Main()
    {
        var result = MinMax(new int[] { 10, 5, 25, 3, 15 });
        Console.WriteLine($"{{ min: {result.min}, max: {result.max} }}");
        // Output: { min: 3, max: 25 }
    }

    static (int min, int max) MinMax(int[] arr)
    {
        return (arr.Min(), arr.Max());
    }
}
```

111. Validate Zip Code

The **`IsValid`** function validates whether a given string is a valid zip code.

```
using System;

class Program111 {
    static void Main() {
        Console.WriteLine(IsValid("12345"));
        // Output: True
        Console.WriteLine(IsValid("1234"));
        // Output: False
        Console.WriteLine(IsValid("123456"));
        // Output: False
        Console.WriteLine(IsValid("12 45"));
        // Output: False
    }

    static bool IsValid(string zip) {
        return System.Text.RegularExpressions.Regex.IsMatch(zip, @"^\d{5}$");
    }
}
```

112. Remove Null Values from a List.

The **`RemoveNull`** function removes null values from a list.

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

class Program112
{
    static void Main()
    {
        List<object> array = new List<object> { 1, null, 2, 3, null, 4, null };
        var result = RemoveNull(array);
        Console.WriteLine($"Output: [{string.Join(", ", result)}]");
        // Output: [1, 2, 3, 4]
    }

    static List<object> RemoveNull(List<object> arr)
    {
        return arr.Where(item => item != null).ToList();
    }
}
```

113. Maurice's Racing Snails

Maurice and Steve engage in a snail race, each owning three snails of different speeds: slow (s), medium (m), and fast (f). Although Maurice's snails are generally faster, Steve's strategy poses a challenge. In each of the three rounds, they pit their snails against each other strategically. Maurice's plan involves sacrificing his slower snails strategically to ensure victory. This function evaluates Maurice's plan, determining if he wins at least 2 out of 3 games against Steve's snails.

```
using System;

class Program {
    static void Main() {
        Console.WriteLine(MauriceWins(new int[] {1, 2, 3}, new int[] {3, 2,
1}));
        // Output: true
    }

    static bool MauriceWins(int[] mSnails, int[] sSnails) => (mSnails[0] >
sSnails[2] ? 1 : 0) + (mSnails[1] > sSnails[0] ? 1 : 0) + (mSnails[2] >
sSnails[1] ? 1 : 0) >= 2;
}
```

114. Calculate the Sum of Cubes of an Array

The **`SumOfCubes`** function calculates the sum of cubes of an array.

```
using System;
using System.Linq;

class Program114
{
    static void Main()
    {
        int[] arr = { 1, 2, 3, 4, 5 };
        int sum = SumOfCubes(arr);
        Console.WriteLine($"Output: {sum}");
        // Output: 225
    }

    static int SumOfCubes(int[] arr)
    {
        return arr.Sum(x => x * x * x);
    }
}
```

115. Shuffle the Characters of a String

The **ShuffleString** function suffle the characters of a String.

```
using System;
using System.Linq;

class Program115
{
    static void Main()
    {
        string str = "hello";
        string shuffledString = ShuffleString(str);
        Console.WriteLine($"Output: {shuffledString}");
        // Output: oelhl
    }

    static string ShuffleString(string str)
    {
        Random rand = new Random();
        var shuffledChars = str.OrderBy(c => rand.Next()).ToArray();
        return new string(shuffledChars);
    }
}
```