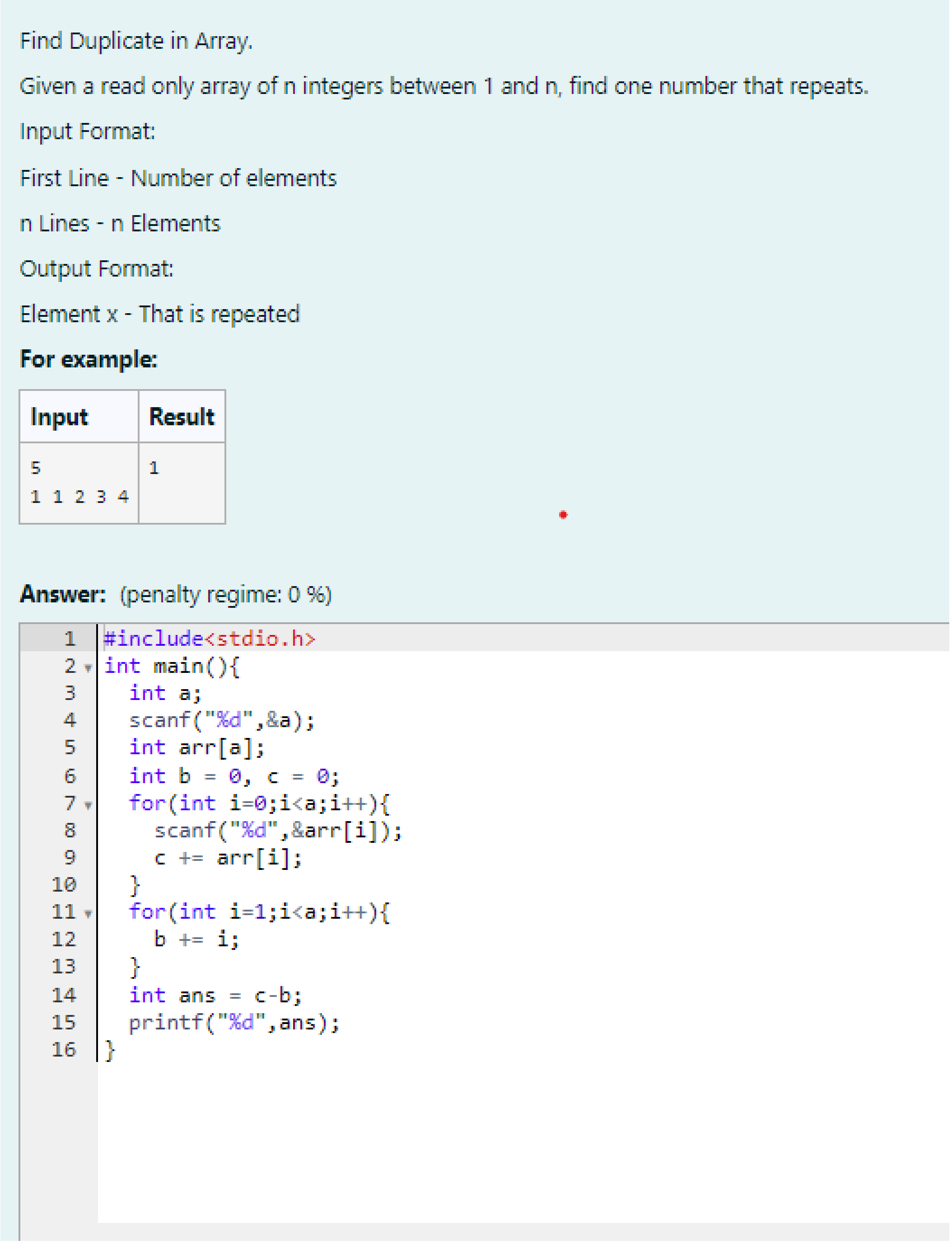


Find Duplicate in Array.

