1.Two sums:

using System;

using System.Collections.Generic;

public class Solution {

public int[] TwoSum(int[] nums, int target) {

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

for (int i = 0; i < nums.Length; i++) {

int element = target - nums[i];

if (index.ContainsKey(element)) {

return new int[] { index[element], i };

}

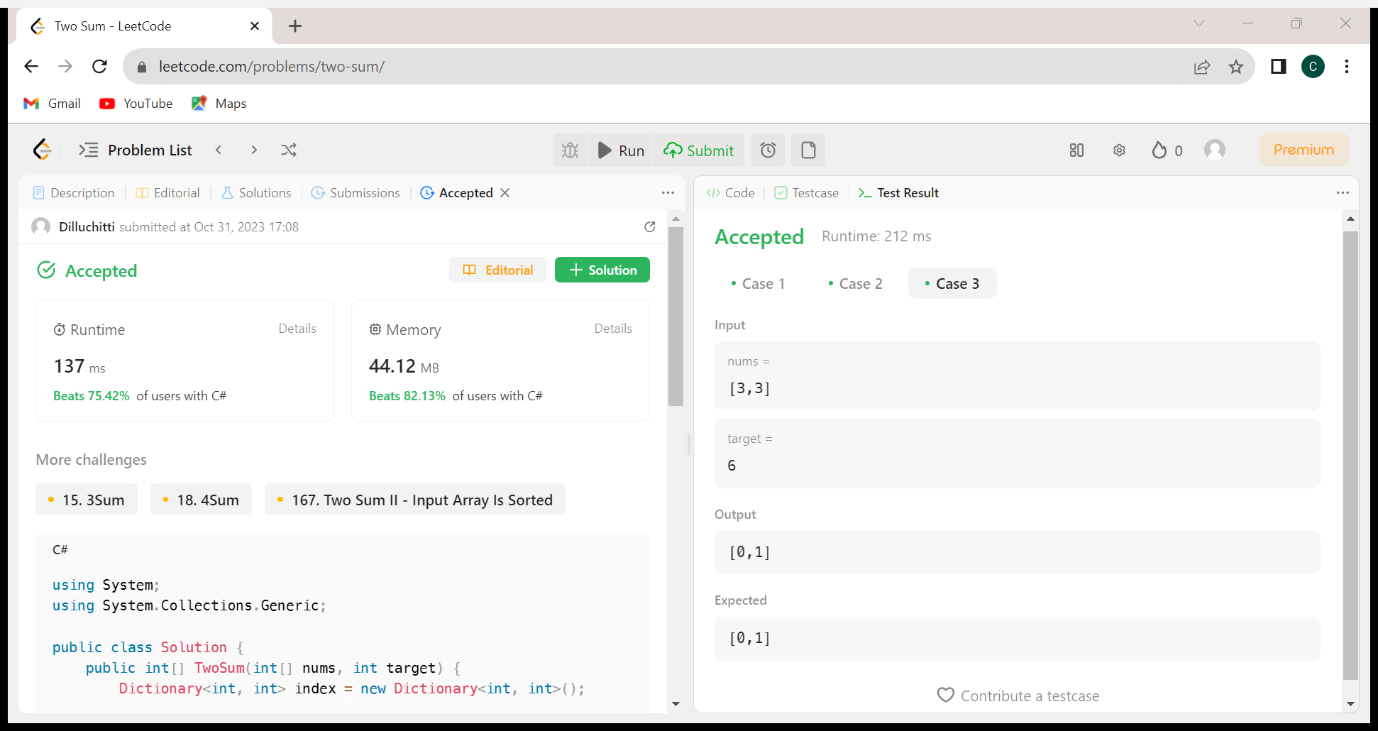
index[nums[i]] = i;

}

throw new ArgumentException("solution not found.");

}

}



2. Palindrome:

public class Solution{

public bool IsPalindrome(int x)

{

if (x < 0)

{

return false;

