***Union of Two Arrays***

Given two unsorted arrays that represent two sets (elements in every array are distinct), find the union and intersection of two arrays.

**Example:**

arr1[] = {7, 1, 5, 2, 3, 6}

arr2[] = {3, 8, 6, 20, 7}

Then your program should print Union as {1, 2, 3, 5, 6, 7, 8, 20}.

**Method: (Using map data structure)**

From the knowledge of data structures, we know that map stores distinct keys only. So if we insert any key appearing more than one time it gets stored only once. The idea is to insert both the arrays in one common map which would then store the distinct elements of both arrays (union of both the array).

Below is the implementation of the above method:

C++

// C++ program for the above approach

#include <bits/stdc++.h>

using namespace std;

void printUnion(int\* a, int n, int\* b, int m)

{

// Defining map container mp

map<int, int> mp;

// Inserting array elements in mp

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

mp.insert({ a[i], i });

for (int i = 0; i < m; i++)

mp.insert({ b[i], i });

cout << "The union set of both arrays is :" << endl;

for (auto itr = mp.begin(); itr != mp.end(); itr++)

cout << itr->first

<< " "; // mp will contain only distinct

// elements from array a and b

}

// Driver Code

int main()

{

int a[7] = { 1, 2, 5, 6, 2, 3, 5 };

int b[9] = { 2, 4, 5, 6, 8, 9, 4, 6, 5 };

printUnion(a, 7, b, 9);

}

**Output**

The union set of both arrays is :

1 2 3 4 5 6 8 9

**Time Complexity:**O(m \* log(m) + n \* log(n)), for using map data structure.  
**Auxiliary Space:** O(m + n)