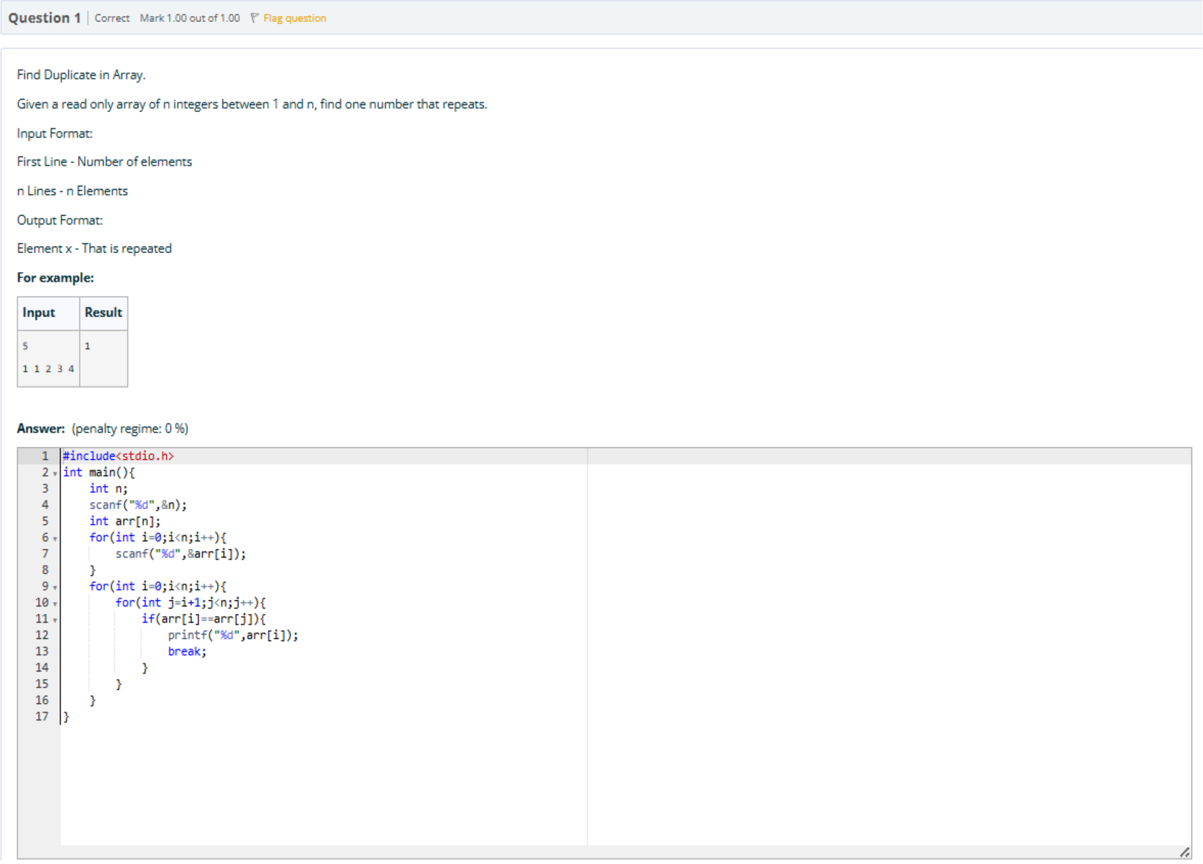
DAA WEEK-COMPETITIVE PROGRAMMING

1)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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

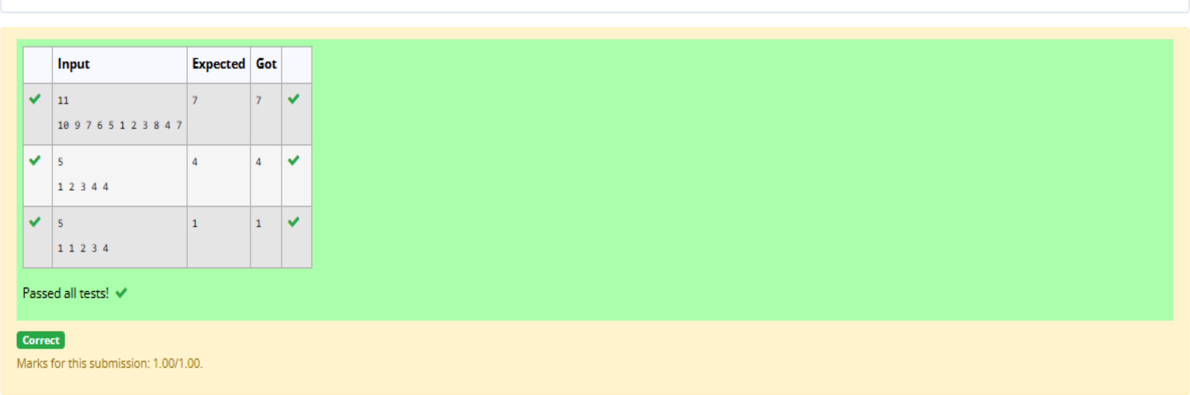
For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

