COMPETITIVE PROGRAMMING

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

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>
int main(){
    int t;scanf("%d",&t);
    while(t!=0){t--;
    int a,b;
    scanf("%d",&a);
    int arr[a];
    for(int i=0;i<a;i++)scanf("%d",&arr[i]);
    scanf("%d",&b);
    int brr[b];
    for(int i=0;i<b;i++)scanf("%d",&brr[i]);

    for(int i=0;i<a;i++){
        for(int j=0;j<b;j++){
            if(arr[i]==brr[j]){
                printf("%d ",arr[i]);
                break;

            }

        }

    }
    }
}
```

	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! ✔

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

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>
int main(){
    int t;scanf("%d",&t);
    while(t!=0){
        t--;
        int a,b;scanf("%d",&a);
        int arr[a];
        for(int i=0;i<a;i++)scanf("%d",&arr[i]);
        scanf("%d",&b);
        int brr[b];
        for(int i=0;i<b;i++)scanf("%d",&brr[i]);
        int i=0,j=0;
        while(i<a&&j<b){
            if(arr[i]==brr[j]){
                printf("%d ",arr[i]);
                i++;j++;}
            else if(arr[i]<brr[j])i++;
            else 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! ✔

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

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 a;scanf("%d", &a);
    int arr[a];
    for(int i=0;i<a;i++)
    scanf("%d", &arr[i]);
    int c=0,k;
    scanf("%d", &k);
    for(int i=0;i<a;i++){
        for(int j=i+1;j<a;j++){
            if(arr[j]-arr[i]==k){c=1;break;

            }

        }

    }
    printf("%d\n",c);

}
```

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

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 a;scanf("%d", &a);
    int arr[a];
    for(int i=0;i<a;i++)scanf("%d", &arr[i]);
    int c=0,k;scanf("%d", &k);
    int i=0,j=1;
    while(j<a){
        int d=arr[j]-arr[i];
        if (d==k&&i!=j){
            c=1;break;

        }
        else if(d<k)j++;
        else i++;

    }
    if(c==1)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! ✔