COMPETITIVE PROGRAMMING:-

1) 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

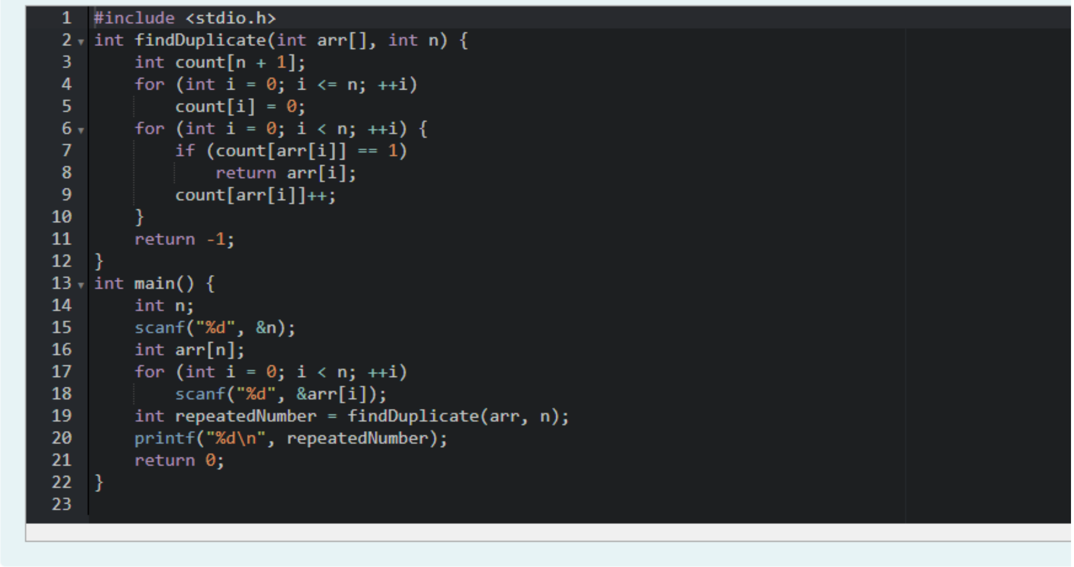
n Lines - n Elements

Output Format:

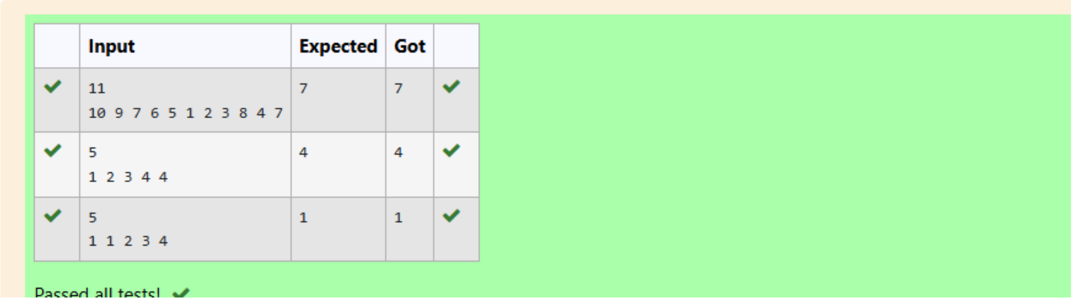
Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1



```c
#include <stdio.h>
int findDuplicate(int arr[], int n) {
    int count[n + 1];
    for (int i = 0; i <= n; ++i)
        count[i] = 0;
    for (int i = 0; i < n; ++i) {
        if (count[arr[i]] == 1)
            return arr[i];
        count[arr[i]]++;
    }
    return -1;