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

OUTPUT:



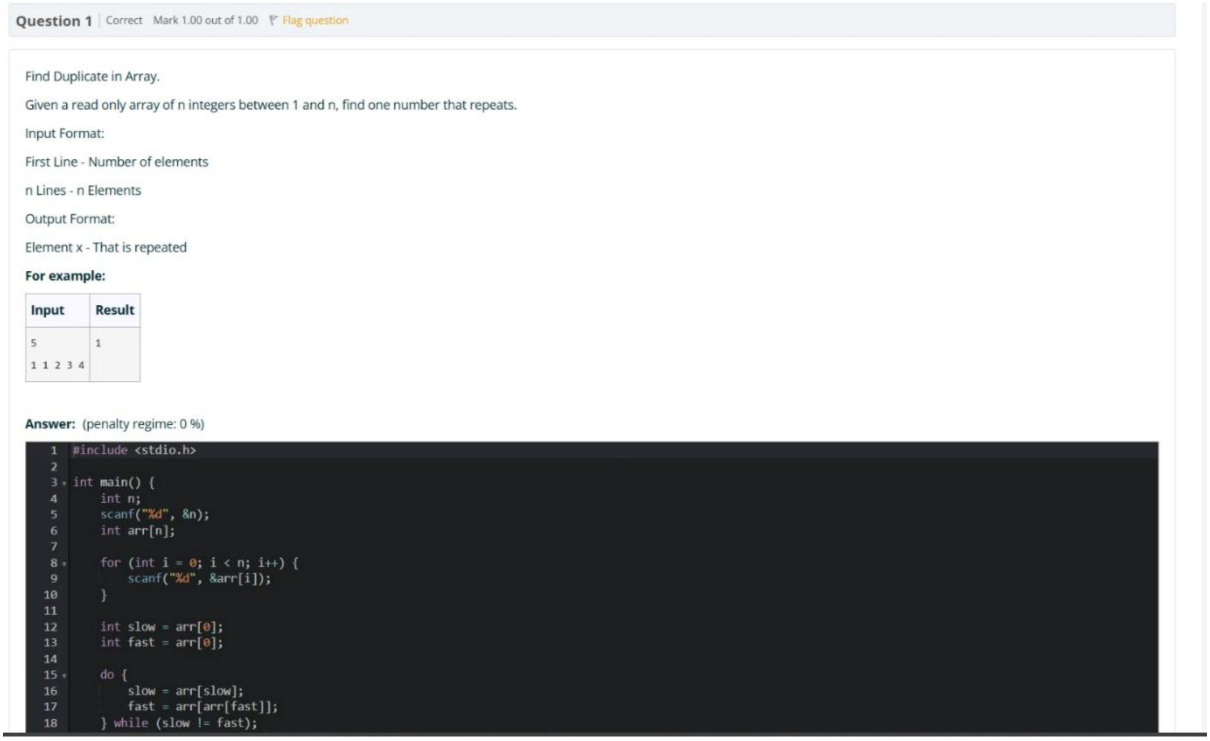
	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

2)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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:

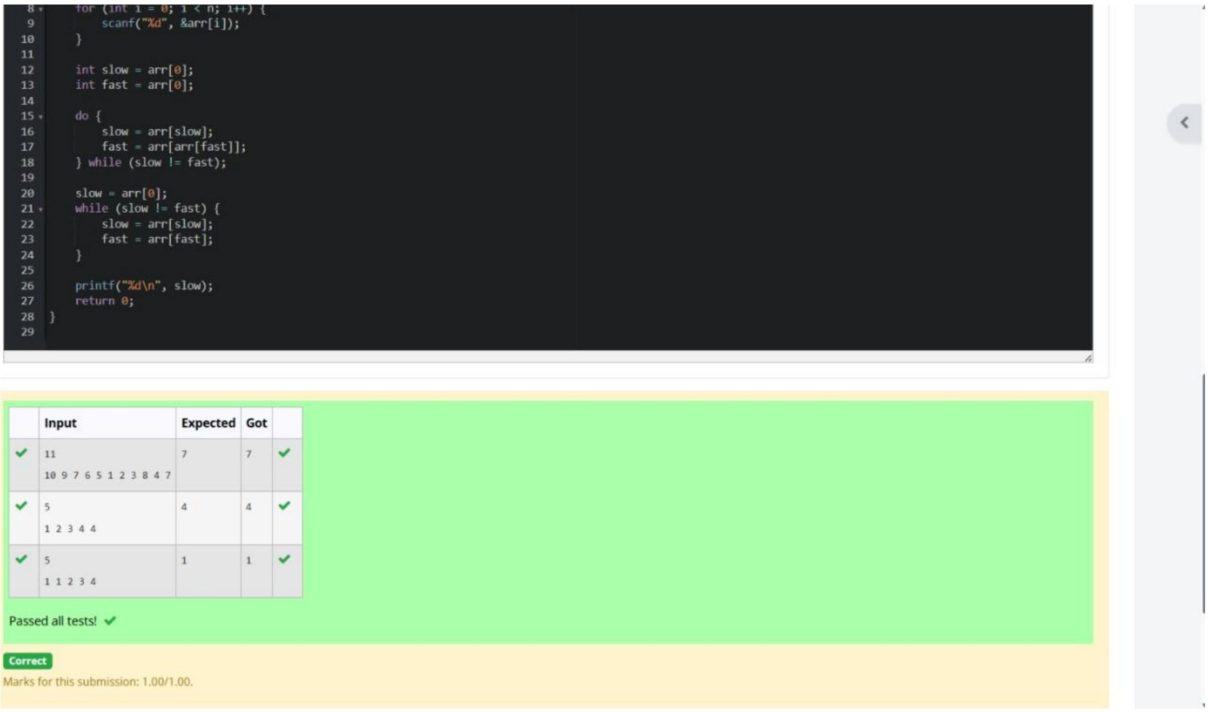
Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int slow = arr[0];
    int fast = arr[0];

