Classwork

Swap alternate elements in an array in **C++**

```
#include<iostream>
using namespace std;

void alternateSwap(int arr[],int arraySize){
    for(int i=0;i<arraySize;i=i+2){
        if((i+1)<(arraySize)){
            swap(arr[i],arr[i+1]);
        }
    }

    cout<<"Array, After alternate swaping "<<endl;
    for(int i=0;i<arraySize;i++){
        cout<<arr[i]<<" ";
    }
}

int main(){
    int arr[1000];
    int arraySize;
    cout<<"Enter size of array "<<endl;
    cin>>arraySize;

    cout<<"Enter elements in an array"<<endl;
    for(int i=0;i<arraySize;i++){
        cin>>arr[i];
    }

    cout<<"Array, Before alternate swaping "<<endl;
    for(int i=0;i<arraySize;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;

    alternateSwap(arr,arraySize);
}
```

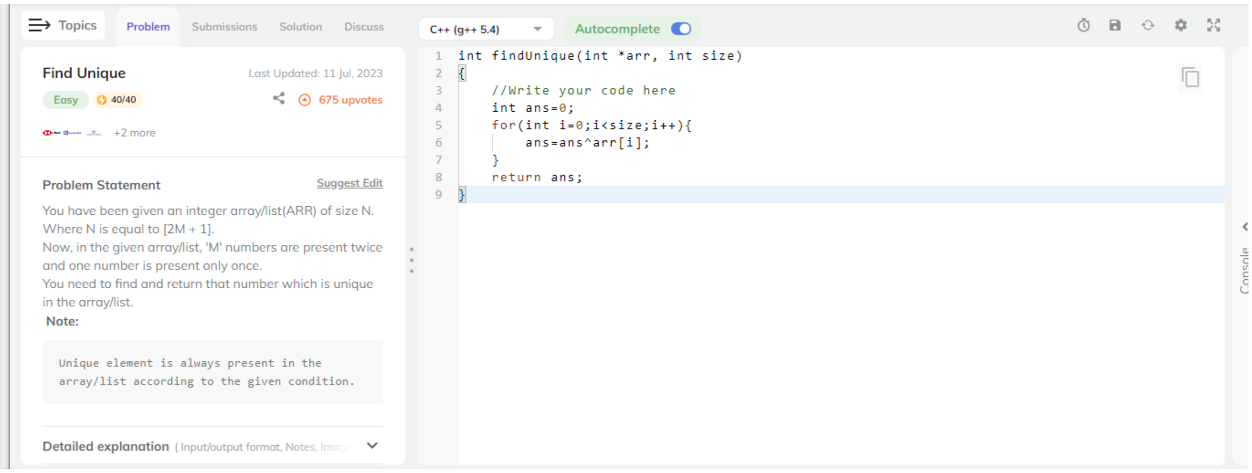
Swap alternate elements in an array in **Java**

```
package com.help.code;
import java.util.*;
public class SwapAlternate {

     public static void swapArrayAlternatively(int arr[],int
arraySize) {
          for(int i=0;i<arraySize;i=i+2) {
               if((i+1)<arraySize) {
                    int temp=arr[i];
                    arr[i]=arr[i+1];
                    arr[i+1]=temp;
               }
          }

          System.out.println("Array, After alternate swaping ");
          for(int i=0;i<arraySize;i++) {
               System.out.print(arr[i]+" ");
          }
     }

     public static void main(String[] args) {
          // TODO Auto-generated method stub
          Scanner sc=new Scanner(System.in);
          int[] arr=new int[1000];
          System.out.println("Enter size of array ");
          int arraySize=sc.nextInt();

          System.out.println("Enter elements in an array");
          for(int i=0;i<arraySize;i++) {
               arr[i]=sc.nextInt();
          }

          System.out.println("Array, Before alternate swaping
");
          for(int i=0;i<arraySize;i++) {
               System.out.print(arr[i]+" ");
          }
          System.out.println();
          swapArrayAlternatively(arr,arraySize);
     }
}
```

Find Unique [Coding Ninjas]

Question Link:-

https://www.codingninjas.com/studio/problems/find-unique_625159?source=youtube&campaign=love_babbar_codestudio1&utm_source=youtube&utm_medium=affiliate&utm_campaign=love_babbar_codestudio1&leftPanelTab=0



Find Unique

Last Updated: 11 Jul, 2023

Problem Statement

You have been given an integer array/list(ARR) of size N. Where N is equal to [2M + 1].
Now, in the given array/list, 'M' numbers are present twice and one number is present only once.
You need to find and return that number which is unique in the array/list.