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; ++i)
        scanf("%d", &arr[i]);
    int repeatedNumber = findDuplicate(arr, n);
    printf("%d\n", repeatedNumber);
    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! ✓

2) 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

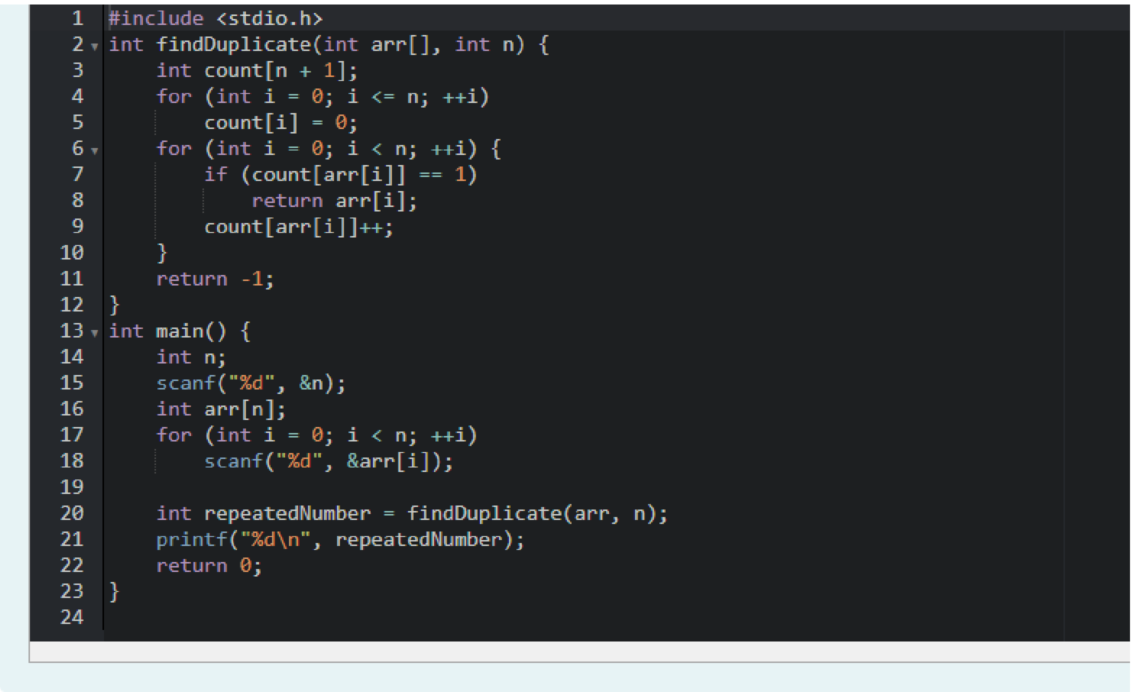
n Lines - n Elements

Output Format:

Element x - That is repeated

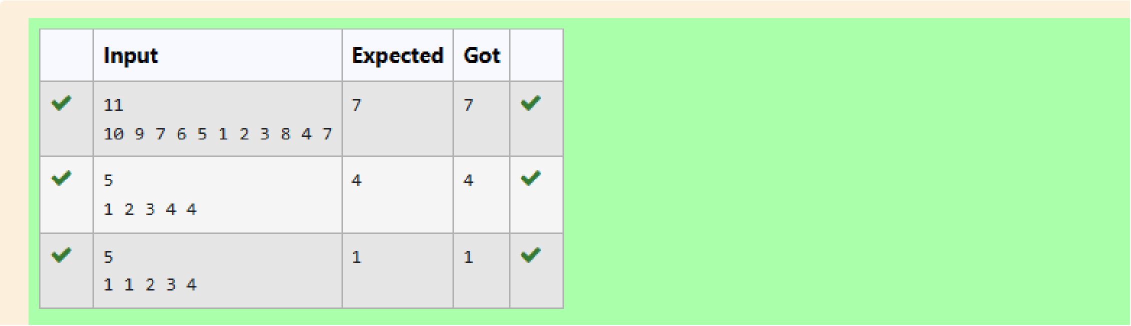
For example:

Input	Result
5 1 1 2 3 4	1



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

    int repeatedNumber = findDuplicate(arr, n);