    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);
```



```c
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int slow = arr[0];
    int fast = arr[0];

    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);
```

```c

    slow = arr[0];
    while (slow != fast) {
        slow = arr[slow];
        fast = arr[fast];
    }

    printf("%d\n", slow);
    return 0;
}

```

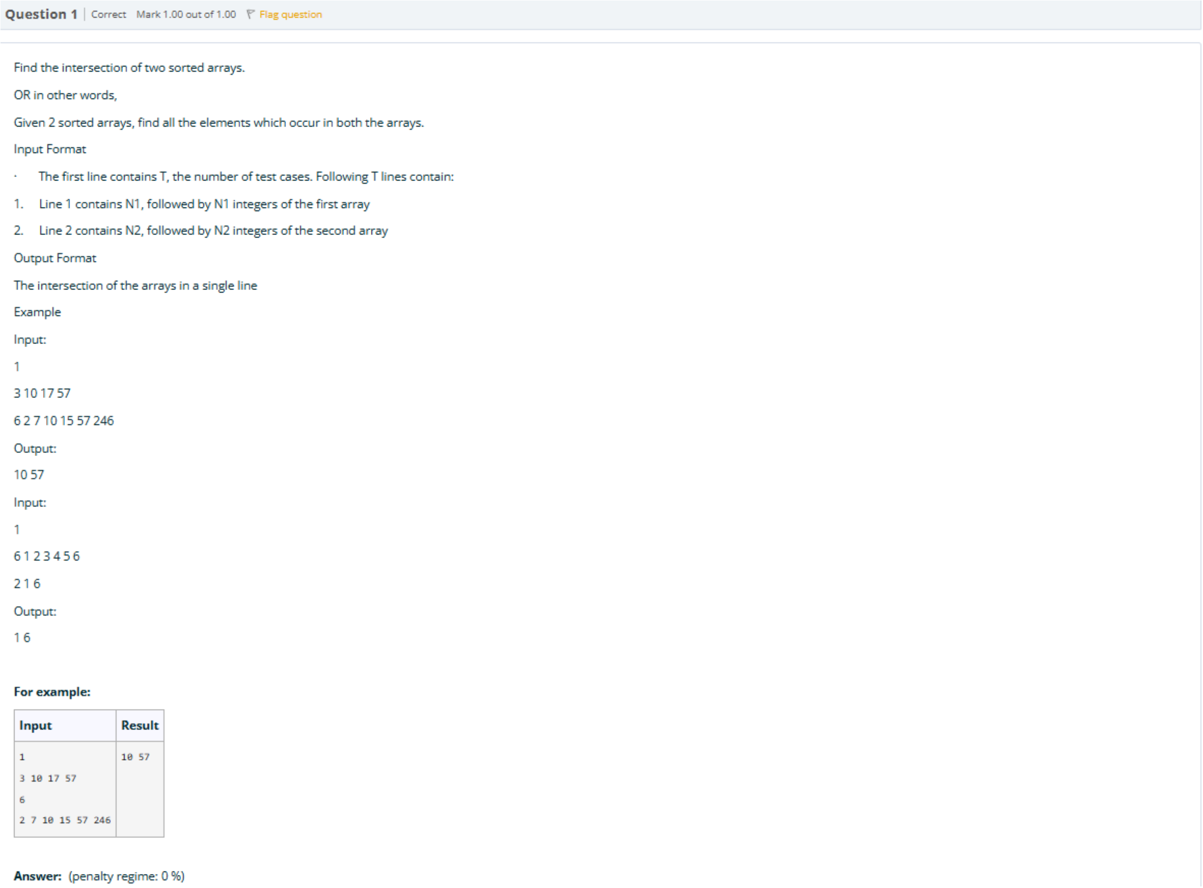
	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

3)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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

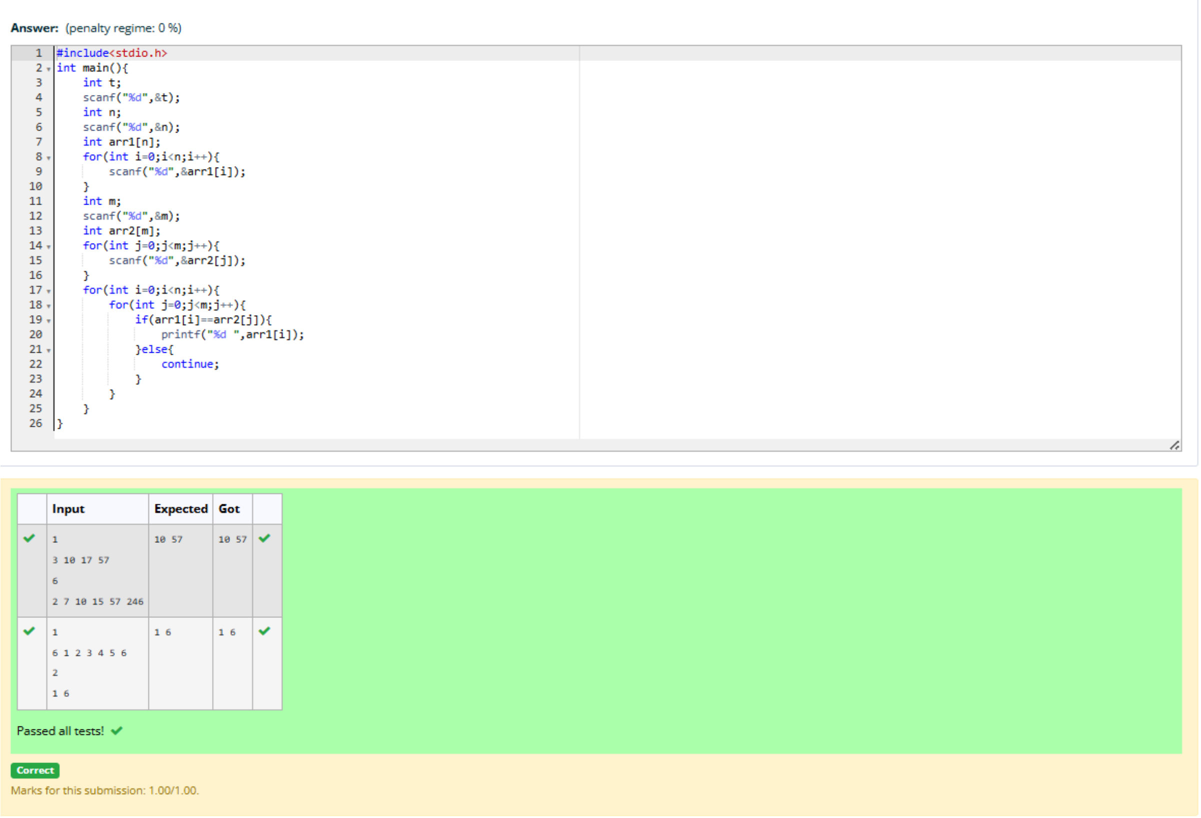
For example:

Input	Result