Note:

```
Unique element is always present in the
array/list according to the given condition.
```

Detailed explanation (Input/output format, Notes, Image

```
int findUnique(int *arr, int size)
{
    //Write your code here
    int ans=0;
    for(int i=0;i<size;i++){
        ans=ans^arr[i];
    }
    return ans;
}
```

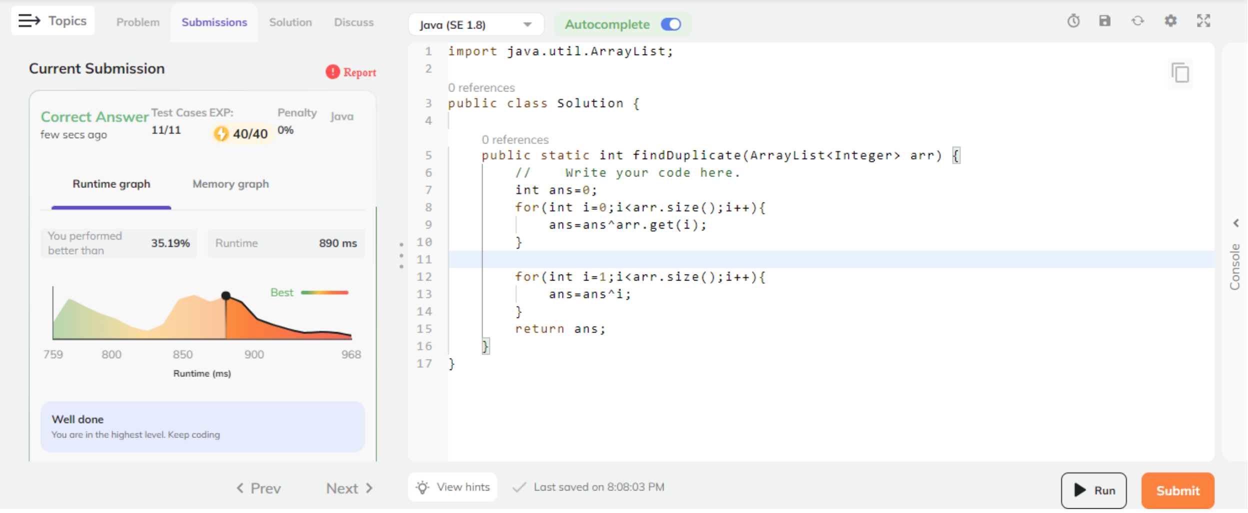
C++ code

```
int findUnique(int *arr, int size)
{
    //Write your code here
    int ans=0;
    for(int i=0;i<size;i++){
        ans=ans^arr[i];
    }
    return ans;
}
```

Duplicate In Array [Coding Ninjas]

Question Link:-

https://www.codingninjas.com/studio/problems/duplicate-in-array_893397?source=youtube&campaign=love_babbar_codestudio1&utm_source=youtube&utm_medium=affiliate&utm_campaign=love_babbar_codestudio1&leftPanelTab=0



Current Submission

Correct Answer
few secs ago

Test Cases: 11/11
EXP: 40/40
Penalty: 0%
Java

You performed better than 35.19%

Runtime 890 ms

Well done
You are in the highest level. Keep coding

```
import java.util.ArrayList;

0 references
public class Solution {

    0 references
    public static int findDuplicate(ArrayList<Integer> arr) {
        //    Write your code here.
        int ans=0;
        for(int i=0;i<arr.size();i++){
            ans=ans^arr.get(i);
        }

        for(int i=1;i<arr.size();i++){
            ans=ans^i;
        }
        return ans;
    }
}
```