}

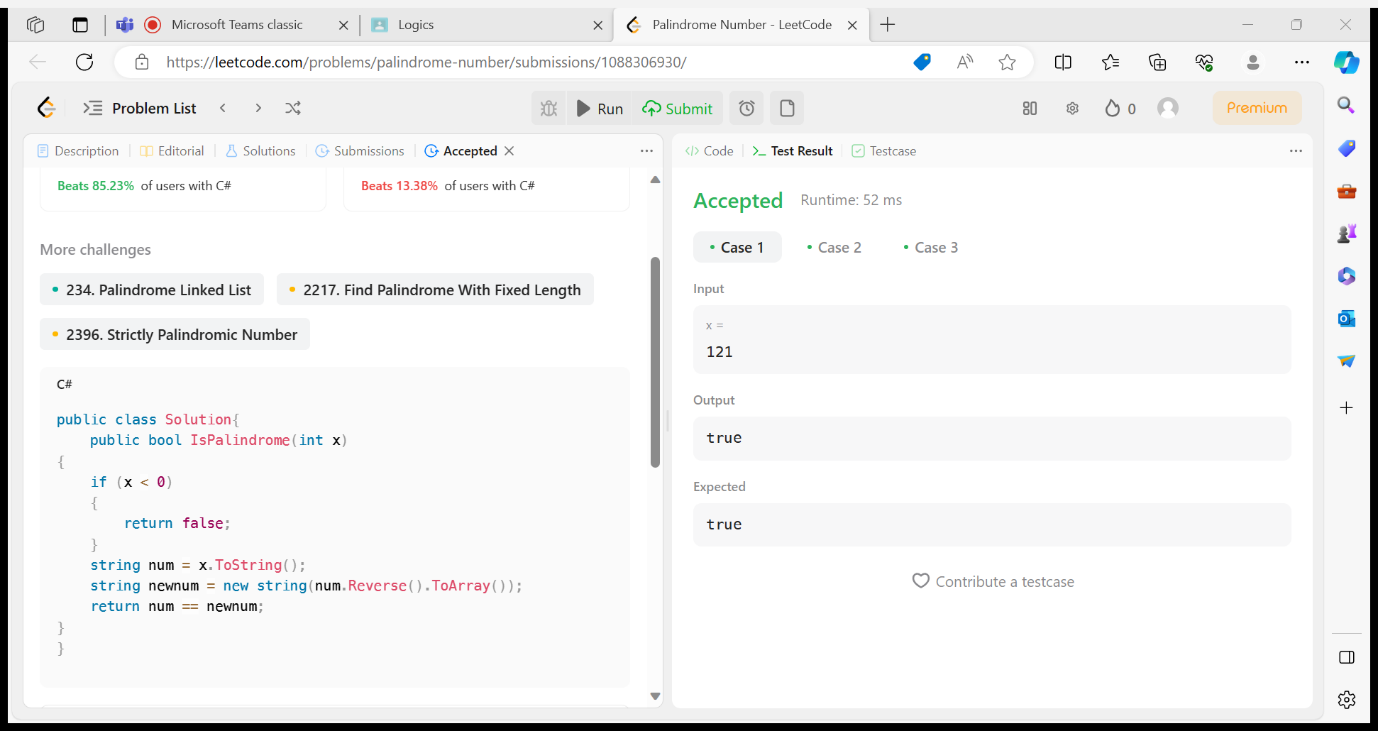
string num = x.ToString();

string newnum = new string(num.Reverse().ToArray());

return num == newnum;

}

}



3.Remove Duplicates from Sorted Array:

public class Solution {

public int RemoveDuplicates(int[] nums)

{

if (nums.Length == 0)

return 0;

int k = 1;

for (int i = 1; i < nums.Length; i++) {

if (nums[i] != nums[i - 1])

{

nums[k] = nums[i];

k++;