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:

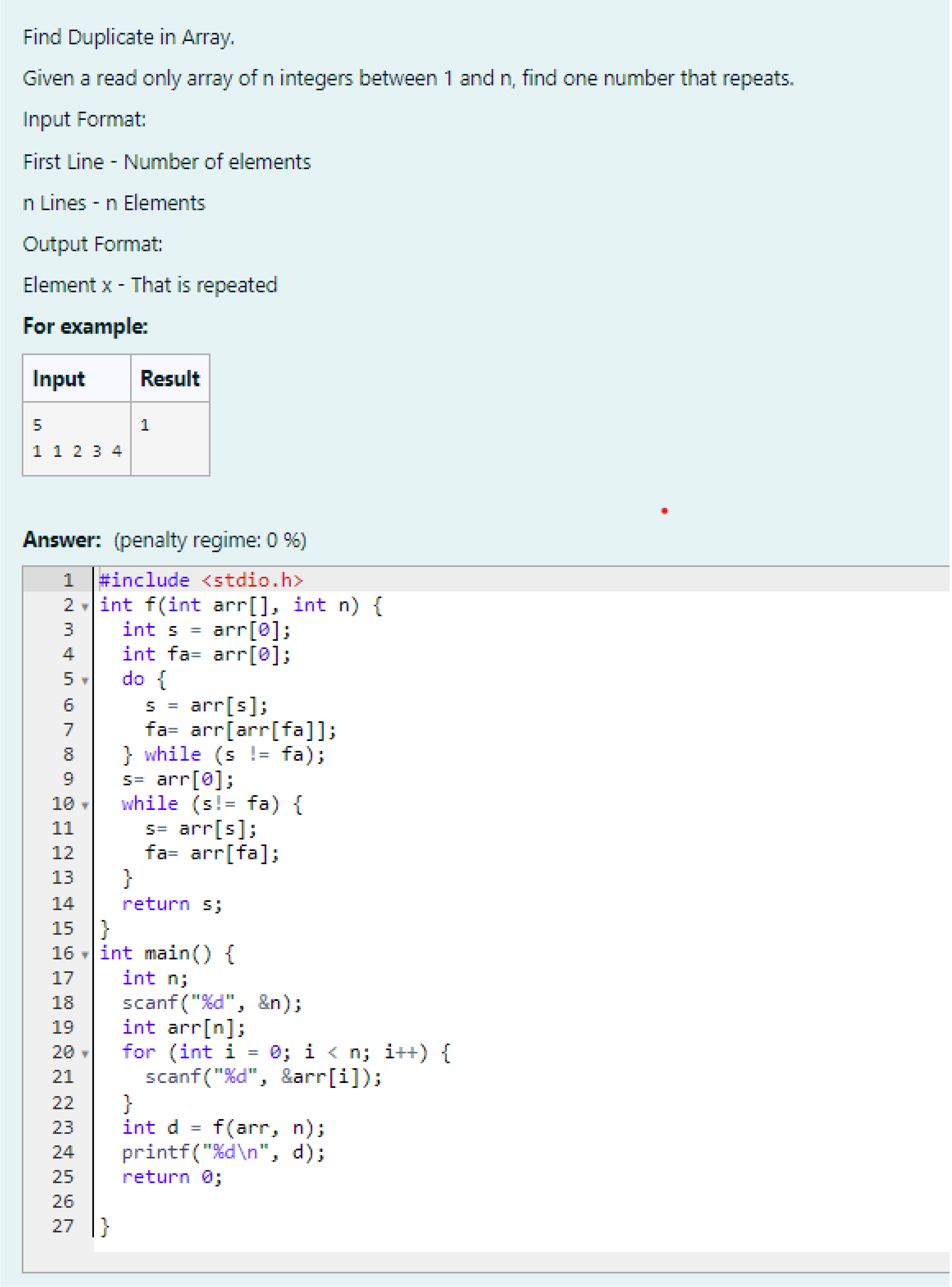
Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
  int a;
  scanf("%d",&a);
  int arr[a];
  int b = 0, c = 0;
  for(int i=0;i<a;i++){
    scanf("%d",&arr[i]);
    c += arr[i];
  }
  for(int i=1;i<a;i++){
    b += i;
  }
  int ans = c-b;
  printf("%d",ans);
}
```



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

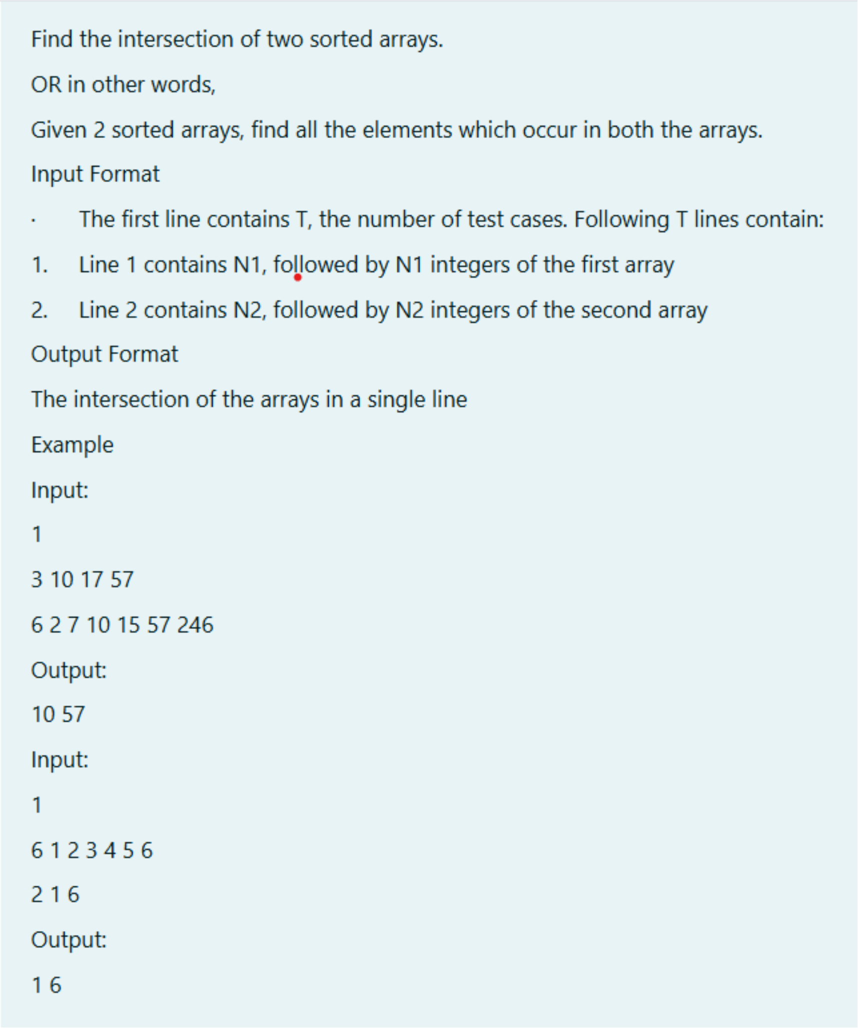
For example:

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int f(int arr[], int n) {
  int s = arr[0];
  int fa= arr[0];
  do {
    s = arr[s];
    fa= arr[arr[fa]];
  } while (s != fa);
  s= arr[0];
  while (s!= fa) {
    s= arr[s];
    fa= arr[fa];
  }
  return s;
}
int main() {
  int n;
  scanf("%d", &n);
  int arr[n];
  for (int i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
  }
  int d = f(arr, n);
  printf("%d\n", d);
  return 0;

}
```