    printf("%d\n", repeatedNumber);
    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	✓

3) 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:

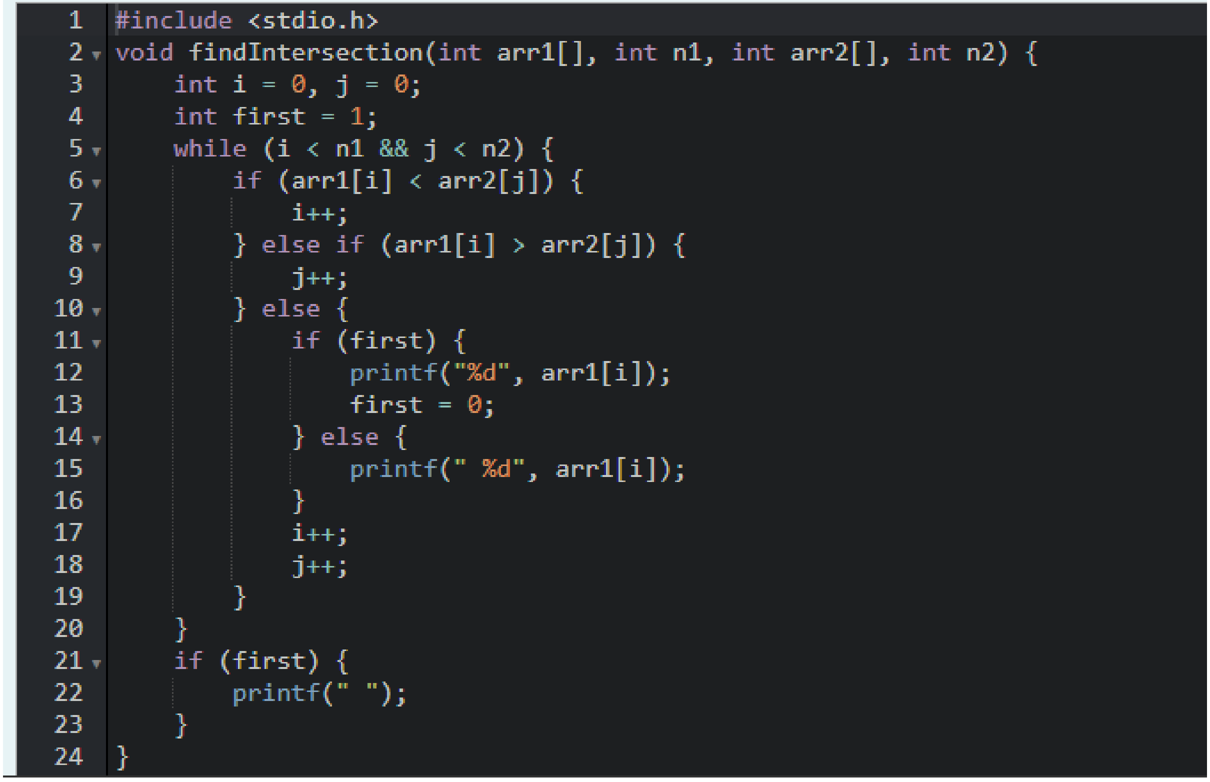
1 6

For example:

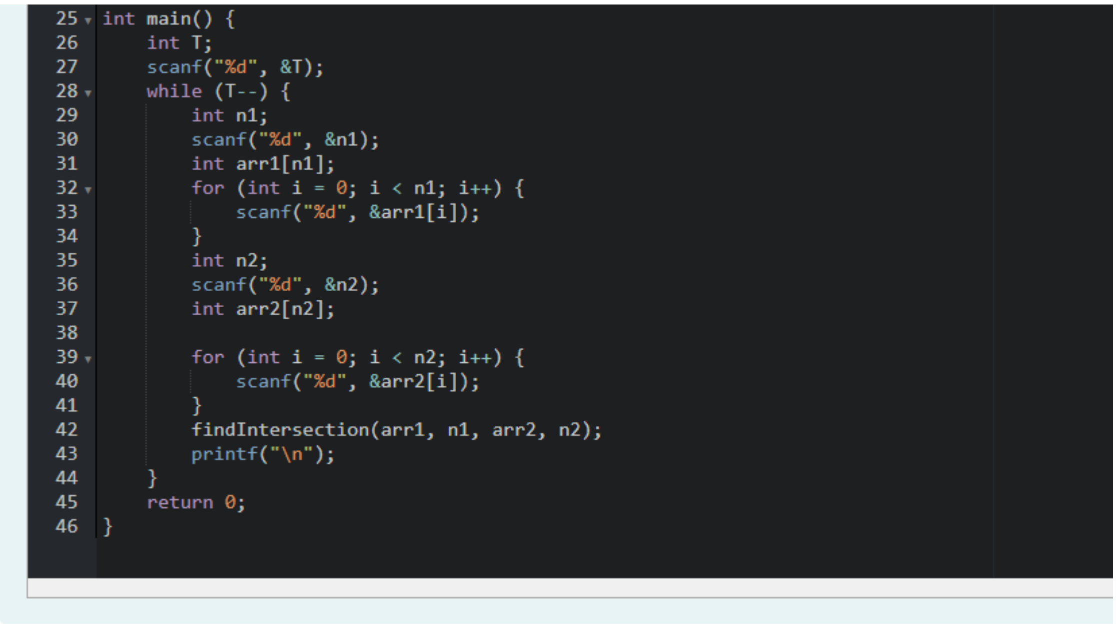
Input	**Result**

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

Input	Result

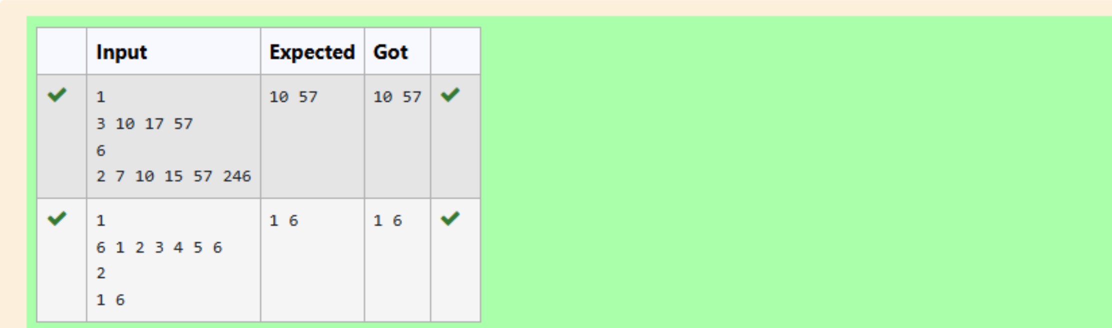


```c
#include <stdio.h>
void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int first = 1;
    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr1[i] > arr2[j]) {