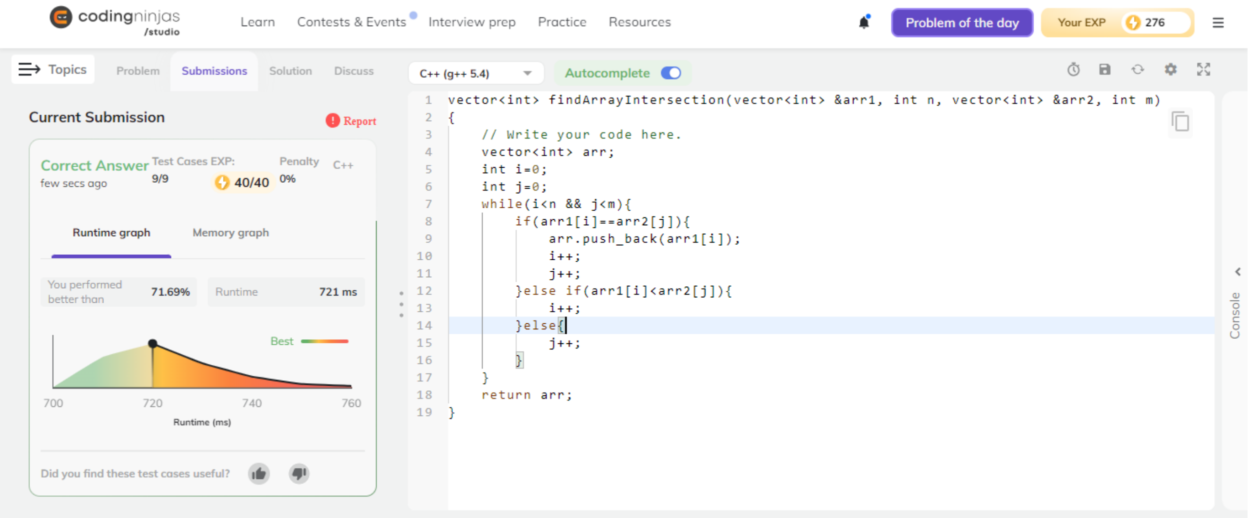
Last saved on 8:08:03 PM

Java Code

```
import java.util.ArrayList;

public class Solution {

    public static int findDuplicate(ArrayList<Integer> arr) {
        //    Write your code here.
        int ans=0;
        for(int i=0;i<arr.size();i++){
            ans=ans^arr.get(i);
        }

        for(int i=1;i<arr.size();i++){
            ans=ans^i;
        }
        return ans;
    }
}
```

Intersection Of Two Sorted Arrays [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/intersection-of-2-arrays_1082149?source=youtube&campaign=love_babbar_codestudio1&utm_source=youtube&utm_medium=affiliate&utm_campaign=love_babbar_codestudio1&leftPanelTab=0



Current Submission

Report

Correct Answer few secs ago

Test Cases 9/9

EXP: 40/40

Penalty 0%

C++

You performed better than 71.69%

Runtime 721 ms

Did you find these test cases useful?

```
vector<int> findArrayIntersection(vector<int> &arr1, int n, vector<int> &arr2, int m)
{
    // Write your code here.
    vector<int> arr;
    int i=0;
    int j=0;
    while(i<n && j<m){
        if(arr1[i]==arr2[j]){
            arr.push_back(arr1[i]);
            i++;
            j++;
        }else if(arr1[i]<arr2[j]){
            i++;
        }else{
            j++;
        }
    }
    return arr;
}
```

