- **EASY**
- **Max Score: 30 Points**

SORT THE HALF SORTED

Given an integer array of which both the first half and second half are sorted. The task is to merge two sorted halves of the array into a single sorted array.

Note: The two halves can be of arbitrary sizes.

Input Format

line 1: contains an integer n denoting the size of the array.

line 2: contains n-spaced integers denoting elements of the array.

Output Format

Print an array consisting of n elements denoting the sorted array.

Example 1

Input

```
6
2 3 8 −1 7 10
```

Output

```
−1 2 3 7 8 10
```

Explanation

The first half sorted array is 2 3 8 and the second half is -1 7 10 and if we merge them and sort them we will get -1 2 3 7 8 10

Example 2

Input

```
5
3 4 0 1 2
```

Output

```
0 1 2 3 4
```

Explanation

The first half sorted array is 3 4 and the second half is 0 1 2 and if we merge them and sort them we will get 0 1 2 3 4

Constraints

1<=n<=10^5

-10^6<=arr[i]<=10^6

Expected Time Complexity: O(N)

Expected Space Complexity: O(N)

Topic Tags

- **Arrays**

My code

// in java

```java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
    public static void main (String[] args) throws
java.lang.Exception
    {
        //your code here
    Scanner s=new Scanner(System.in);
    int n=s.nextInt();
    // int arrb[]=new int[n];
     int arr[]=new int[n];
    for(int i=0;i<n;i++)
     arr[i]=s.nextInt();
     int l=0;

        int r=n;
        for(int i=0;i<n-1;i++)
            if(arr[i]>arr[i+1])
        {
            r=i+1;
            break;
        }
        int p=r;
        while(l<p && r<n)
            {
                if(arr[l]<arr[r])
                {
```

```java
                    System.out.print(arr[l]+" ");
                    l++;
                }
                    else
                    {
                    System.out.print(arr[r]+" ");
                        r++;
                    }
            }

        while(l<p)
            {
                    System.out.print(arr[l]+" ");
                    l++;
            }

        while( r<n)
            {
                System.out.print(arr[r]+" ");
                    r++;

            }

        }
}
```