

## Union and Intersection of two sorted arrays

For example, if the input arrays are:

arr1[] = {1, 3, 4, 5, 7}

arr2[] = {2, 3, 5, 6}

Then your program should print Union as {1, 2, 3, 4, 5, 6, 7} and Intersection as {3, 5}.

### Algorithm Union(arr1[], arr2[]):

For union of two arrays, follow the following merge procedure.

- 1) Use two index variables i and j, initial values i = 0, j = 0
- 2) If arr1[i] is smaller than arr2[j] then print arr1[i] and increment i.
- 3) If arr1[i] is greater than arr2[j] then print arr2[j] and increment j.
- 4) If both are same then print any of them and increment both i and j.
- 5) Print remaining elements of the larger array.

```
#include<stdio.h>
```

```
/* Function prints union of arr1[] and arr2[]
   m is the number of elements in arr1[]
   n is the number of elements in arr2[] */
int printUnion(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while(i < m && j < n)
    {
        if(arr1[i] < arr2[j])
            printf(" %d ", arr1[i++]);
        else if(arr2[j] < arr1[i])
            printf(" %d ", arr2[j++]);
        else
        {
            printf(" %d ", arr2[j++]);
        }
    }
}
```

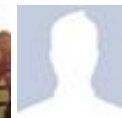
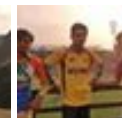
Google™ Custom Search



GeeksforGeeks



53,522 people like [GeeksforGeeks](#).



Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

Geometric Algorithms

```

        i++;
    }
}

/* Print remaining elements of the larger array */
while(i < m)
    printf(" %d ", arr1[i++]);
while(j < n)
    printf(" %d ", arr2[j++]);
}

/* Driver program to test above function */
int main()
{
    int arr1[] = {1, 2, 4, 5, 6};
    int arr2[] = {2, 3, 5, 7};
    int m = sizeof(arr1)/sizeof(arr1[0]);
    int n = sizeof(arr2)/sizeof(arr2[0]);
    printUnion(arr1, arr2, m, n);
    getchar();
    return 0;
}

```

Time Complexity:  $O(m+n)$

### Algorithm Intersection(arr1[], arr2[]):

For Intersection of two arrays, print the element only if the element is present in both arrays.

- 1) Use two index variables i and j, initial values  $i = 0$ ,  $j = 0$
- 2) If  $arr1[i]$  is smaller than  $arr2[j]$  then increment i.
- 3) If  $arr1[i]$  is greater than  $arr2[j]$  then increment j.
- 4) If both are same then print any of them and increment both i and j.

```

#include<stdio.h>

/* Function prints Intersection of arr1[] and arr2[]
   m is the number of elements in arr1[]
   n is the number of elements in arr2[] */
int printIntersection(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while(i < m && j < n)
    {
        if(arr1[i] < arr2[j])
            i++;
        else if(arr2[j] < arr1[i])
            j++;
        else
            printf("%d ", arr1[i]);
            i++;
            j++;
    }
}

```



## Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

        j++;
    else /* if arr1[i] == arr2[j] */
    {
        printf(" %d ", arr2[j++]);
        i++;
    }
}

/* Driver program to test above function */
int main()
{
    int arr1[] = {1, 2, 4, 5, 6};
    int arr2[] = {2, 3, 5, 7};
    int m = sizeof(arr1)/sizeof(arr1[0]);
    int n = sizeof(arr2)/sizeof(arr2[0]);
    printIntersection(arr1, arr2, m, n);
    getchar();
    return 0;
}

```

Time Complexity:  $O(m+n)$

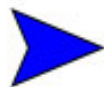
Please write comments if you find any bug in above codes/algorithms, or find other ways to solve the same problem.

## 2 Foods To NEVER Eat:

~~~~~



Trim away a bit  
of stubborn flab  
like belly fat  
every week by  
never eating  
these 2 foods...



Never Eat



## Related Topics:

- Remove minimum elements from either side such that  $2 \times \min$  becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3



3



Tweet

0



0

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

## Recent Comments

**Aman** Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 24 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 28 minutes ago

**Sanjay Agarwal** bool

tree::Root\_to\_leaf\_path\_given\_sum(tree...

Root to leaf path sum equal to a given number · 53 minutes ago

**GOPI GOPINATH** @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 54 minutes ago

**newCoder3006** If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

**newCoder3006** Code without using while loop. We can do it...


Find subarray with given sum · 1 hour ago

AdChoices 

▶ [JavaScript Array](#)

▶ [C++ Vector](#)


▶ [C++ Code](#)

AdChoices 

▶ [Arrays in PHP](#)

▶ [Java Array](#)

▶ [C++ Array](#)

AdChoices 

▶ [Programming C++](#)

▶ [C Union](#)

▶ [Array of Arrays](#)

@geeksforgeeks, **Some rights reserved**

**Contact Us!**

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team