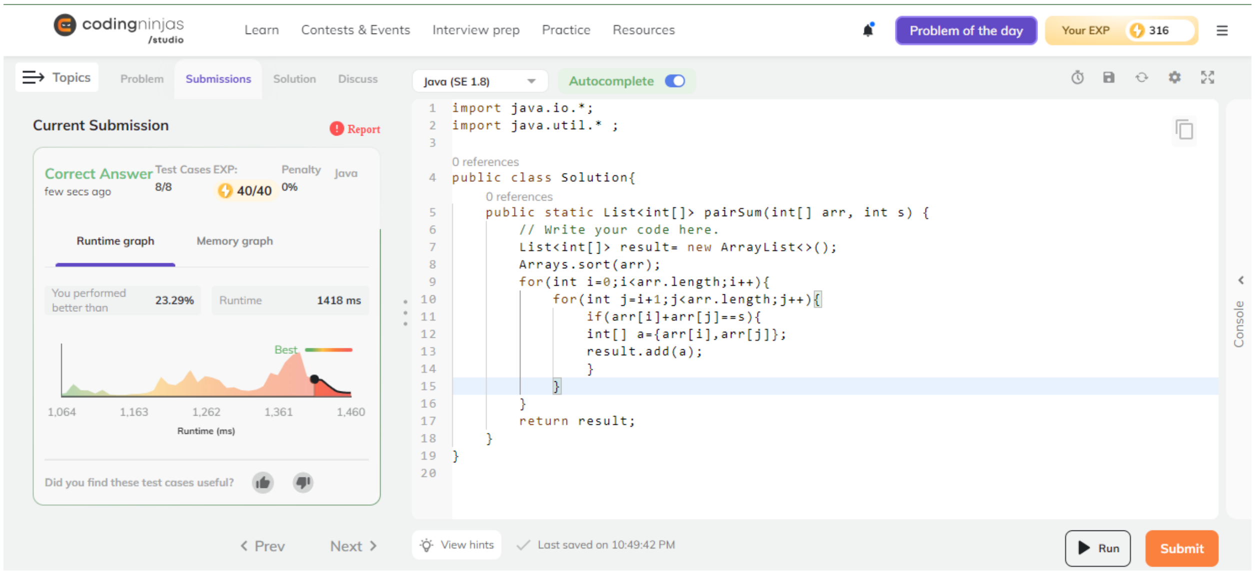
C++ Code

```
vector<int> findArrayIntersection(vector<int> &arr1, int n, vector<int>
&arr2, int m)
{
    // Write your code here.
    vector<int> arr;
    int i=0;
    int j=0;
    while(i<n && j<m){
        if(arr1[i]==arr2[j]){
            arr.push_back(arr1[i]);
            i++;
            j++;
        }else if(arr1[i]<arr2[j]){
            i++;
        }else{
            j++;
        }
    }
    return arr;
}
```

Pair Sum [Coding Ninjas]

Question Link:-

https://www.codingninjas.com/studio/problems/pair-sum_697295?leftPanelTab=0



Current Submission

Report

Correct Answer
few secs ago

Test Cases 8/8 EXP: 40/40 Penalty 0% Java

You performed better than 23.29%

Runtime 1418 ms

Did you find these test cases useful?

```
import java.io.*;
import java.util.* ;

0 references
public class Solution{
    0 references
    public static List<int[]> pairSum(int[] arr, int s) {
        // Write your code here.
        List<int[]> result= new ArrayList<>();
        Arrays.sort(arr);
        for(int i=0;i<arr.length;i++){
            for(int j=i+1;j<arr.length;j++){
                if(arr[i]+arr[j]==s){
                int[] a={arr[i],arr[j]};
                result.add(a);
                }
            }
        }
        return result;
    }
}
```

