***Intersection of two arrays***

**Intersection**

1. Iterate in while loop till any of the one array is finished.
2. In each iteration we look for smaller of the two elements from both the array and increase its pointer because it will not be in other list, hence not part of intersection.
3. For intersection, if both the elements are equal we print it and increment both pointer only if it is not same as the last element printed in intersection.

C++

// C++ code to find intersection when

// elements may not be distinct

#include <bits/stdc++.h>

using namespace std;

// Function to find intersection

void intersection(int a[], int b[], int n, int m)

{

int i = 0, j = 0;

while (i < n && j < m) {

if (a[i] > b[j]) {

j++;

}

else if (b[j] > a[i]) {

i++;

}

else {

// when both are equal

cout << a[i] << " ";

i++;

j++;

}

}

}

// Driver Code

int main()

{

int a[] = { 1, 3, 2, 3, 3, 4, 5, 5, 6 };

int b[] = { 3, 3, 5 };

int n = sizeof(a) / sizeof(a[0]);

int m = sizeof(b) / sizeof(b[0]);

// sort

sort(a, a + n);

sort(b, b + m);

// Function call

intersection(a, b, n, m);

}

**Output**

3 3 5

**Time Complexity:**O(max(m\*log(m), n\*log(n)) + min(m, n) )  
**Auxiliary Space:** O(1)