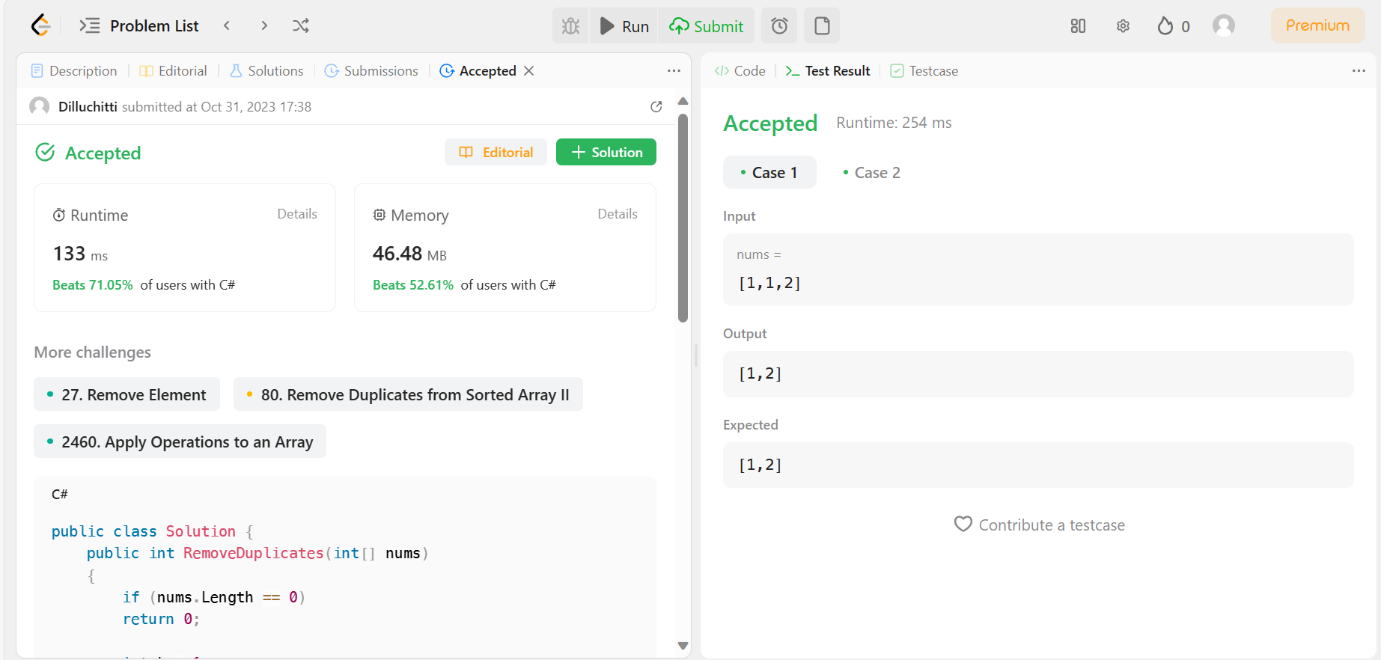
}

}

return k;

}

}



4.Longest Common Prefix:

public class Solution

{

public string LongestCommonPrefix(string[] strs)

{

if (strs == null || strs.Length == 0)

{

return "";

}

string prefix = strs[0];

for (int i = 1; i < strs.Length; i++)

{

while (strs[i].IndexOf(prefix) != 0)

{

prefix = prefix.Substring(0, prefix.Length - 1);

if (string.IsNullOrEmpty(prefix))

{

return "";