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

Answer: (penalty regime: 0 %)

OUTPUT:



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    int n;
    scanf("%d",&n);
    int arr1[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr1[i]);
    }
    int m;
    scanf("%d",&m);
    int arr2[m];
    for(int j=0;j<m;j++){
        scanf("%d",&arr2[j]);
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(arr1[i]==arr2[j]){
                printf("%d ",arr1[i]);
            }else{
                continue;
            }
        }
    }
}
```

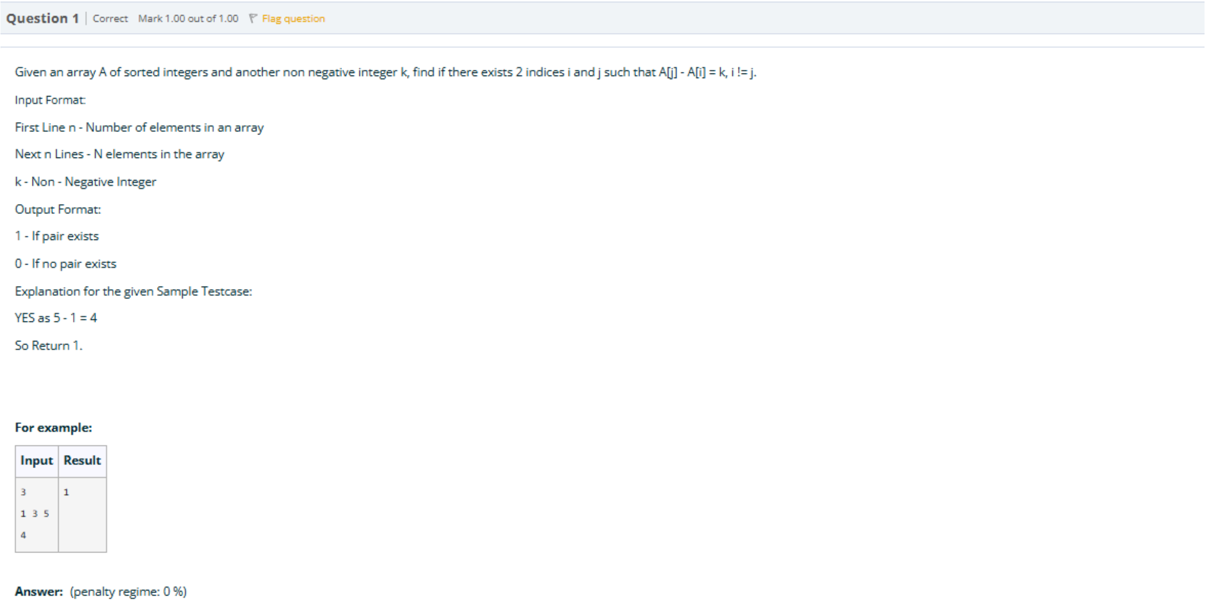
	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