            j++;
        } else {
            if (first) {
                printf("%d", arr1[i]);
                first = 0;
            } else {
                printf(" %d", arr1[i]);
            }
            i++;
            j++;
        }
    }
    if (first) {
        printf(" ");
    }
}
```



```c
int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int n1;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }
        int n2;
        scanf("%d", &n2);
        int arr2[n2];

        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }
        findIntersection(arr1, n1, arr2, n2);
        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	✓

4) 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

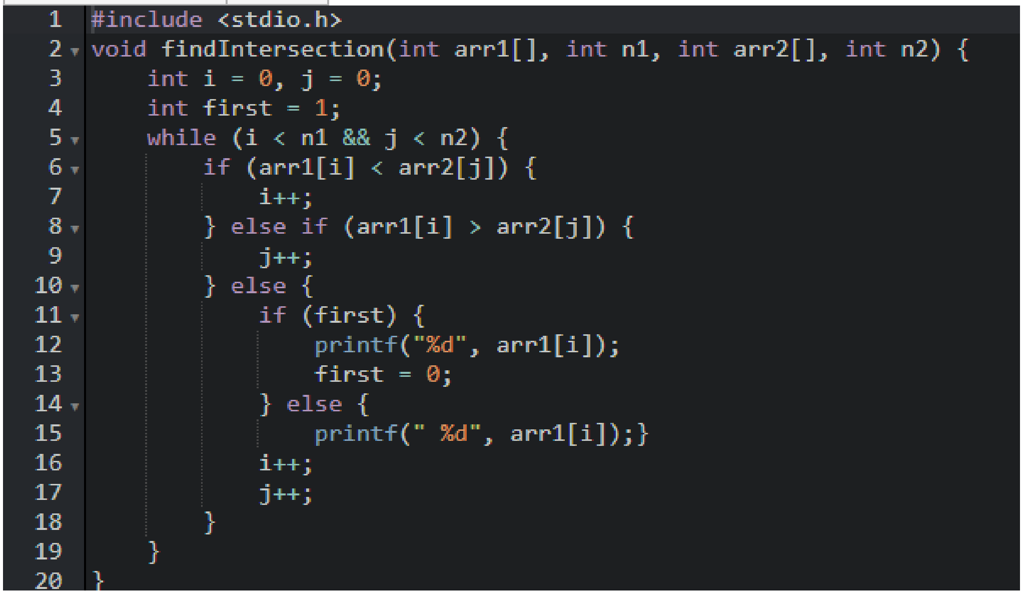
Output:

1 6

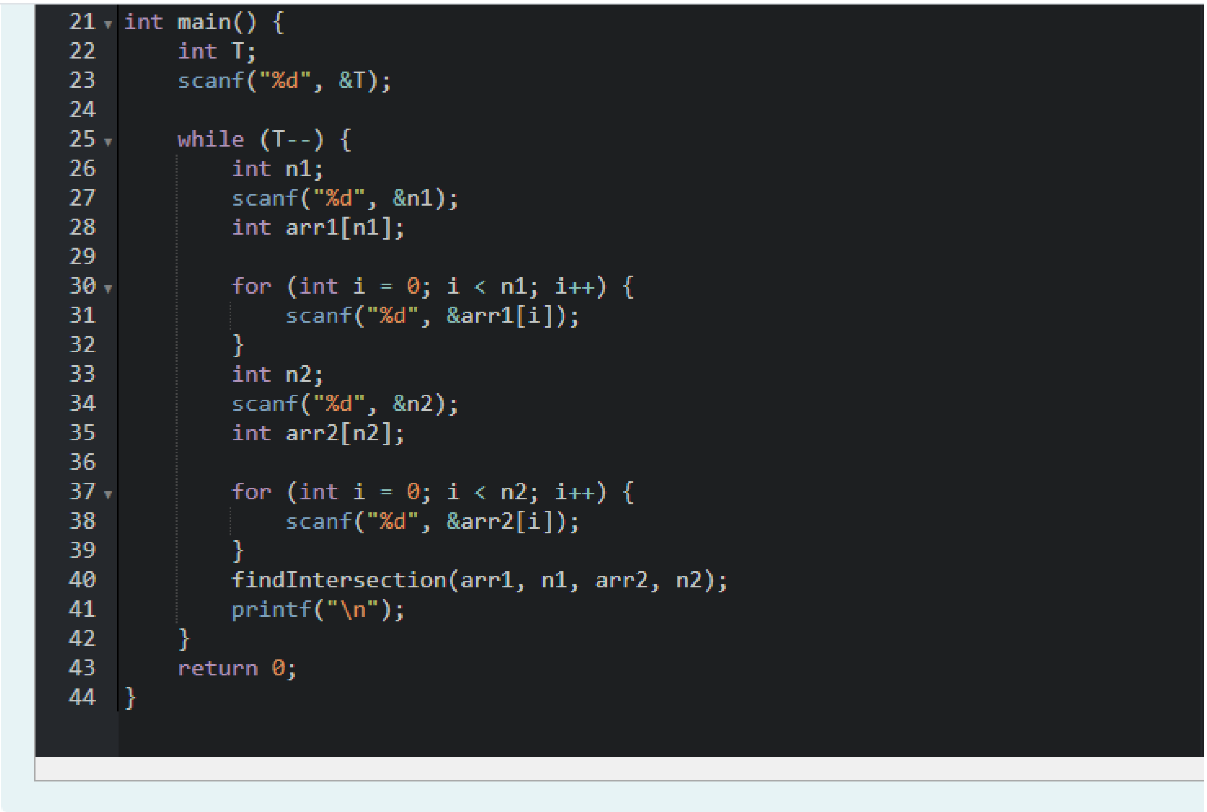
For example:

Input	**Result**

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



```c
#include <stdio.h>
void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int first = 1;
    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr1[i] > arr2[j]) {
            j++;