}

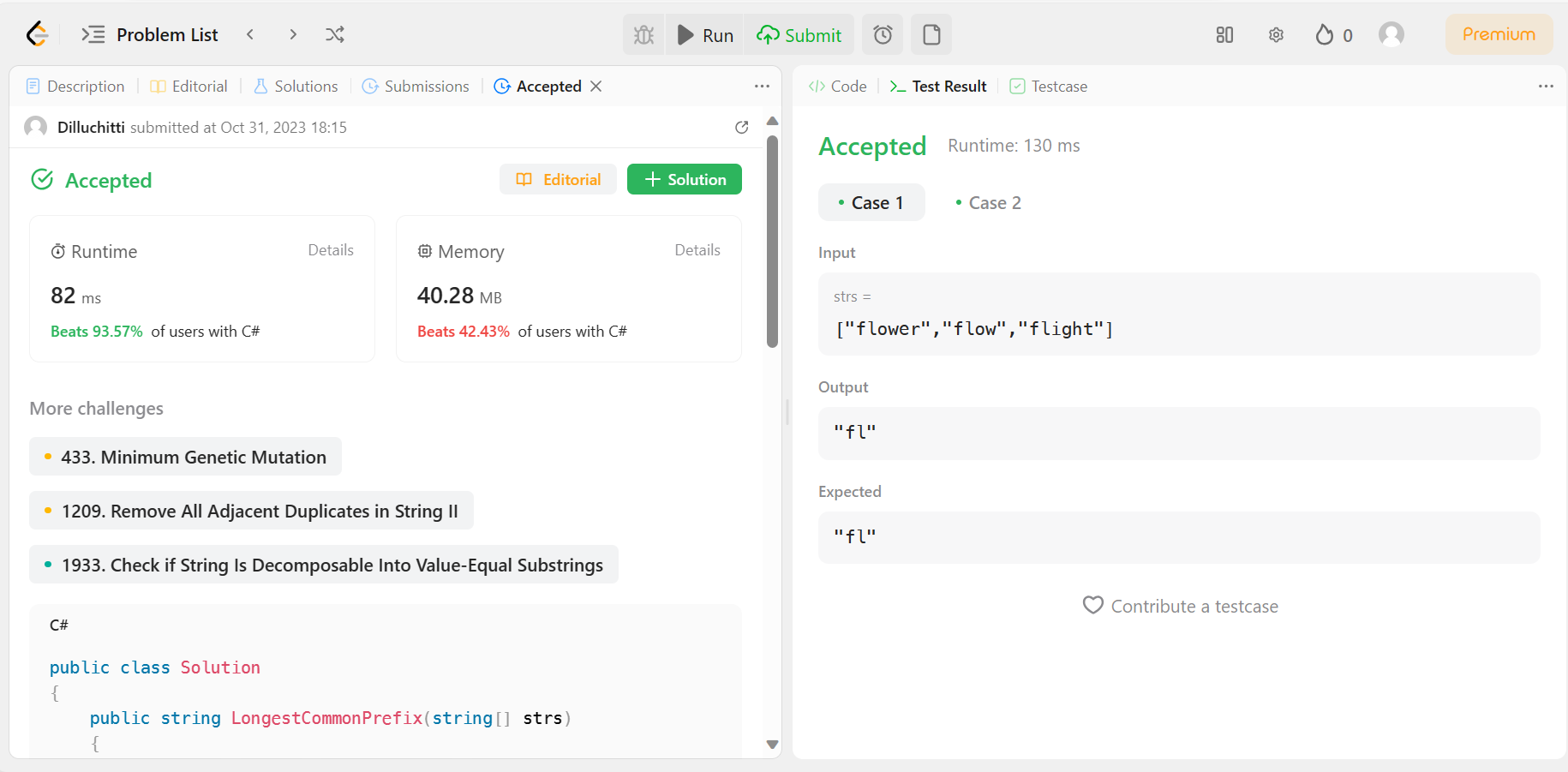
}

}

return prefix;

}

}



5.Regular Expression Matching:

public class Solution

{

public bool IsMatch(string s, string p)

{

int x = s.Length;

int y = p.Length;

bool[,] array = new bool[x+ 1, y + 1];

array[0, 0] = true;

for (int i = 0; i <= x; i++) {

for (int j = 1; j <= y; j++) {

if (p[j - 1] != '\*') {

array[i, j] = i > 0 && (s[i - 1] == p[j - 1] || p[j - 1] == '.') && array[i - 1, j - 1];

} else {

array[i, j] = array[i, j - 2] || (i > 0 && (s[i - 1] == p[j - 2] || p[j - 2] == '.') && array[i - 1, j]);

}

}

}

return array[x, y];

}

}