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:

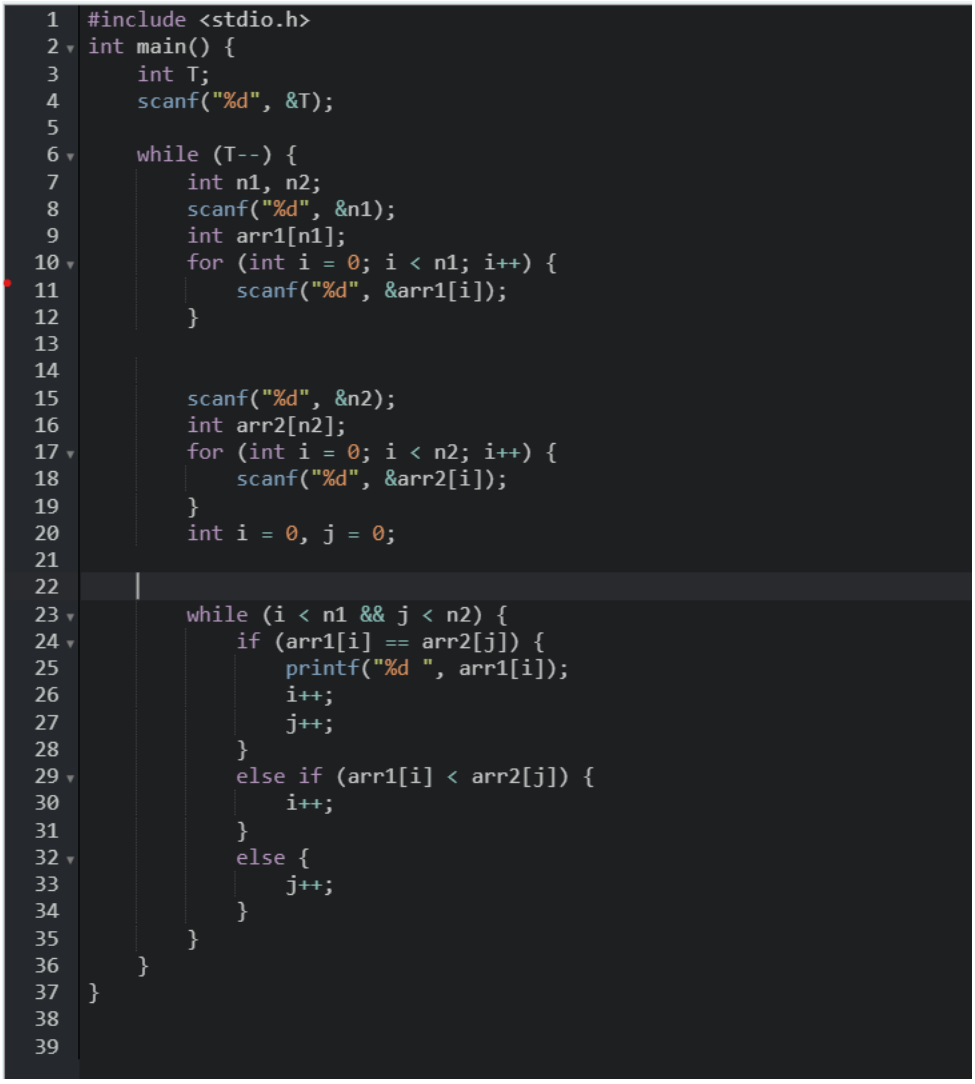
```
10 57
```

Input:

```
1
6 1 2 3 4 5 6
2 1 6
```

Output:

```
1 6
```



```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++;
            }
        }
    }
}
```

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:

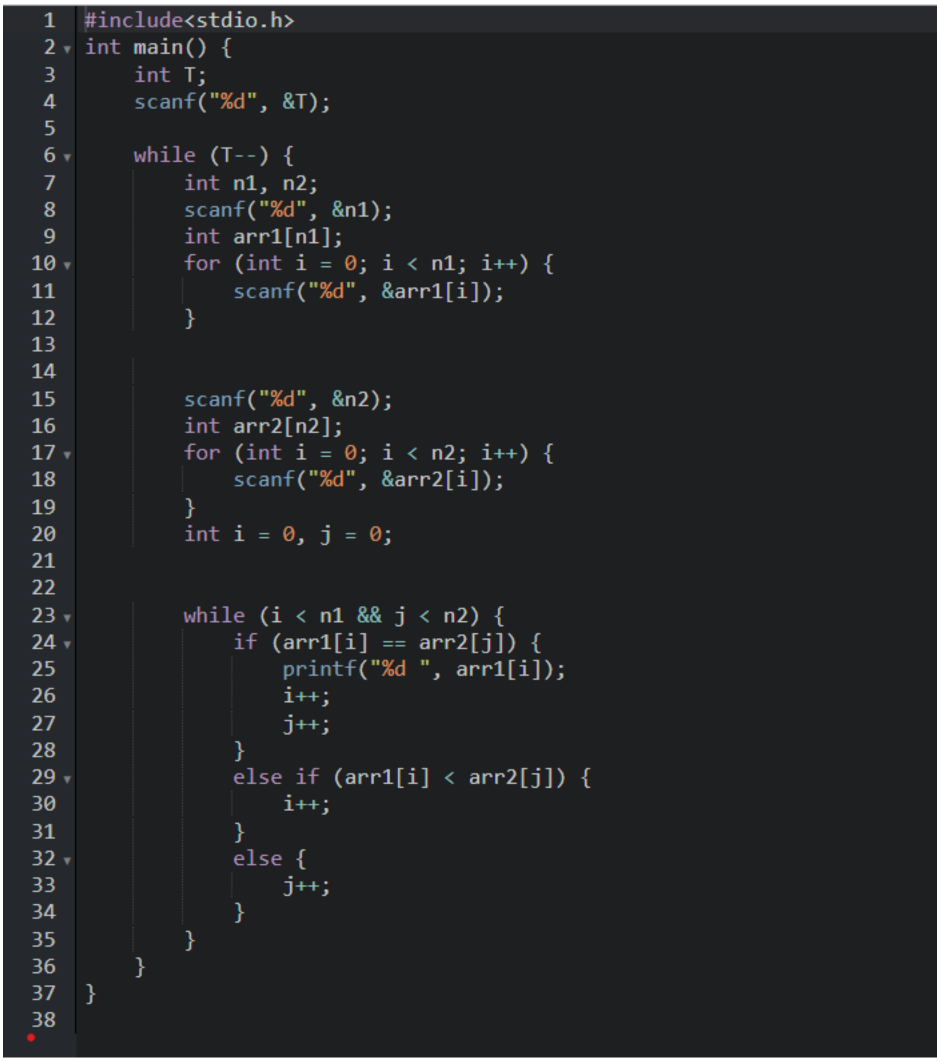
```
10 57
```

Input:

```
1
6 1 2 3 4 5 6
2 1 6
```

Output:

```
1 6
```



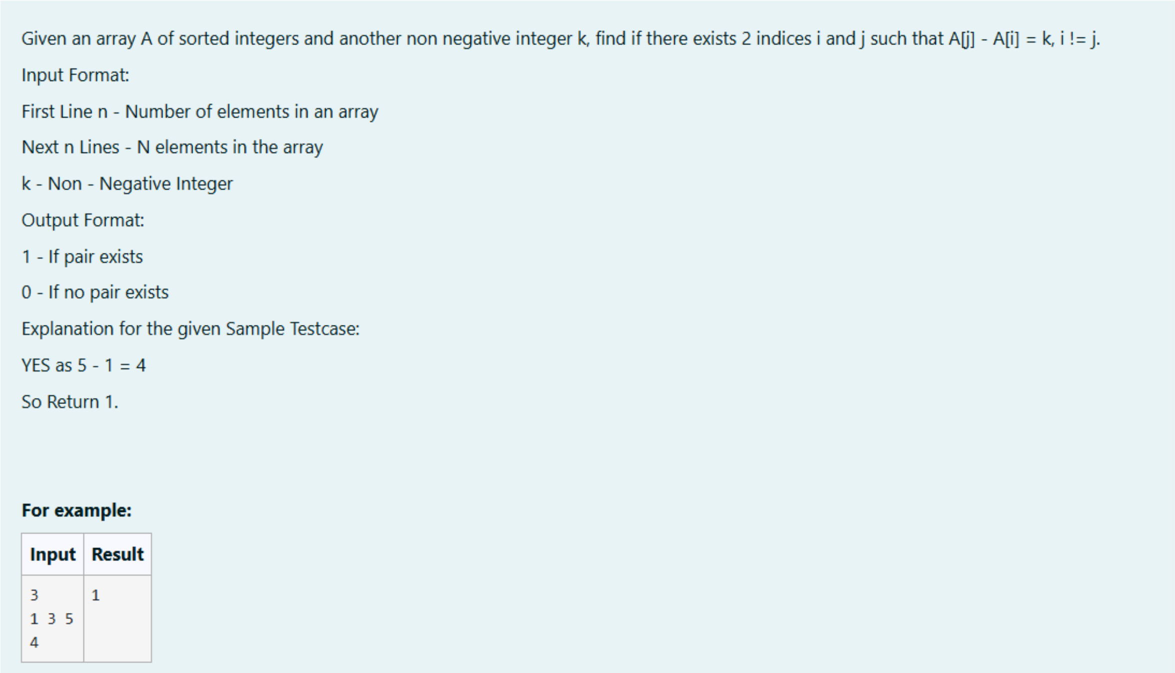
```c
#include<stdio.h>
int main() {
    int T;
    scanf("%d", &T);

    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++;
            }
        }
    }
}

