Started on	Friday, 16 August 2024, 1:40 PM
State	Finished
Completed on	Friday, 16 August 2024, 2:00 PM
Time taken	20 mins 2 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int m;
    scanf("%d",&m);
    int a[m];
    for(int i=0;i<m;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<m;i++)
    {
        for(int j=i+1;j<m;j++)
        {
           if(a[i]==a[j])
           printf("%d",a[i]);
        }
    }


}
```

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 4-DP-Longest non-decreasing Subsequence

Jump to...

2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity ►

Started on	Friday, 16 August 2024, 2:00 PM
State	Finished
Completed on	Friday, 16 August 2024, 2:33 PM
Time taken	32 mins 35 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int m,r=0;
    scanf("%d",&m);
    int a[m];
    for(int i=0;i<m;i++)
    {
        scanf("%d",&a[i]);
    }
    int s=m*(m-1)/2;
    for(int i=0;i<m;i++)
    {
        r+=a[i];
    }
    printf("%d",r-s);


}
```

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity

Jump to...

3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity ►

Started on	Monday, 18 November 2024, 7:31 PM
State	Finished
Completed on	Monday, 18 November 2024, 9:31 PM
Time taken	2 hours
Marks	1.00/1.00
Grade	**30.00** out of 30.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

· The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int m, n;
        scanf("%d", &m);
        int arr1[m];
        for (int i = 0; i < m; i++)
        scanf("%d", &arr1[i]);
        scanf("%d", &n);
        int arr2[n];
        for (int i = 0; i < n; i++)
        scanf("%d", &arr2[i]);
        int i = 0, j = 0, found = 0;
```

```
        while (i < m && j < n) {
            if (arr1[i] < arr2[j])
            i++;
            else if (arr1[i] > arr2[j])
            j++;
            else {
                if (found == 0)
                found = 1;
                else
                printf(" ");
                printf("%d", arr1[i]);
                i++;
                j++;
            }
        }
    }
}
```

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity

Jump to...

4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity ►

Started on	Monday, 18 November 2024, 7:32 PM
State	Finished
Completed on	Monday, 18 November 2024, 9:32 PM
Time taken	2 hours
Marks	1.00/1.00
Grade	**30.00** out of 30.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

· The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int m, n;
        scanf("%d", &m);
        int arr1[m];
        for (int i = 0; i < m; i++)
        scanf("%d", &arr1[i]);
        scanf("%d", &n);
        int arr2[n];
        for (int i = 0; i < n; i++)
        scanf("%d", &arr2[i]);
        int i = 0, j = 0, found = 0;
```

```
        while (i < m && j < n) {
            if (arr1[i] < arr2[j])
            i++;
            else if (arr1[i] > arr2[j])
            j++;
            else {
                if (found == 0)
                found = 1;
                else
                printf(" ");
                printf("%d", arr1[i]);
                i++;
                j++;
            }
        }
    }
}
```

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity

Jump to...

5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity ►

Started on	Friday, 25 October 2024, 1:46 PM
State	Finished
Completed on	Friday, 25 October 2024, 3:00 PM
Time taken	1 hour 13 mins
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as 5 - 1 = 4

So Return 1.

For example:

Input	Result
3 1 3 5 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int array()
{
    int n,k,r=0;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    scanf("%d",&arr[i]);
    scanf("%d",&k);
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(i!=j && (arr[i]-arr[j] )== k)
            {
                r=1;
            }
        }
    }
    return r;
}
int main()
{
    printf("%d",array());
}
```

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

Jump to...

6-Pair with Difference -O(n) Time Complexity,O(1) Space Complexity ►

Started on	Monday, 18 November 2024, 7:33 PM
State	Finished
Completed on	Tuesday, 19 November 2024, 10:32 PM
Time taken	1 day 2 hours
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as 5 - 1 = 4

So Return 1.

For example:

Input	Result
3 1 3 5 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>
int res()
{
    int n,k;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
     scanf("%d",&arr[i]);
    }
    scanf("%d",&k);
   int j=1,f=1;
   int i=0;
   if(k==0)
    return 0;
   for( i=0;i<n;i++)
   {
       if(abs(arr[i]-arr[j])==k  &&i!=j)
       {
           return f;
       }
       if(i==n-1)
       {
           i=0;
           j++;
       }
   }
   return 0;

}
int main()
{
```

```
34    if(res())
35     printf("1");
36    else
37     printf("0");
38  }
```

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity

Jump to...