4-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:

1. Read the number of test cases T.
2. For each test case, read the sizes n1 and n2, then input arrays arr1 and arr2.
3. Use two pointers i and j to traverse both arrays and print the common elements where arr1[i] == arr2[j].
4. Move the pointers i and j accordingly when a match is found or when an element in one array is smaller than the other.
5. Print the common elements for each test case on a new line.

Code:

#include <stdio.h>

int main() {

int T;

scanf("%d", &T);

while (T--) {

int n1, n2;

scanf("%d", &n1);

int arr1[n1];

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

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

}

scanf("%d", &n2);

int arr2[n2];

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

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

}

int i = 0, j = 0;

while (i < n1 && j < n2) {

if (arr1[i] < arr2[j]) {

i++;

}

else if (arr2[j] < arr1[i]) {

j++;

}

else {

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

i++;

j++;

}

}

printf("\n");

}

}

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