4)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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.

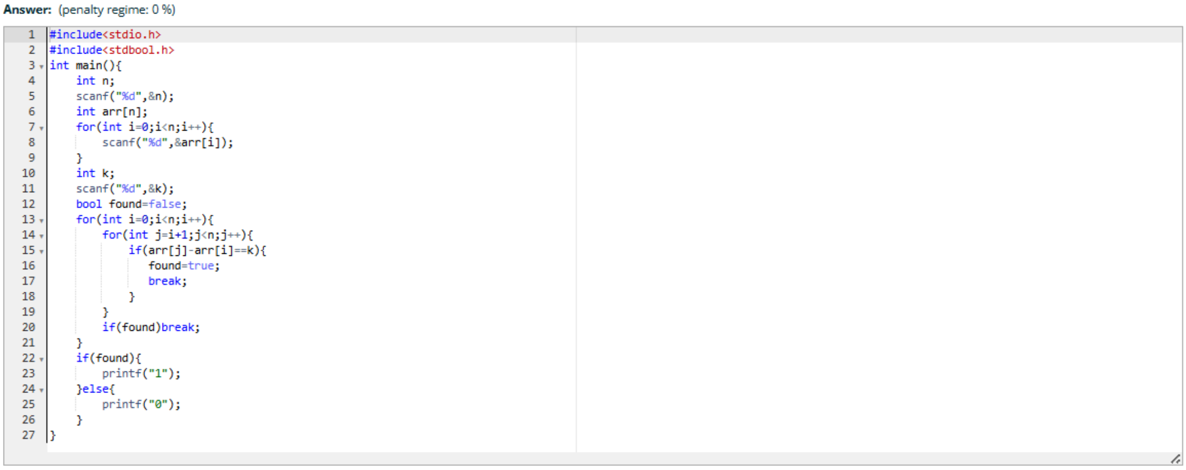
For example:

Input	Result

3
1 3 5
4
	1

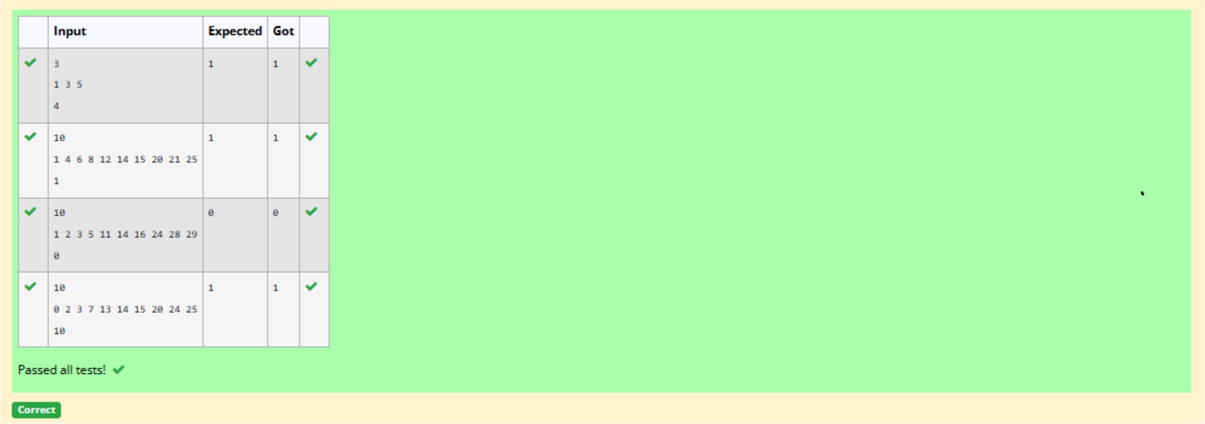
Answer: (penalty regime: 0 %)

OUTPUT:



Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdbool.h>
int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int k;
    scanf("%d",&k);
    bool found=false;
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if(arr[j]-arr[i]==k){
                found=true;
                break;
            }
        }
        if(found)break;
    }
    if(found){
        printf("1");
    }else{
        printf("0");
    }
}
```

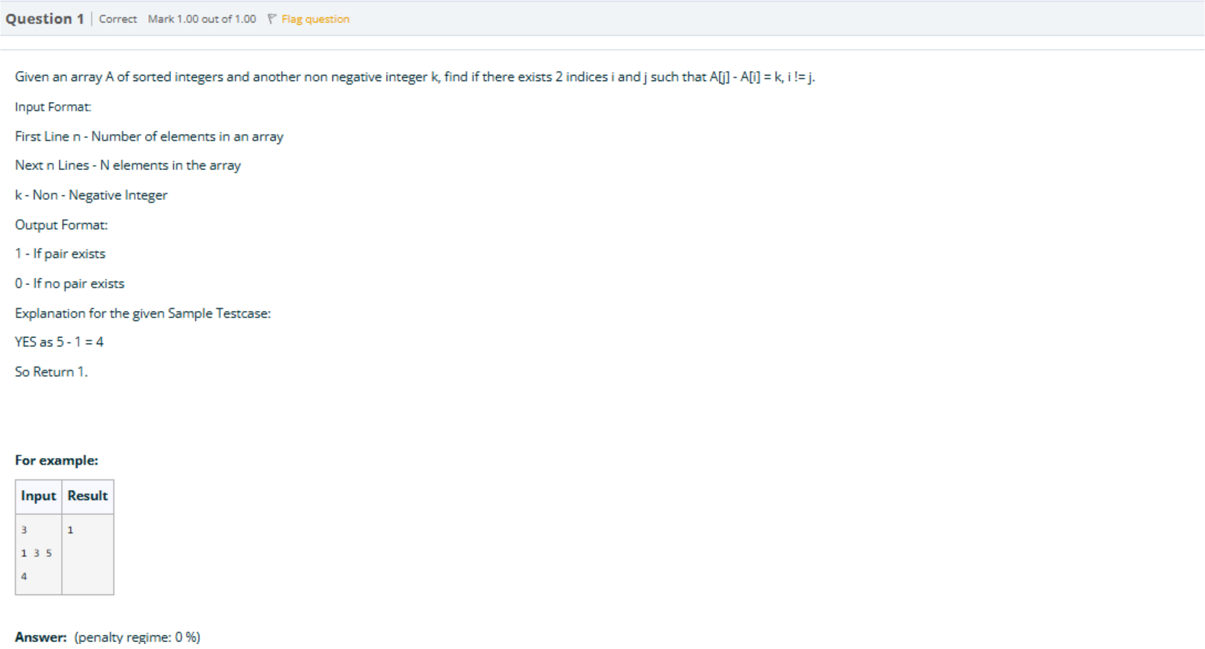


	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓
✓	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✓
✓	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✓
✓	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✓

