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

Aim:

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 %)

Algorithm:

 Read the integer n (number of test cases).

 For each test case, read the size n1 and array arr1, then read the size n2 and array arr2.

 Compare each element of arr1 with each element of arr2.

 If they match, print the common element.

Code:

#include<stdio.h>

int main(){

int n;

scanf("%d",&n);

for(int i=0;i<n;i++){

int n1;

scanf("%d",&n1);

int arr1[n1];

for(int j=0;j<n1;j++){

scanf("%d ",&arr1[j]);

}

int n2;

scanf("%d",&n2);

int arr2[n2];

for(int j=0;j<n2;j++){

scanf("%d ",&arr2[j]);

}

for(int j=0;j<n1;j++){

for(int k=0;k<n2;k++){

if(arr1[j]==arr2[k]){

printf("%d ",arr1[j]);

}

}

}

}

}

Output:

|  | **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.

Result:

The expected output was obtained