Last saved on 10:49:42 PM

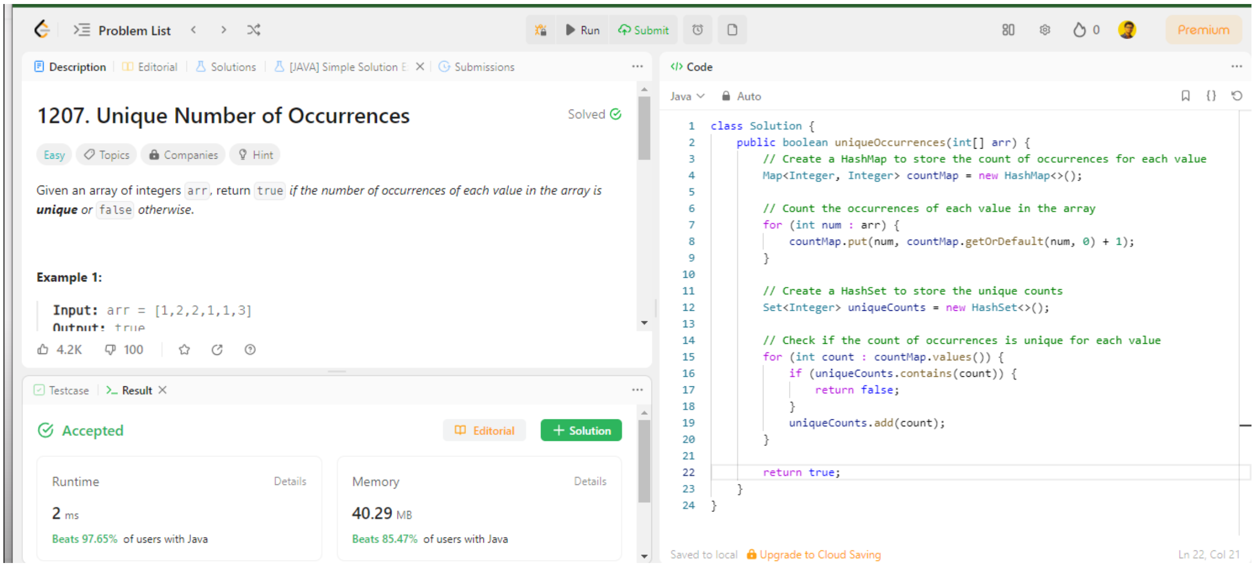
Java Code

```
import java.io.*;
import java.util.* ;

public class Solution{
    public static List<int[]> pairSum(int[] arr, int s) {
        // Write your code here.
        List<int[]> result= new ArrayList<>();
        Arrays.sort(arr);
        for(int i=0;i<arr.length;i++){
            for(int j=i+1;j<arr.length;j++){
                if(arr[i]+arr[j]==s){
                int[] a={arr[i],arr[j]};
                result.add(a);
                }
            }
        }
        return result;
    }
}
```

Homework

Unique Number of Occurrences [LeetCode]

Question Link:-
https://leetcode.com/problems/unique-number-of-occurrences/description/



1207. Unique Number of Occurrences

Given an array of integers `arr`, return `true` *if the number of occurrences of each value in the array is* ***unique*** *or* `false` *otherwise.*

