Exam Preparation

Test your tasks in the Judge system: https://judge.softuni.org/Contests/4455/Exam-Preparation-II

1. Magic Numbers

Write a program that:

- Reads an integer number **N** from the console, always greater than or equal to 1
- Print all **magic** numbers in range **[1; N]** - separated by single space
- The number is magic when:
  - **All of its digits are prime numbers**
  - **Sum of all digits is divisible by 2**
- If the are no such numbers print "**no**"

Example

Input	Output	Comments
25	2 22	We have to check numbers in range [1; 25] Magic numbers in this range are 2 and 22.
100	2 22 33 35 37 53 55 57 73 75 77	

2. Min / Max Values

Write a program that:

- Reads an **array of integer numbers** from the first line of the console, separated by single space
- Read an **integer number N** from the second line of the console
- Find **max number in the first N elements** in the given array
- Find **min number in the first N elements** in the given array
- Print **max number** and **min number**, each on **separate line**

Example

Input	Output	Comments
3 42 61 7 8 9 10 23 4	61 3	First 4 numbers in the array are: 3 42 61 7 Max number is: 61 Min number is: 3
12 34 98 42 65 12 3	98 12	First 3 numbers in the array are: 12 34 98 Max number is: 98 Min number is: 12

3. Unit Test Method: Pascal Triangle

Test a given method which takes in **an integer n** and prints that number of **rows** from the **pascal triangle**.

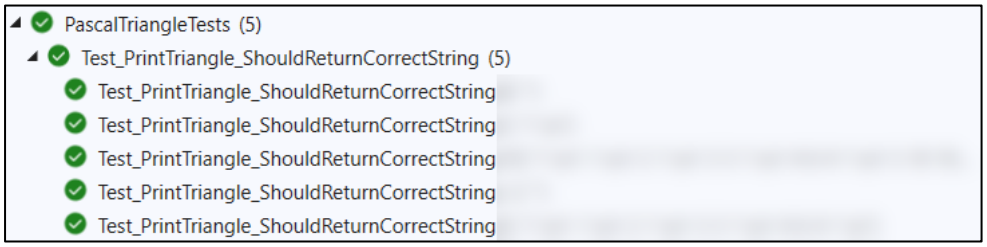
The method is found in the **`PascalTriangle.cs`** file:

```
public class PascalTriangle
{
    1 reference
    public static string PrintTriangle(int n)
    {
        string result = string.Empty;
        for (int line = 0; line < n; line++)
        {
            int number = 1;
            for (int j = 0; j <= line; j++)
            {
                result += $"{number} ";
                number = number * (line - j) / (j + 1);
            }

            result += "\n";
        }

        return result;
    }
}
```

You are given a **test file `PascalTriangleTests.cs`** containing **5 empty test cases**. Implement all the cases:

```
public class PascalTriangleTests
{
    //[TestCase()]
    //[TestCase()]
    //[TestCase()]
    //[TestCase()]
    //[TestCase()]
    0 references
    public void Test_PrintTriangle_ShouldReturnCorrectString(int n, string expected)
    {
        // TODO: implement the test and finish the test cases
    }
}
```

When you are ready make sure your **tests run:**



IMPORTANT: DO NOT REMOVE OR CHANGE ANY NAMESPACES AND USINGS.

4. Unit Test Array: Longest Increasing Subsequence

Test a given method which takes in **an integer array** and finds the **longest increasing subsequence** (LIS).

The method is found in the **`LongestIncreasingSubsequence.cs`** file:

```
public class LongestIncreasingSubsequence
{
    1 reference
    public static string GetLis(int[]? arr)
    {
        if (arr is null)
        {
            throw new ArgumentNullException(nameof(arr));
        }

        int[] length = new int[arr.Length];
        int[] previous = new int[arr.Length];
        int maxLength = 0;
        int maxIndex = 0;
```

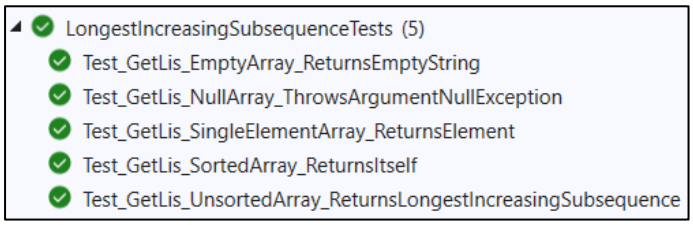
```
        for (int i = 0; i < arr.Length; i++)
        {
            length[i] = 1;
            previous[i] = -1;

            for (int j = 0; j < i; j++)
            {
                if (arr[j] >= arr[i] || length[j] + 1 <= length[i])
                {
                    continue;
                }

                length[i] = length[j] + 1;
                previous[i] = j;
            }

            if (length[i] > maxLength)
            {
                maxLength = length[i];
                maxIndex = i;
            }
        }
```

```
        int[] sequence = new int[maxLength];
        int current = maxIndex;

        for (int i = maxLength - 1; i >= 0; i--)
        {
            sequence[i] = arr[current];
            current = previous[current];
        }

        return string.Join(" ", sequence);
    }
}
```

You are given a **test file `LongestIncreasingSubsequenceTests.cs`** containing **5 empty tests**. Implement all the unit tests:

```
public class LongestIncreasingSubsequenceTests
{
    [Test]
    0 references
    public void Test_GetLis_NullArray_ThrowsArgumentNullException()...

    [Test]
    0 references
    public void Test_GetLis_EmptyArray_ReturnsEmptyString()...

    [Test]
    0 references
    public void Test_GetLis_SingleElementArray_ReturnsElement()...

    [Test]
    0 references
    public void Test_GetLis_UnsortedArray_ReturnsLongestIncreasingSubsequence()...

    [Test]
    0 references
    public void Test_GetLis_SortedArray_ReturnsItself()...
}
```

When you are ready make sure your **tests run:**



IMPORTANT: DO NOT REMOVE OR CHANGE ANY NAMESPACES AND USINGS.