Date: 18th Nov 2024

1. Bubble Sort (TC: O(N^2))

class Solution {

// Function to sort the array using bubble sort algorithm.

public static void bubbleSort(int arr[]) {

// code here

for(int i=0;i<arr.length-1;i++)

{

boolean swapped=false;

for(int j=0;j<arr.length-1-i;j++)

{

if(arr[j+1]<arr[j])

{

int temp=arr[j+1];

arr[j+1]=arr[j];

arr[j]=temp;

swapped=true;

}

}

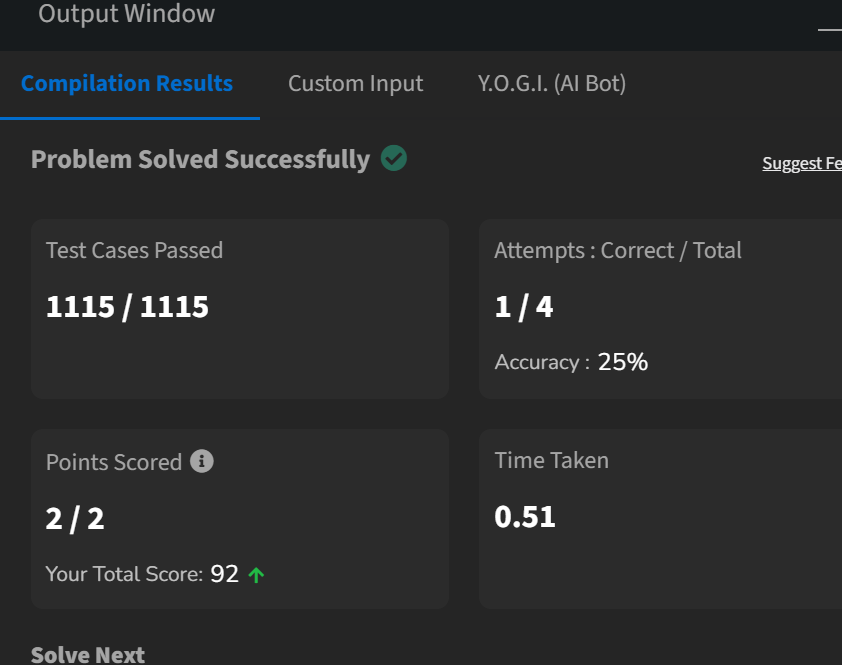
if(!swapped) break;

}

}

}

Output:



1. Quick Sort (TC: O(n log n)):

class Solution {

static void swap(int[] arr,int i,int j){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

static void quickSort(int arr[], int low, int high) {

if(low<high)

{

int pi=partition(arr,low,high);

quickSort(arr,low,pi-1);

quickSort(arr,pi+1,high);

}

}

static int partition(int arr[], int low, int high) {

int piv=arr[high];

int i=low-1;

for(int j=low;j<=high-1;j++)

{

if(arr[j]<piv)

{

i++;

swap(arr,i,j);