        } else {
            if (first) {
                printf("%d", arr1[i]);
                first = 0;
            } else {
                printf(" %d", arr1[i]);}
            i++;
            j++;
        }
    }
}
```

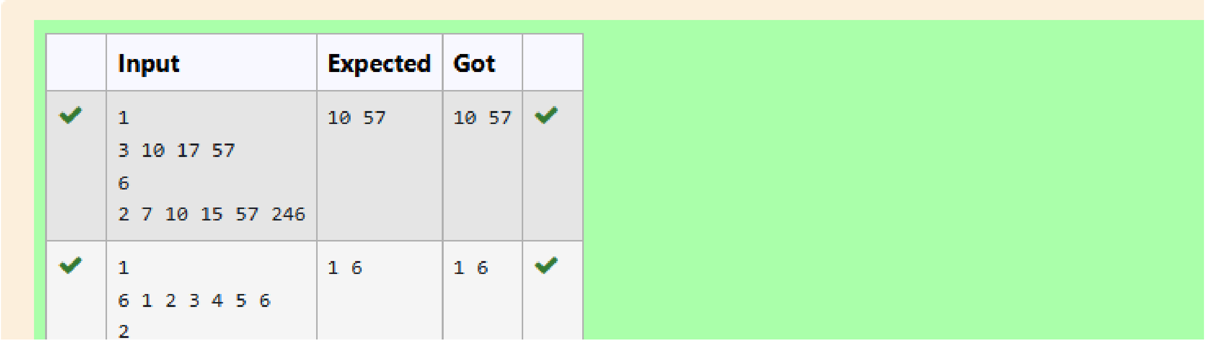


```c
int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1;
        scanf("%d", &n1);
        int arr1[n1];

        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }
        int n2;
        scanf("%d", &n2);
        int arr2[n2];

        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }
        findIntersection(arr1, n1, arr2, n2);
        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	✓

5) 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:

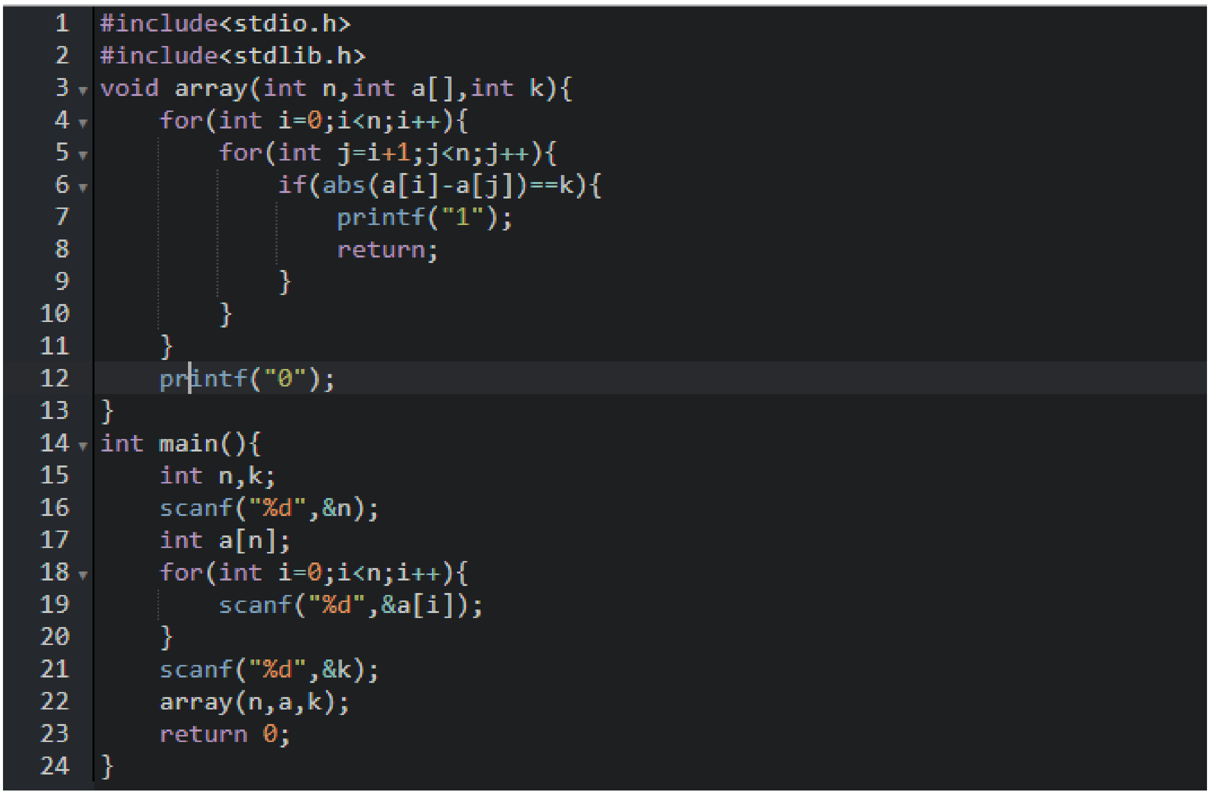
YES as 5 - 1 = 4

So Return 1.

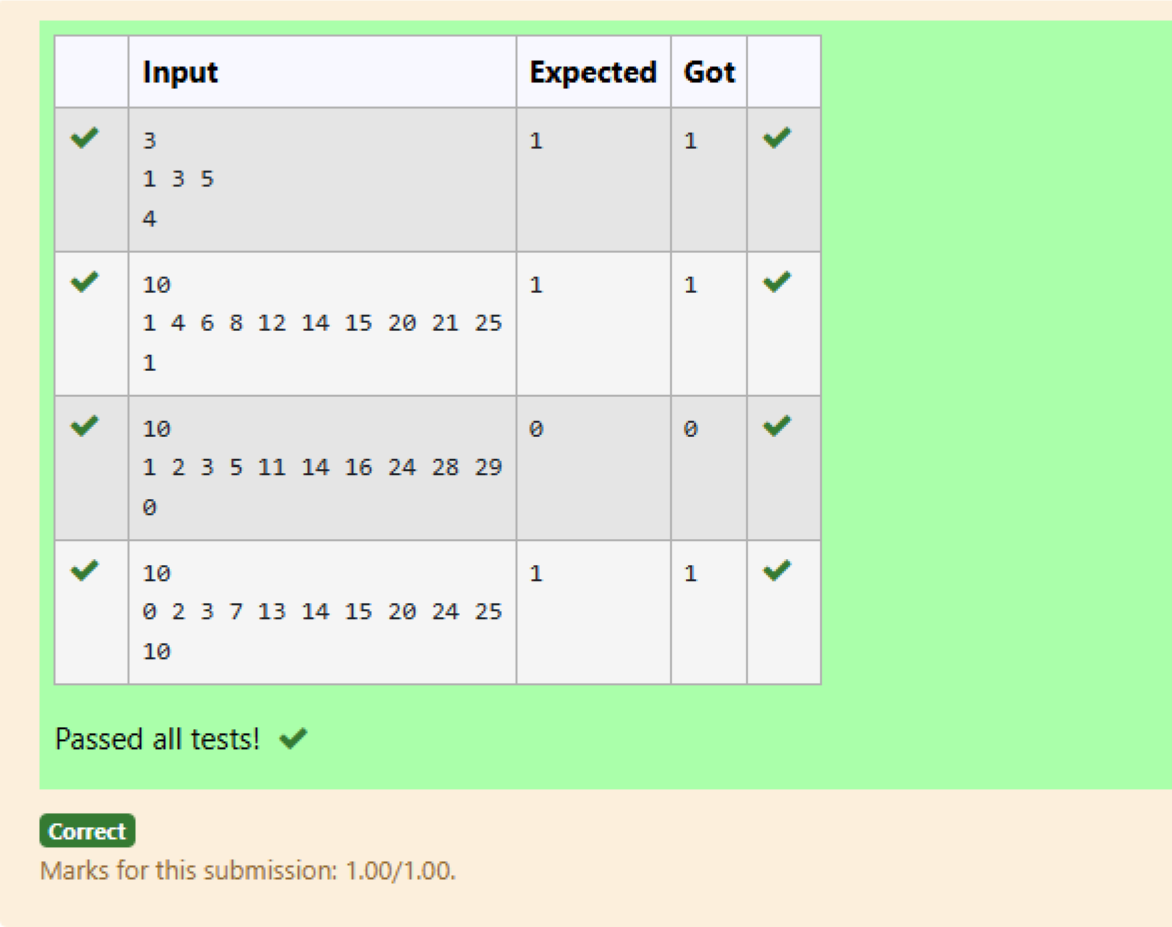
For example:

Input	Result

3
1 3 5
4
	1



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

Marks for this submission: 1.00/1.00.

6) 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:

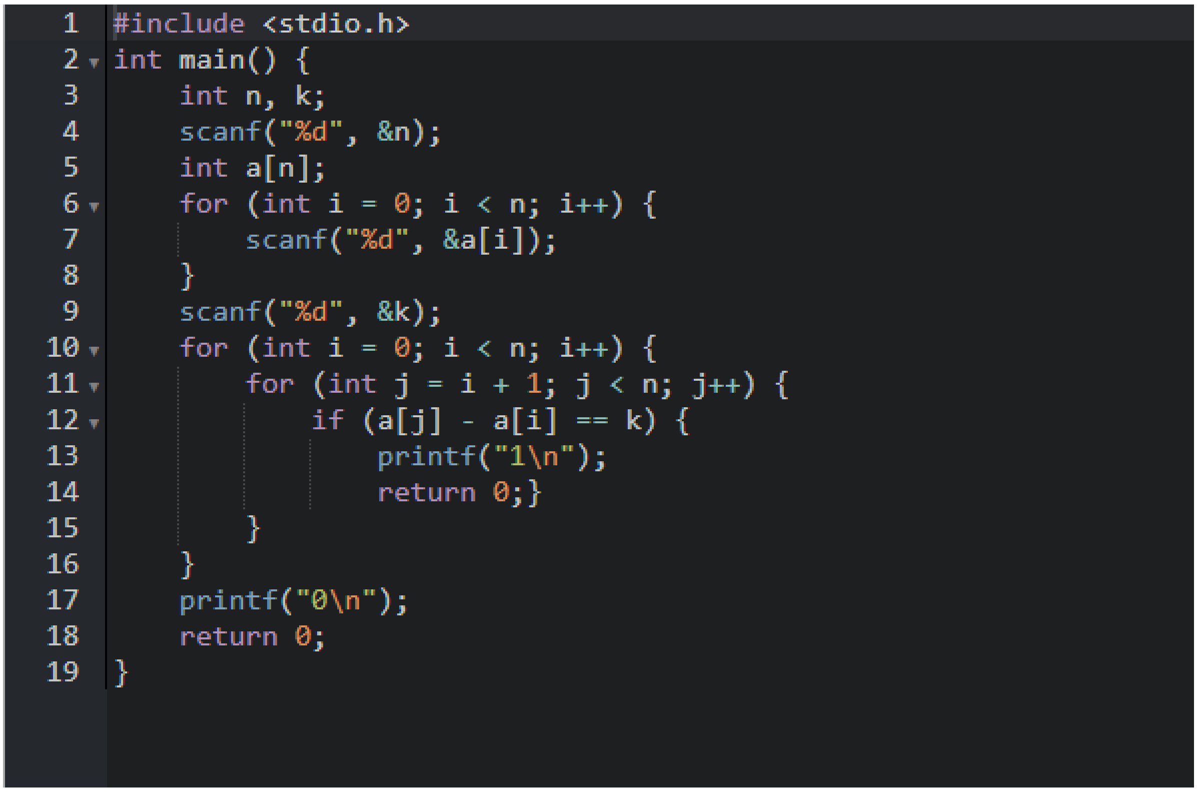
YES as 5 - 1 = 4

So Return 1.