```



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:

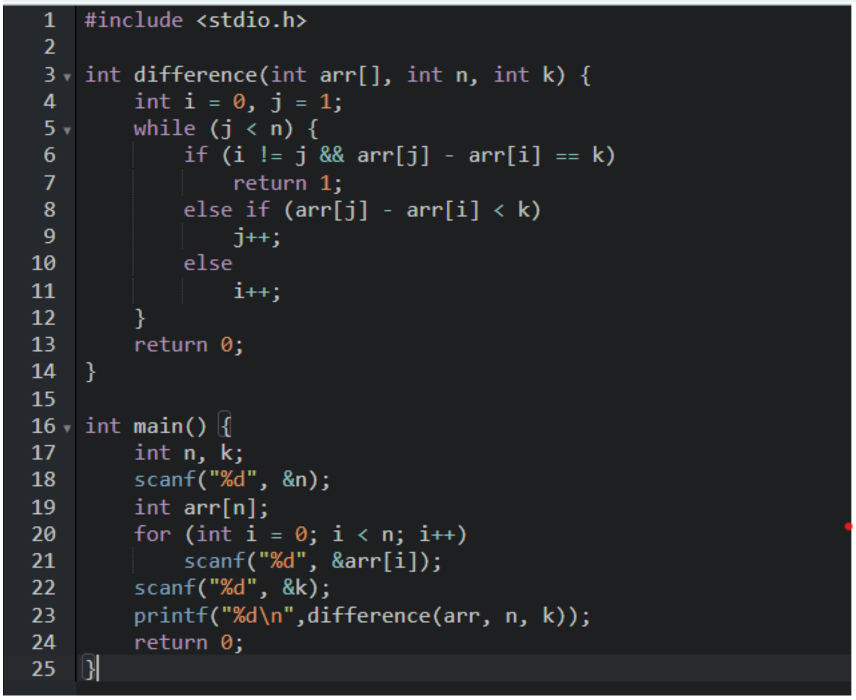
YES as 5 - 1 = 4

So Return 1.

For example:

Input	Result

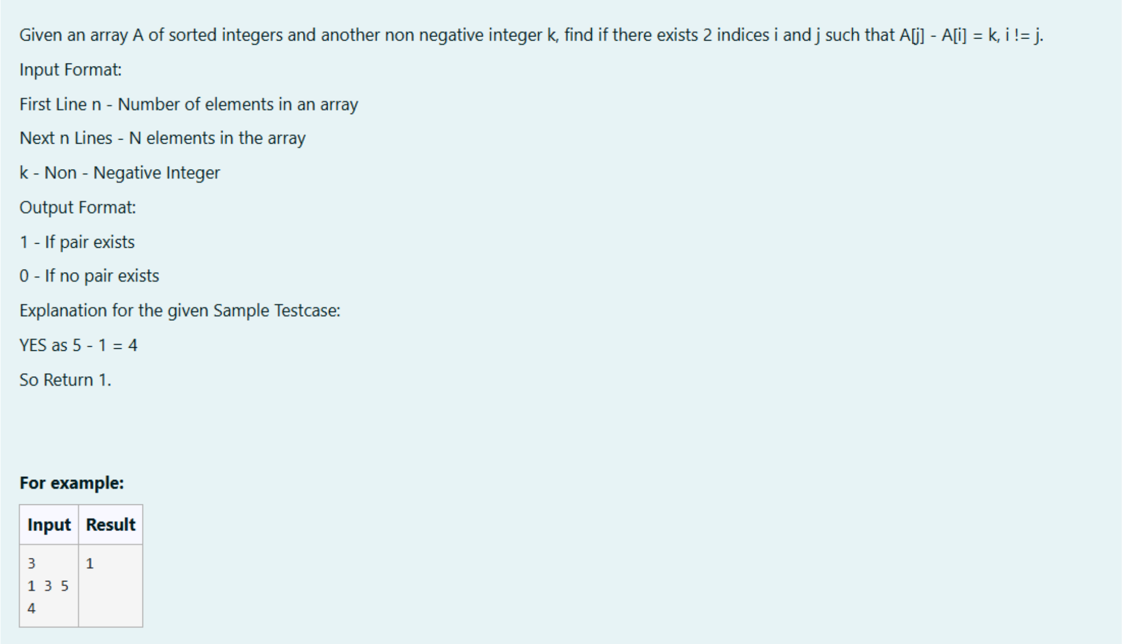
3
1 3 5
4
	1



```c
#include <stdio.h>

int difference(int arr[], int n, int k) {
    int i = 0, j = 1;
    while (j < n) {
        if (i != j && arr[j] - arr[i] == k)
            return 1;
        else if (arr[j] - arr[i] < k)
            j++;
        else
            i++;
    }
    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    scanf("%d", &k);
    printf("%d\n",difference(arr, n, k));
    return 0;
}
```