Example 1:

```
Input: arr = [1,2,2,1,1,3]
Output: true
```

Accepted

Runtime 2 ms Beats 97.65% of users with Java

Memory 40.29 MB Beats 85.47% of users with Java

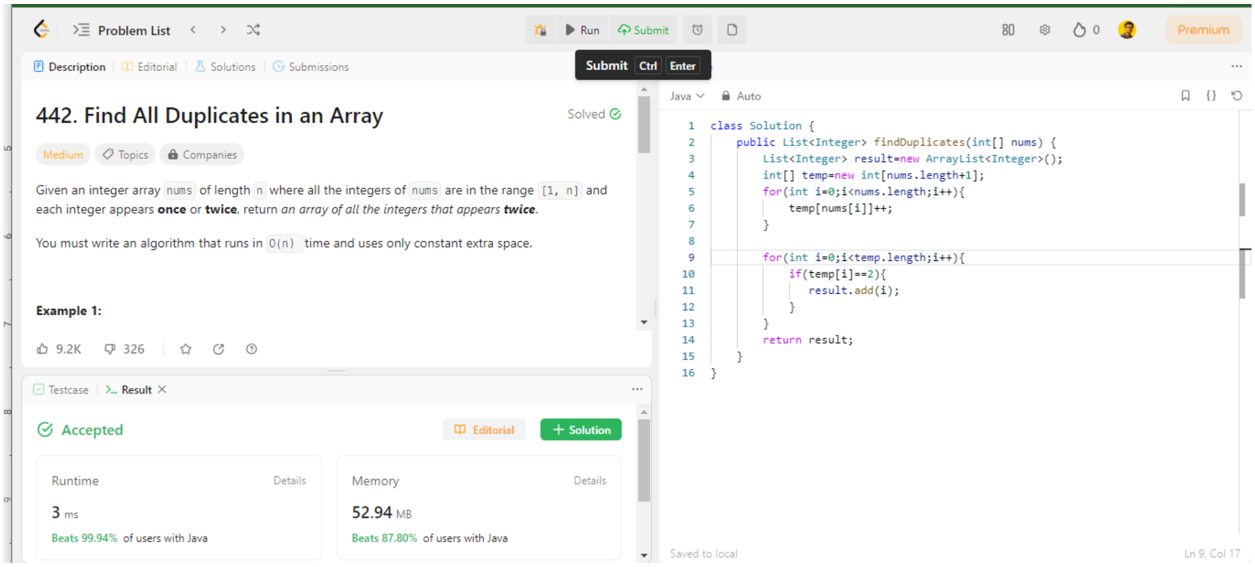
```java
class Solution {
    public boolean uniqueOccurrences(int[] arr) {
        // Create a HashMap to store the count of occurrences for each value
        Map<Integer, Integer> countMap = new HashMap<>();

        // Count the occurrences of each value in the array
        for (int num : arr) {
            countMap.put(num, countMap.getOrDefault(num, 0) + 1);
        }

        // Create a HashSet to store the unique counts
        Set<Integer> uniqueCounts = new HashSet<>();

        // Check if the count of occurrences is unique for each value
        for (int count : countMap.values()) {
            if (uniqueCounts.contains(count)) {
                return false;
            }
            uniqueCounts.add(count);
        }

        return true;
    }
}
```

Java Code

```java
class Solution {
    public boolean uniqueOccurrences(int[] arr) {
        // Create a HashMap to store the count of occurrences for each value
        Map<Integer, Integer> countMap = new HashMap<>();

        // Count the occurrences of each value in the array
        for (int num : arr) {
            countMap.put(num, countMap.getOrDefault(num, 0) + 1);
        }

        // Create a HashSet to store the unique counts
        Set<Integer> uniqueCounts = new HashSet<>();

        // Check if the count of occurrences is unique for each value
        for (int count : countMap.values()) {
            if (uniqueCounts.contains(count)) {
                return false;
            }
            uniqueCounts.add(count);
        }

        return true;
    }
}
```

Find all duplicates in an Array [LeetCode]

Question Link:-
https://leetcode.com/problems/find-all-duplicates-in-an-array/description/



442. Find All Duplicates in an Array

Given an integer array `nums` of length `n` where all the integers of `nums` are in the range `[1, n]` and each integer appears **once** or **twice**, return *an array of all the integers that appears* ***twice***.

You must write an algorithm that runs in `O(n)` time and uses only constant extra space.

Example 1:

Accepted

Runtime 3 ms Beats 99.94% of users with Java

Memory 52.94 MB Beats 87.80% of users with Java

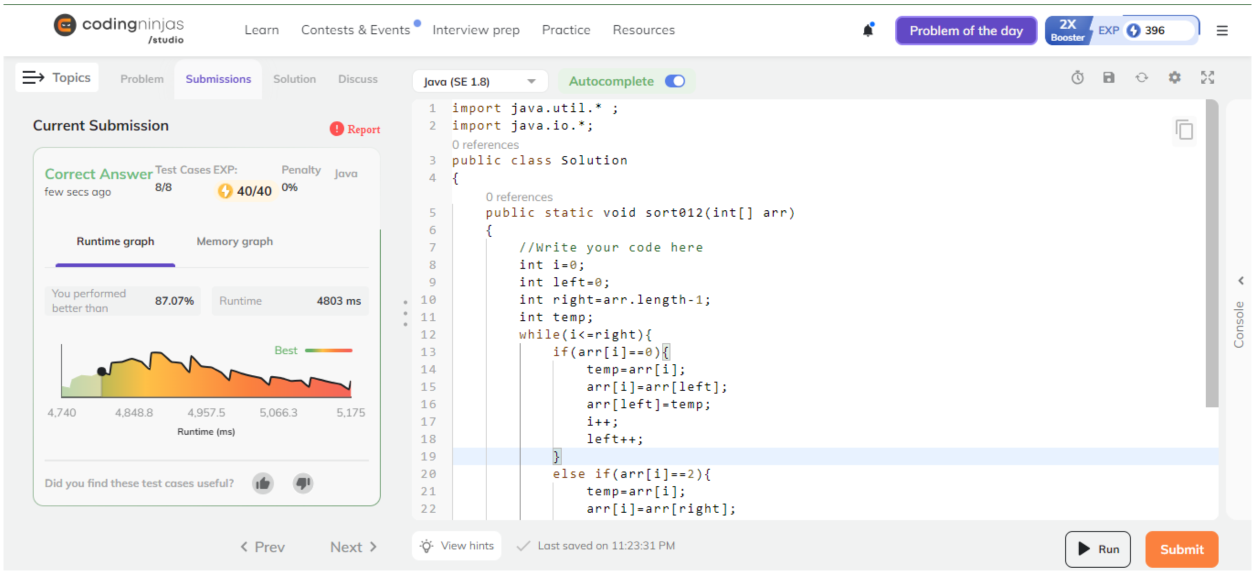
```
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        List<Integer> result=new ArrayList<Integer>();
        int[] temp=new int[nums.length+1];
        for(int i=0;i<nums.length;i++){
            temp[nums[i]]++;
        }

        for(int i=0;i<temp.length;i++){
            if(temp[i]==2){
                result.add(i);
            }
        }
        return result;
    }
}
```

Java Code

```
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        List<Integer> result=new ArrayList<Integer>();
        int[] temp=new int[nums.length+1];
        for(int i=0;i<nums.length;i++){
            temp[nums[i]]++;
        }

        for(int i=0;i<temp.length;i++){
            if(temp[i]==2){
               result.add(i);
            }
        }
        return result;
    }
}
```

Sort 0 1 2 [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/sort-0-1-2_631055?source=youtube&campaign=LoveBabbar_Codestudiovideo1&utm_source=youtube&utm_medium=affiliate&utm_campaign=LoveBabbar_Codestudiovideo1&leftPanelTab=0



Current Submission

Report

Correct Answer few secs ago — Test Cases 8/8 — EXP: 40/40 — Penalty 0% — Java

You performed better than 87.07%

Runtime 4803 ms

Did you find these test cases useful?

Last saved on 11:23:31 PM

```
import java.util.* ;
import java.io.*;
0 references
public class Solution
{
    0 references
    public static void sort012(int[] arr)
    {
        //Write your code here
        int i=0;
        int left=0;
        int right=arr.length-1;
        int temp;
        while(i<=right){
            if(arr[i]==0){
                temp=arr[i];
                arr[i]=arr[left];
                arr[left]=temp;
                i++;
                left++;
            }
            else if(arr[i]==2){
                temp=arr[i];
                arr[i]=arr[right];
```

Java Code

```
import java.util.* ;
import java.io.*;
public class Solution
{
    public static void sort012(int[] arr)
    {
        //Write your code here
        int i=0;
        int left=0;
        int right=arr.length-1;
        int temp;
        while(i<=right){
            if(arr[i]==0){
                temp=arr[i];
                arr[i]=arr[left];
                arr[left]=temp;
                i++;
                left++;
            }
            else if(arr[i]==2){
                temp=arr[i];
                arr[i]=arr[right];
                arr[right]=temp;
                right--;
            }
            else{
                i++;
            }
        }
    }
}
```