

[Dashb...](#) / [My cou...](#) / [CS23331-DAA-202...](#) / [Competitive Progra...](#) / [3-Print Intersection of 2 sorted arrays- \$O\(m \cdot n\)\$  Time Complexity,  \$O\(1\)\$  Sp...](#)

<b>Started on</b>	Tuesday, 19 November 2024, 6:55 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 19 November 2024, 6:56 PM
<b>Time taken</b>	1 min 9 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>30.00</b> out of 30.00 ( <b>100%</b> )

## 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 %)

```
1 #include<stdio.h>
2 int main() {
3     int T;
4     scanf("%d", &T);
5
6     while (T--) {
7         int n1, n2;
8         scanf("%d", &n1);
9         int arr1[n1];
10        for (int i = 0; i < n1; i++) {
11            scanf("%d", &arr1[i]);
12        }
13
14
15        scanf("%d", &n2);
16        int arr2[n2];
17        for (int i = 0; i < n2; i++) {
18            scanf("%d", &arr2[i]);
19        }
20        int i = 0, j = 0;
21
22
```

```
23 while (i < n1 && j < n2) {
24     if (arr1[i] == arr2[j]) {
25         printf("%d ", arr1[i]);
26         i++;
27         j++;
28     }
29     else if (arr1[i] < arr2[j]) {
30         i++;
31     }
32     else {
33         j++;
34     }
35 }
36 }
37 }
38 }
```

	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 ▶