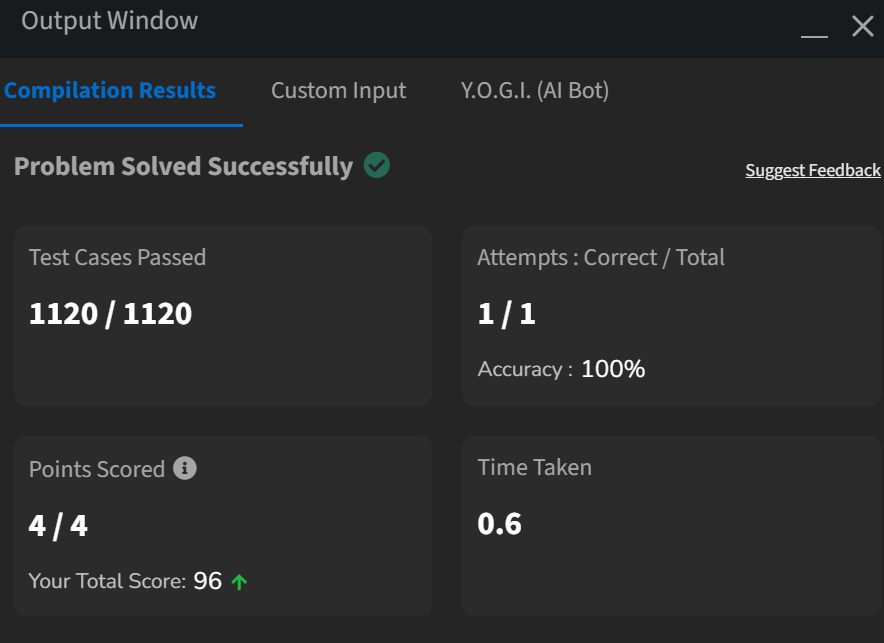
} }

swap(arr,i+1,high);

return i+1;}

}

Output:



3.Non Repeating Character(TC: O(N)):

class Solution {

static char nonRepeatingChar(String s) {

HashMap<Character,Integer> a=new HashMap<>();

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

{

a.put(s.charAt(i),a.getOrDefault(s.charAt(i),0)+1);

}

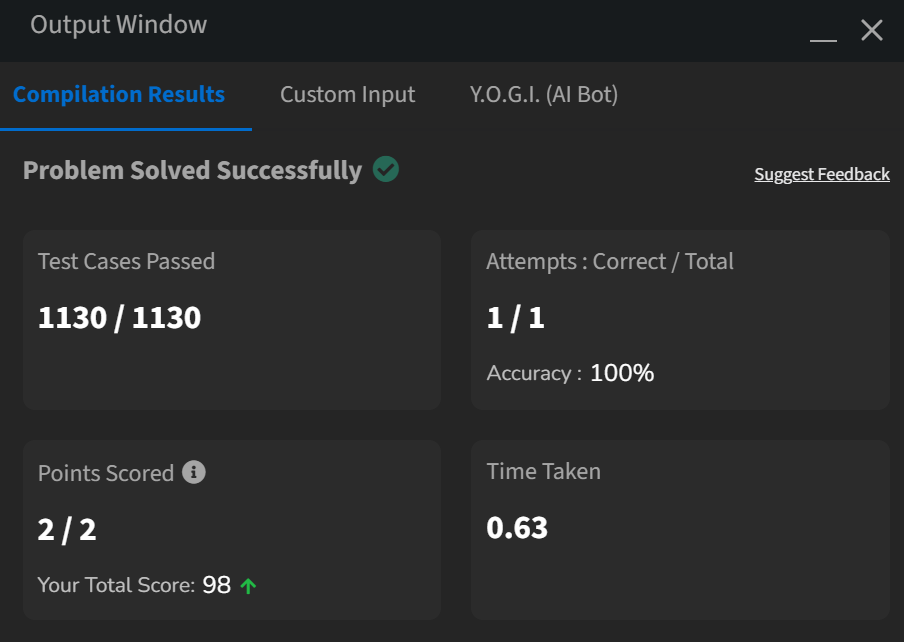
for(int i=0;i<s.length();i++){

if(a.get(s.charAt(i))==1)

{return s.charAt(i);}}

return '$';}}

Output:



4.K-largest elements(TC: O(n))

class Solution {

static List<Integer> kLargest(int arr[], int k) {

List<Integer> a=new ArrayList<>();

Arrays.sort(arr);

for(int i=arr.length-1;i>arr.length-k-1;i--)

{

a.add(arr[i]);