Passed all tests! ✓

Correct

5)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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.

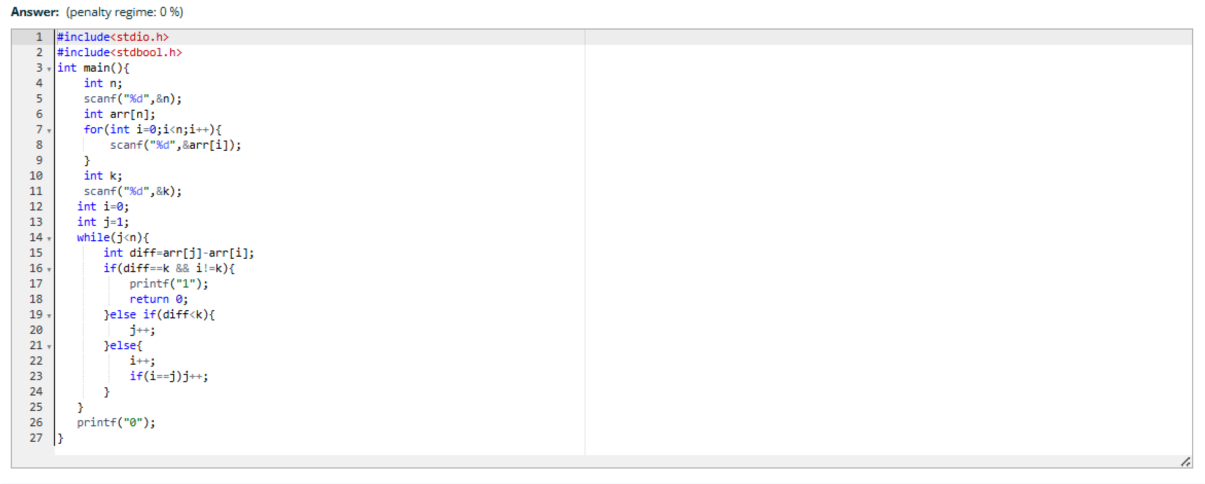
For example:

Input	Result

3
1 3 5
4
	1

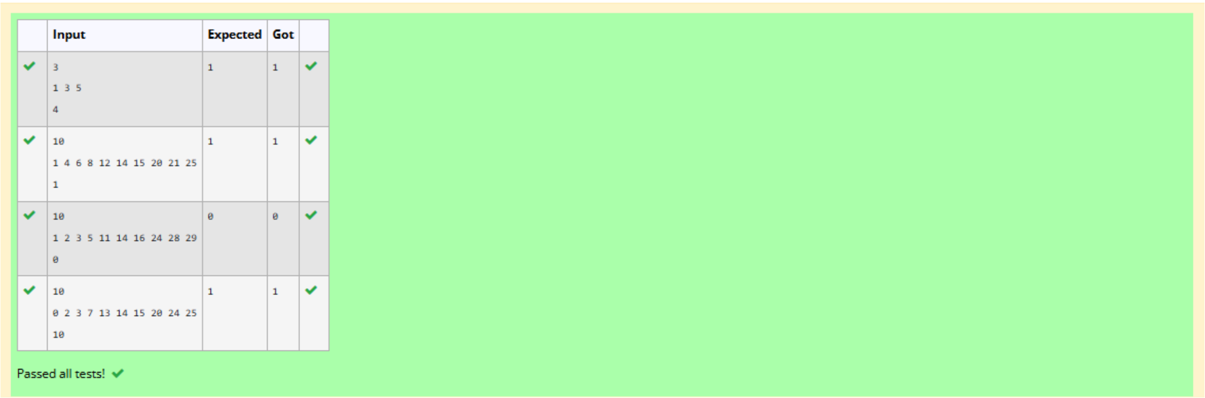
Answer: (penalty regime: 0 %)

OUTPUT:



Answer: (penalty regime: 0 %)

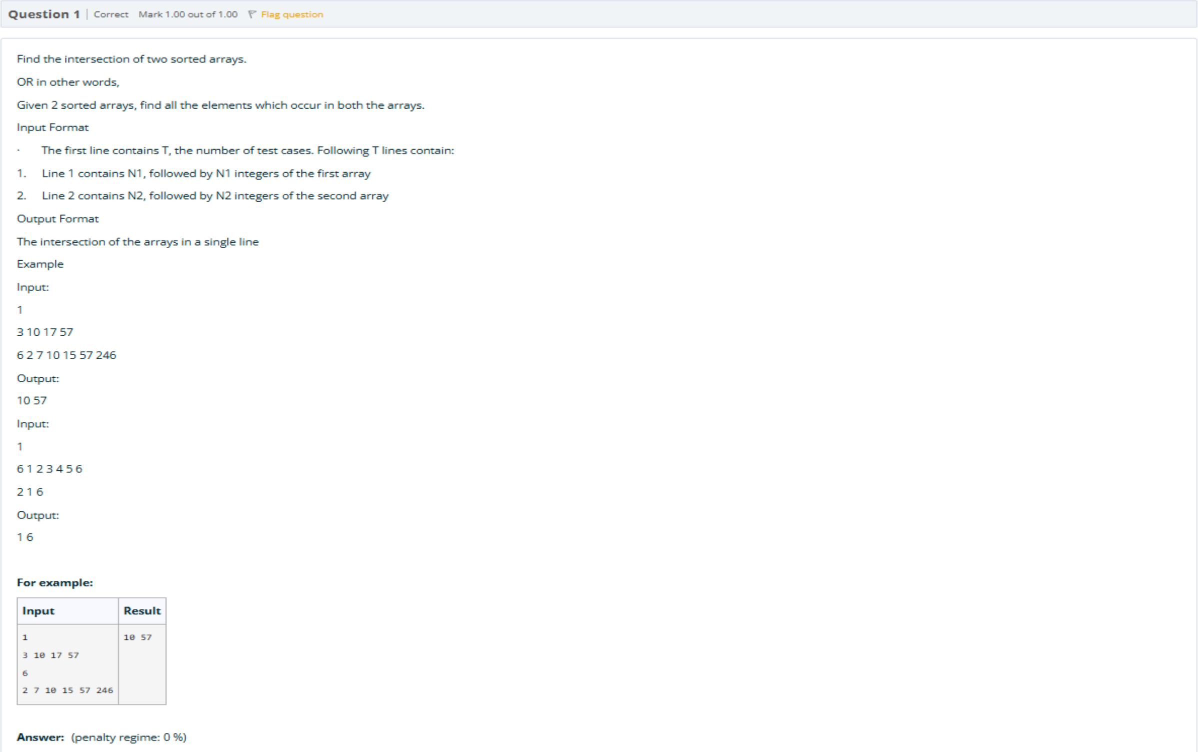
```c
#include<stdio.h>
#include<stdbool.h>
int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int k;
    scanf("%d",&k);
   int i=0;
   int j=1;
   while(j<n){
       int diff=arr[j]-arr[i];
       if(diff==k && i!=k){
        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



