

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

Started on	Sunday, 10 November 2024, 7:27 PM
State	Finished
Completed on	Sunday, 10 November 2024, 7:54 PM
Time taken	26 mins 25 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

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 #include <stdlib.h>
3 #define MAX_SIZE 100000
4
5 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
6     int *hashTable = (int *)calloc(MAX_SIZE, sizeof(int));
7
8     for (int i = 0; i < n1; i++) {
9         hashTable[arr1[i]] = 1;
10    }
11
12    int found = 0;
13    for (int j = 0; j < n2; j++) {
14        if (hashTable[arr2[j]] == 1) {
15            printf("%d ", arr2[j]);
```

```
16         found = 1;
17         hashTable[arr2[j]] = 0;
18     }
19 }
20
21 if (!found) {
22     printf("No intersection");
23 }
24 printf("\n");
25
26 free(hashTable);
27 }
28 int main() {
29     int T;
30     scanf("%d", &T);
31
32     while (T--) {
33         int n1;
34         scanf("%d", &n1);
35         int arr1[n1];
36         for (int i = 0; i < n1; i++) {
37             scanf("%d", &arr1[i]);
38         }
39
40         int n2;
41         scanf("%d", &n2);
42         int arr2[n2];
43         for (int i = 0; i < n2; i++) {
44             scanf("%d", &arr2[i]);
45         }
46
47         findIntersection(arr1, n1, arr2, n2);
48     }
49
50     return 0;
51 }
```

	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 ▶