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:

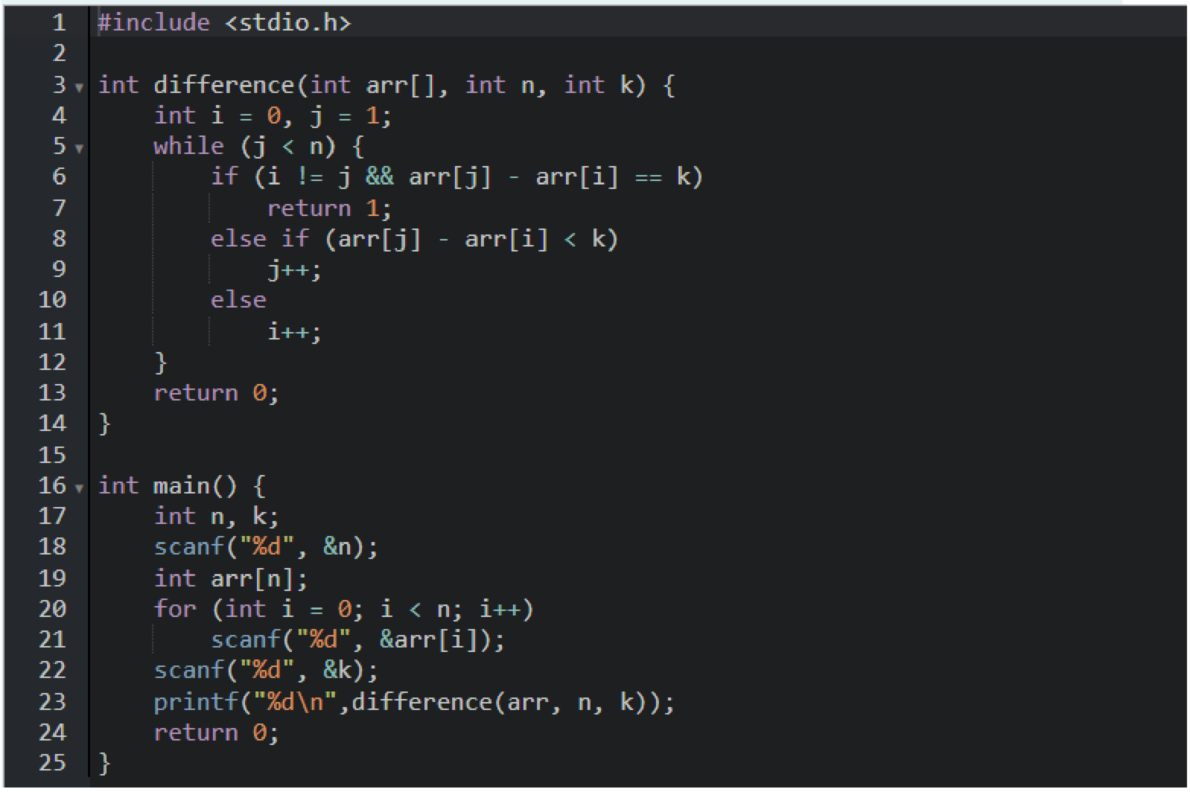
YES as 5 - 1 = 4

So Return 1.

For example:

Input	Result

3
1 3 5
4
	1



```c
#include <stdio.h>

int difference(int arr[], int n, int k) {
    int i = 0, j = 1;
    while (j < n) {
        if (i != j && arr[j] - arr[i] == k)
            return 1;
        else if (arr[j] - arr[i] < k)
            j++;
        else
            i++;
    }
    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    scanf("%d", &k);
    printf("%d\n",difference(arr, n, k));
    return 0;
}
```