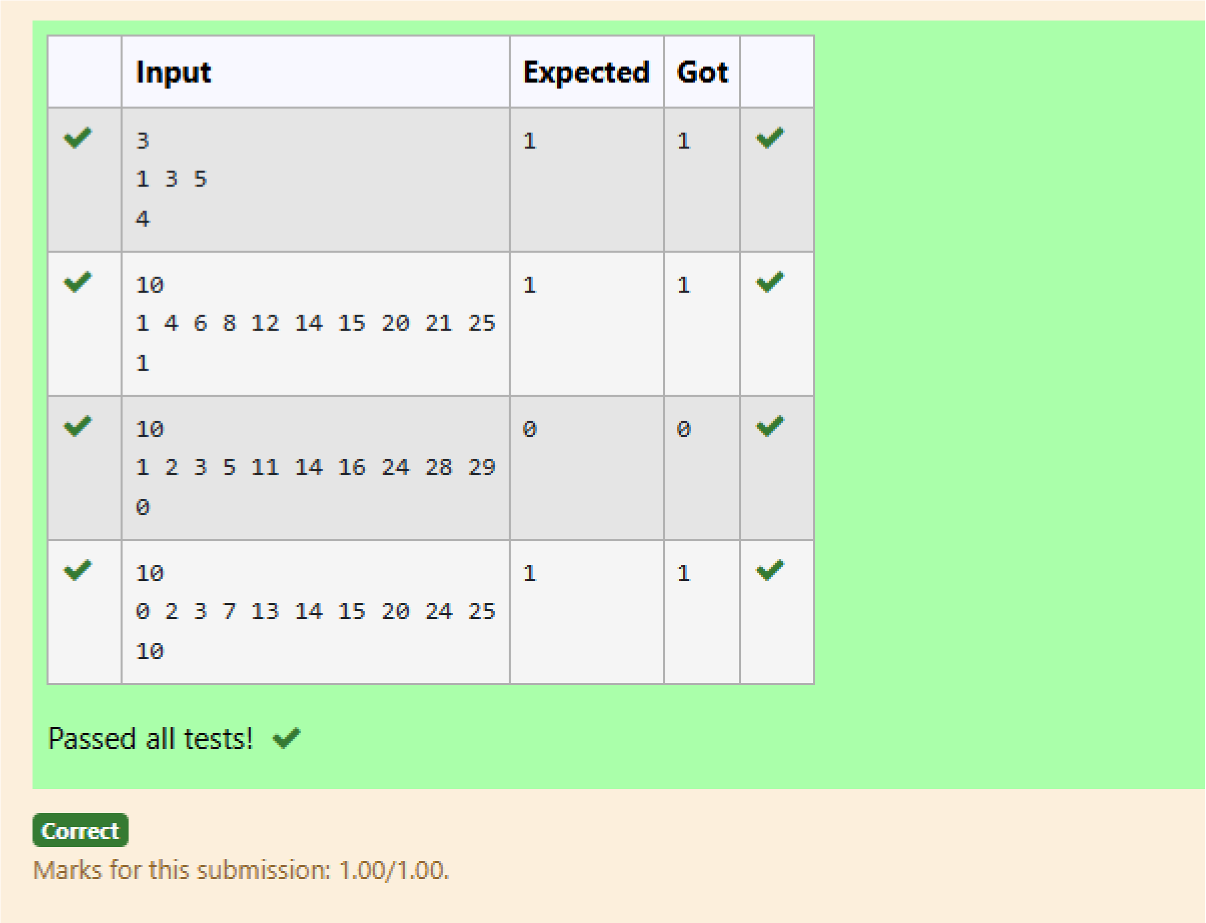
For example:

Input	Result

3
1 3 5
4
	1



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

Marks for this submission: 1.00/1.00.