

# CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Competitive Programming / 3-Print Intersection of 2 sorted arrays-O(m\*n)Time Complexity,O(1) Space Complexity

Quiz navigation

1  
✓

[Finish review](#)

<b>Started on</b>	Friday, 25 October 2024, 1:36 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 25 October 2024, 2:23 PM
<b>Time taken</b>	46 mins 45 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	30.00 out of 30.00 (100%)

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:
  - Line 1 contains N1, followed by N1 integers of the first array
  - 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 ntest, n1, n2;
4     scanf("%d", &ntest);
5     scanf("%d", &n1);
6     int narray1[n1];
7     for(int i=0; i<n1; i++){
8         scanf("%d", &narray1[i]);
9         //printf("%d ", narray1[i]);
10    }
11
12    scanf("%d", &n2);
13    int narray2[n2];
14    for(int i=0; i<n2; i++){
15        scanf("%d", &narray2[i]);
16        //printf("%d ", narray2[i]);
17    }
18    //int check[1000] = {0};
19    for(int i=0; i<n1; i++){
20        for(int j=0; j<n2; j++){
21            if(narray1[i] == narray2[j]){
22                printf("%d ", narray1[i]);
23                break;
24            }
25        }
26    }
27
28 }
```

	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.

[Finish review](#)

← 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 →