

## DSA Lab 5 Set 2 | SortedUnion

Given two **sorted** linked lists of integers, Write an efficient algorithm to take the union of the two lists such that the resultant list is sorted.

Your algorithm should minimise the number of comparison operations involved. All the elements in a list are distinct and no element appears in both the lists.

### Input

The first line contains two space separated integers  $X$  and  $Y$  denoting the size of two linked lists.

The second line contains  $X$  space separated integers in sorted order, denoting element values in the first linked list.

The third line contains  $Y$  space separated integers in sorted order, denoting element values in the second linked list.

### Constraints:

Basic:

$$1 \leq X \leq 50$$

$$1 \leq Y \leq 50$$

$$1 \leq \text{elements} \leq 100000$$

Advanced:

$$1 \leq X \leq 5000$$

$$1 \leq Y \leq 5000$$

$$1 \leq \text{elements} \leq 100000$$

### Output

The first line contains  $X + Y$  space separated integers of sorted union performed on the list.

The second line contains 1 integer representing the number of comparisons involved (refer to sample for clarity).

### Sample Test Case

#### Input:

4 6

2 5 8 9

1 4 6 7 10 14

#### Output:

1 2 4 5 6 7 8 9 10 14

8

### Explanation:

The lists are 2 5 8 9 and 1 4 6 7 10 14.

1 and 2 are compared of which 1 is picked to be added to new list.

Then 2 and 4 are compared of which 2 is picked and added after 1.

Then 5 and 4 are compared, choosing 4. Similarly,

Compare 5,6 choosing 5

Compare 8,6 choosing 6

Compare 8,7 choosing 7

Compare 8,10 choosing 8

Compare 9,10 choosing 9

Then as first linked list is exhausted, remaining elements of second list (values 10 and 14) are added as is to the new list, requiring no comparisons. Hence, 8 comparisons in all.