	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! ✓

6)



Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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

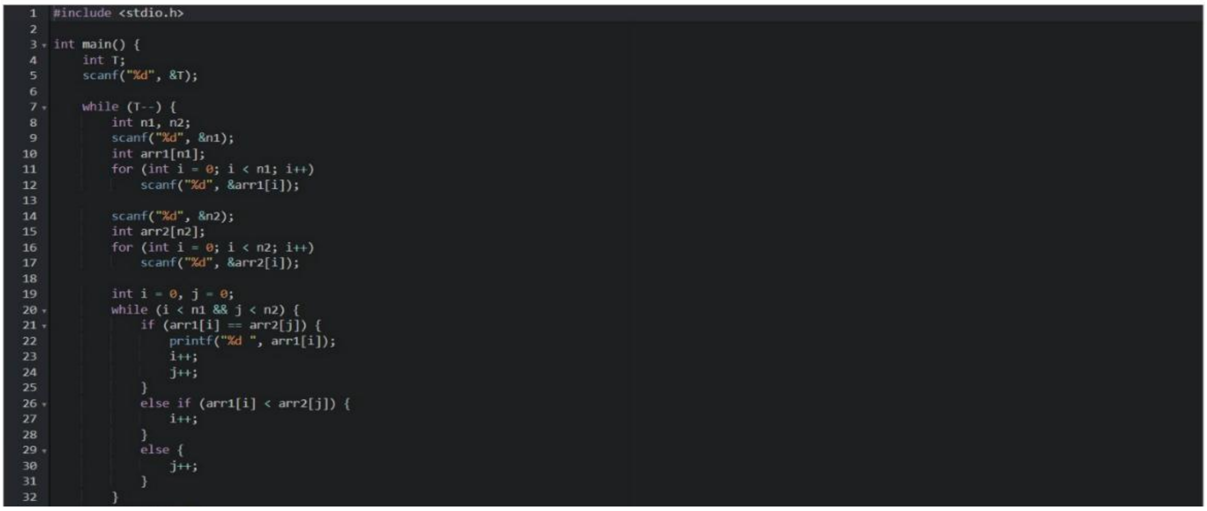
For example:

Input	Result

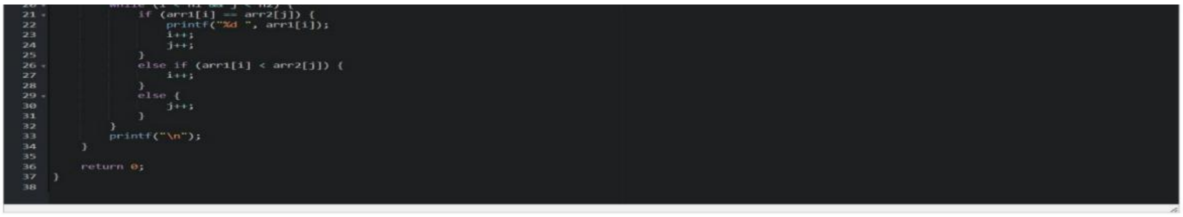
1
3 10 17 57
6
2 7 10 15 57 246
	10 57

Answer: (penalty regime: 0 %)

Answer: (penalty regime: 0 %)

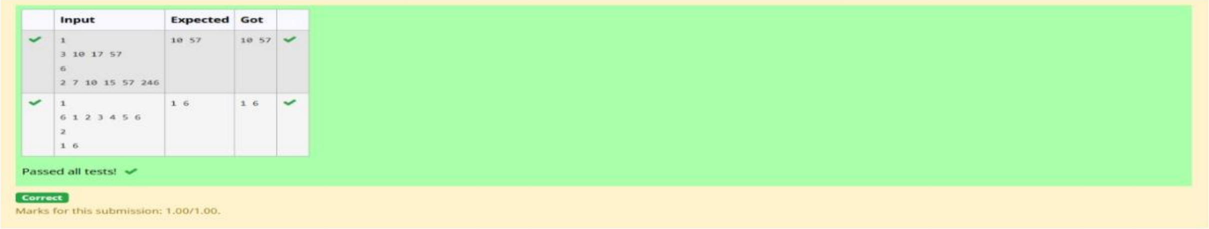


```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++)
            scanf("%d", &arr1[i]);

        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++)
            scanf("%d", &arr2[i]);

        int i = 0, j = 0;
        while (i < n1 && j < n2) {
            if (arr1[i] == arr2[j]) {
                printf("%d ", arr1[i]);
                i++;
                j++;
            }
            else if (arr1[i] < arr2[j]) {
                i++;
            }
            else {
                j++;
            }
        }
```



```c
            if (arr1[i] == arr2[j]) {
                printf("%d ", arr1[i]);
                i++;
                j++;
            }
            else if (arr1[i] < arr2[j]) {
                i++;
            }
            else {
                j++;
            }
        }
```

```c
        printf("\n");
    }

    return 0;
}
```



	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.