Solution@08-12-23

2496. Maximum Value of a String in an Array

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class MaxStringLengthInArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input: Number of strings in the array
        System.out.print("Enter the number of strings: ");
        int n = scanner.nextInt();

        // Input: Array of strings
        String[] strings = new String[n];
        System.out.println("Enter the strings:");
        for (int i = 0; i < n; i++) {
            strings[i] = scanner.next();
        }

        // Find all strings with the maximum length
        List<String> maxValues = findAllMaxLengthValues(strings);

        // Output: Maximum values
        System.out.println("Maximum values in the array based on string length:");
        for (String maxValue : maxValues) {
            System.out.println(maxValue);
        }

        scanner.close();
```

```
    }

    private static List<String> findAllMaxLengthValues(String[] strings) {
        // Initialize with the first string in the array
        String maxLengthValue = strings[0];

        // List to store all strings with the maximum length
        List<String> maxValues = new ArrayList<>();
        maxValues.add(maxLengthValue);

        // Compare each string in the array based on length
        for (int i = 1; i < strings.length; i++) {
            int currentLength = strings[i].length();
            int maxLength = maxLengthValue.length();

            if (currentLength > maxLength) {
                // If the current string is longer than the current maximum, update maxValues
                maxValues.clear();
                maxValues.add(strings[i]);
                maxLengthValue = strings[i];
            } else if (currentLength == maxLength) {
                // If the current string has the same length as the current maximum, add it to maxValues
                maxValues.add(strings[i]);
            }
        }

        return maxValues;
    }
}
```

Output:-

```
PS C:\Users\Ajeet\Desktop\java> javac MaxStringLengthInArray.java
PS C:\Users\Ajeet\Desktop\java> java MaxStringLengthInArray
Enter the number of strings: 5
Enter the strings:
orange apple kiwi banana Guawa
Maximum values in the array based on string length:
orange
banana
```

2470. Number of Subarrays With LCM Equal to K

```
import java.util.HashMap;
import java.util.Map;

public class SubarraysWithLCM {

  public static int countSubarraysWithLCM(int[] arr, int k) {
    int n = arr.length;
    int left = 0, right = 0, count = 0, lcm = 1;

    Map<Integer, Integer> freqMap = new HashMap<>();

    while (right < n) {
      lcm = calculateLCM(lcm, arr[right]);

      while (lcm > k && left <= right) {
        lcm /= arr[left++];
      }

      if (lcm == k) {
        count += right - left + 1;
```

```
      }

      right++;
    }

    return count;
  }

  private static int calculateLCM(int a, int b) {
    return a * b / gcd(a, b);
  }

  private static int gcd(int a, int b) {
    while (b != 0) {
      int temp = b;
      b = a % b;
      a = temp;
    }
    return a;
  }

  public static void main(String[] args) {
    int[] arr = { 2, 5, 6 };
    int k = 10;

    int result = countSubarraysWithLCM(arr, k);
    System.out.println("Number of subarrays with LCM equal to " + k + ": " + result);
  }
}
```

Output:-

```
PS C:\Users\Ajeet\Desktop\java> javac SubarraysWithLCM.java
PS C:\Users\Ajeet\Desktop\java> java SubarraysWithLCM
Number of subarrays with LCM equal to 10: 2
```