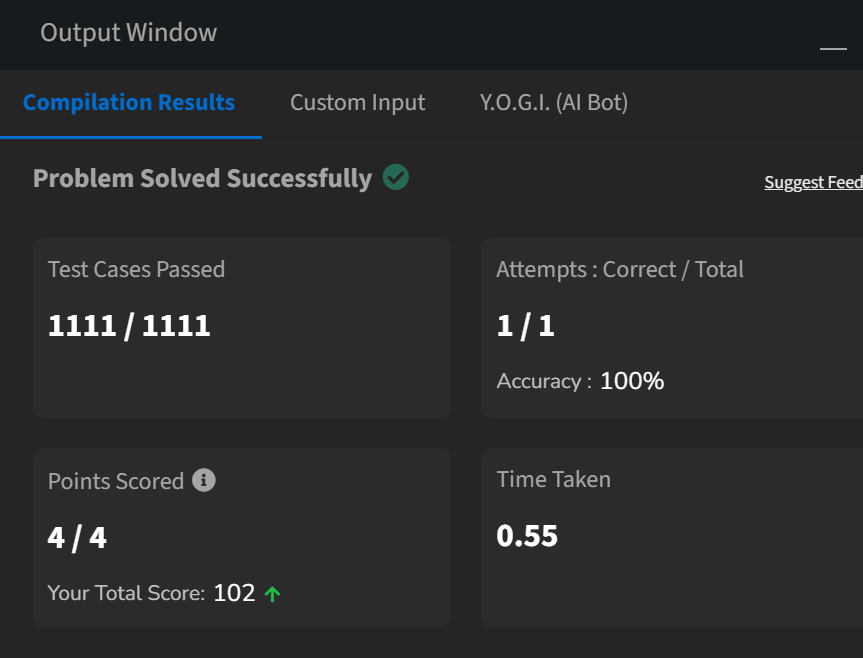
}

return a;

}

}

Output:



5. Form The Largest Number(O(n))

class Solution {

static class MyCompare implements Comparator<String> {

public int compare(String x, String y) {

String xy = x + y;

String yx = y + x;

return xy.compareTo(yx) > 0 ? -1 : 1; r

}

}

String printLargest(int[] arr) {

String[] strArr =

Arrays.stream(arr).mapToObj(String::valueOf).toArray(String[] ::new)

Arrays.sort(strArr, new MyCompare());

StringBuilder ans = new StringBuilder();

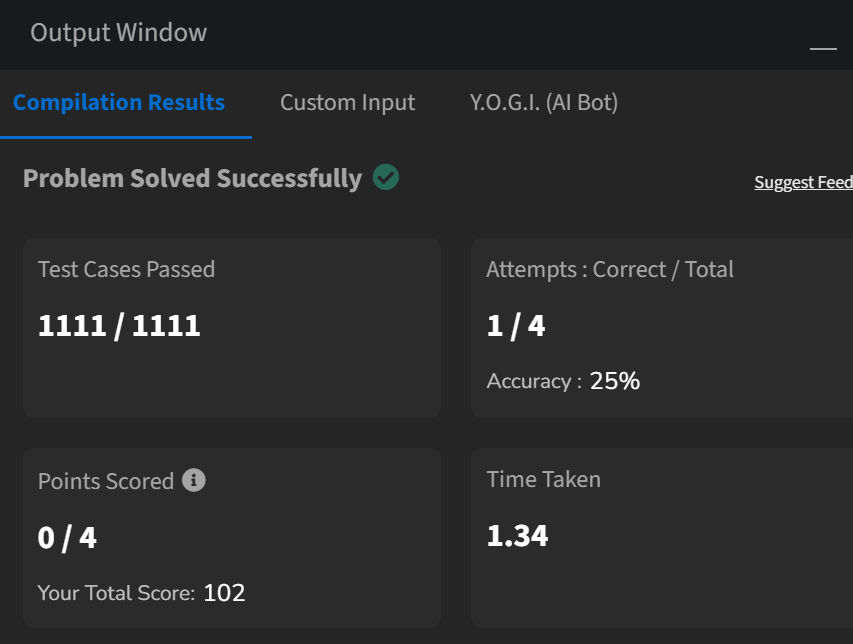
for (String num : strArr) ans.append(num);

return ans.toString();

}

}

Output:



6. Edit Distance(TC:O(N^2)):

class Solution {

public int editDistance(String s1, String s2) {

int m = s1.length();

int n = s2.length();

int[] prev = new int[n + 1];

int[] curr = new int[n + 1];

for (int j = 0; j <= n; j++)

prev[j] = j

for (int i = 1; i <= m; i++) {

curr[0] = i;

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

if (s1.charAt(i - 1) == s2.charAt(j - 1))

curr[j] = prev[j - 1];

else

curr[j] = 1 + Math.min(curr[j - 1], Math.min(prev[j], prev[j - 1]));

}

int[] temp = prev;

prev = curr;

curr = temp;

}

return prev[n];

}

}  
  
Output:

