1. Replace each array element by its corresponding rank

Given an array of distinct integers, replace each array element by its corresponding rank in the array.

The minimum array element has the rank 1; the second minimum element has a rank of 2, and so on…

Ans: package Testing;

```
import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;


public class testing {


	public static void main(String[] args) {

		// TODO Auto-generated method stub

		System.out.println("this is a java program");

		int [] arr =  { 10, 8, 15, 12, 6, 20, 1 } ;

		int []arr1=new int[arr.length];

		int []arr2=new int[arr.length];

		for(int i=0;i<arr.length;i++) {

			arr1[i]=arr[i];
```

```
            }
        Arrays.sort(arr);
        System.out.println();
        for(int i=0;i<arr.length;i++) {
            for(int j=0;j<arr1.length;j++) {
                if(arr1[i]==arr[j]) {
arr2[i]=j+1;

                    break;
                }
            }
        }
        for (int i : arr2) {
            System.out.print(i+" ");
        }
```

2. a string s, find the length of the longest substring without repeating characters.

Example 1:

Input: s = "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: s = "bbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: s = "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

Constraints:

0 <= s.length <= 5 * 104

s consists of English letters, digits, symbols and spaces.

Ans: package Testing;

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class testing {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("this is a java program");
	System.out.println();
System.out.println("this is 2nd question ");
//String str = "abcabcbb";
//String str = "pwwkew";
String str="bbbbb";

    int maxLength = 0;

    for (int i = 0; i < str.length(); i++) {
       Set<Character> set = new HashSet<>();
```

```
    for (int j = i; j < str.length(); j++) {
        char c = str.charAt(j);
        if (set.contains(c)) {

            break;
        } else {
            set.add(c);
            maxLength = Math.max(maxLength, j - i + 1);
        }
    }
}

System.out.println("2nd question ans "+maxLength);
```

3. Find non-repeating characters in a string

Ans: package Testing;

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class testing {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.out.println("this is a java program");
    String st = "vrushali deshmukh";
    System.out.println("Finding a non-repetitive character");

    boolean foundNonR = false;

    for (int i = 0; i < st.length(); i++) {
       char ch = st.charAt(i);
       boolean isRepetitive = false;

       for (int j = 0; j < st.length(); j++) {
```

```
        if (i != j && ch == st.charAt(j)) {
            isRepetitive = true;
            break;
        }
    }

    if (!isRepetitive) {
        System.out.println(ch);
        foundNonR = true;
        break;}
}
```

4. You are given an array of integers.

Write a C# program to find the frequency of each unique element in the array and

store the results in a dictionary where the key is the element and the value is its frequency.

Then, print the elements and their frequencies.

Ans:

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


class Program
{
    static void Main(string[] args)
    {
        int[] arr = { 1, 2, 3, 4, 5, 1, 2, 3, 4, 1, 2, 1 };


        Dictionary<int, int> frequencyMap = new Dictionary<int, int>();


        // Count frequencies
        foreach (int num in arr)
        {
            if (frequencyMap.ContainsKey(num))
            {
                frequencyMap[num]++;
            }
```

```
            else
            {
                frequencyMap[num] = 1;
            }
        }

        // Print frequencies
        foreach (var pair in frequencyMap)
        {
            Console.WriteLine("Element: " + pair.Key + ", Frequency: " +
pair.Value);
        }
    }
}
```