88. Merge Sorted Array

Solved

Easy | Topics | Companies | Hint

You are given two integer arrays `nums1` and `nums2`, sorted in **non-decreasing order**, and two integers `m` and `n`, representing the number of elements in `nums1` and `nums2` respectively.

Merge `nums1` and `nums2` into a single array sorted in **non-decreasing order**.

The final sorted array should not be returned by the function, but instead be *stored inside the array* `nums1`. To accommodate this, `nums1` has a length of `m + n`, where the first `m` elements denote the elements that should be merged, and the last `n` elements are set to `0` and should be ignored. `nums2` has a length of `n`.

Example 1:

```
Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.
```

Example 2:

```
Input: nums1 = [1], m = 1, nums2 = [], n = 0
Output: [1]
Explanation: The arrays we are merging are [1] and [].
The result of the merge is [1].
```

Example 3:

```
Input: nums1 = [0], m = 0, nums2 = [1], n = 1
Output: [1]
Explanation: The arrays we are merging are [] and [1].
The result of the merge is [1].
Note that because m = 0, there are no elements in nums1. The 0 is only there to ensure the
merge result can fit in nums1.
```

Java | Auto

```java
class Solution {
    public void merge(int[] nums1, int m, int[] nums2, int n) {
        int l=0;
        for(int i=0;i<nums1.length;i++){
            if(nums1[i]==0 && l<n){
                nums1[i]=nums2[l];
                l++;
            }
        }
        Arrays.sort(nums1);
    }
}
```

Last Executed Input

nums1 =
[-1,0,0,3,3,3,0,0,0]

m =
6

nums2 =
[1,2,2]

n =
3

Sophie is a student who enjoys playing video games in her free time. One of her favorite games involves an **array** of **integers**. Sophie noticed that the array was unsorted and she wanted to **sort** it in a unique way.

She thought of an interesting challenge - to sort the array acccording to the **cubes** of the elements.

Help Sophie and write a program that sort the array according to teh cubes of the elements.

Input Format

First line take an integer input from user as **N** , where **N** is the size of array.

Second line takes **N** elements as integers input in array.

Constraints

```
1 <= N <= 100
-10^2 <= arr[i] <= 10^2
```

Output Format

Return the sorted array according to their cubes

Sample Input 0

```
5
4 -1 0 -5 6
```

Sample Output 0

```
-5 -1 0 4 6
```

Explanation 0

Print the sorted array

Submitted Code

Language: Java 15

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

public class Solution {

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       int arr[] = new int[n];
       for(int i=0;i<n;i++)arr[i]=sc.nextInt();

       //Bubble sort

       for(int i=0;i<n-1;i++){
           for(int j=0;j<n-1-i;j++){
               if((arr[j]*arr[j]*arr[j])> (arr[j+1]*arr[j+1]*arr[j+1]) ){
                   int temp = arr[j];
                   arr[j]=arr[j+1];
                   arr[j+1]= temp;
               }
           }
       }
       for(int i=0;i<n;i++){
           System.out.print(arr[i]+" ");
       }
   }
}
```

Submitted Code

Language: Java 15

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

public class Solution {

    public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Integer arr[] = new Integer[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();
       Arrays.sort(arr, (a,b)->{
           return (a*a*a)- (b*b*b);
       });
        for(int i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
    }
}
```

Sophie and her friend Luke decided to play a game with an array of integers. They wrote a program that **sort** the array in a unique way. The program sorted an array of integers in such a way that half of the integers were **odd** and the other half were **even**.

Moreover, whenever an **odd** number occurred in the array, it was placed at an **odd index**, and whenever an **even** number occurred in the array, it was placed at an **even index**. Both **odd** and **even** numbers were sorted in **non-decreasing** order.

Return any answer array that satisfies this condition.

Input Format

First line contains an integer **N** representing the size of the array.

Next **N** line contains **N** integers representing elements of the array.

Constraints

```
2 <= nums.length <= 2 * 104

nums.length is even.

Half of the integers in nums are even.

0 <= nums[i] <= 1000
```

Output Format

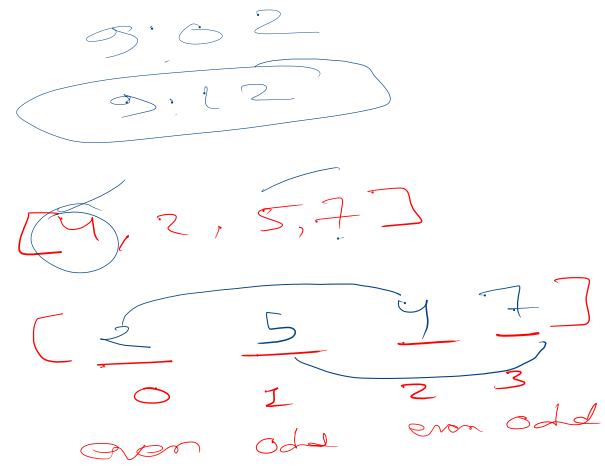
Return the answer where each element is seperated by a single space.

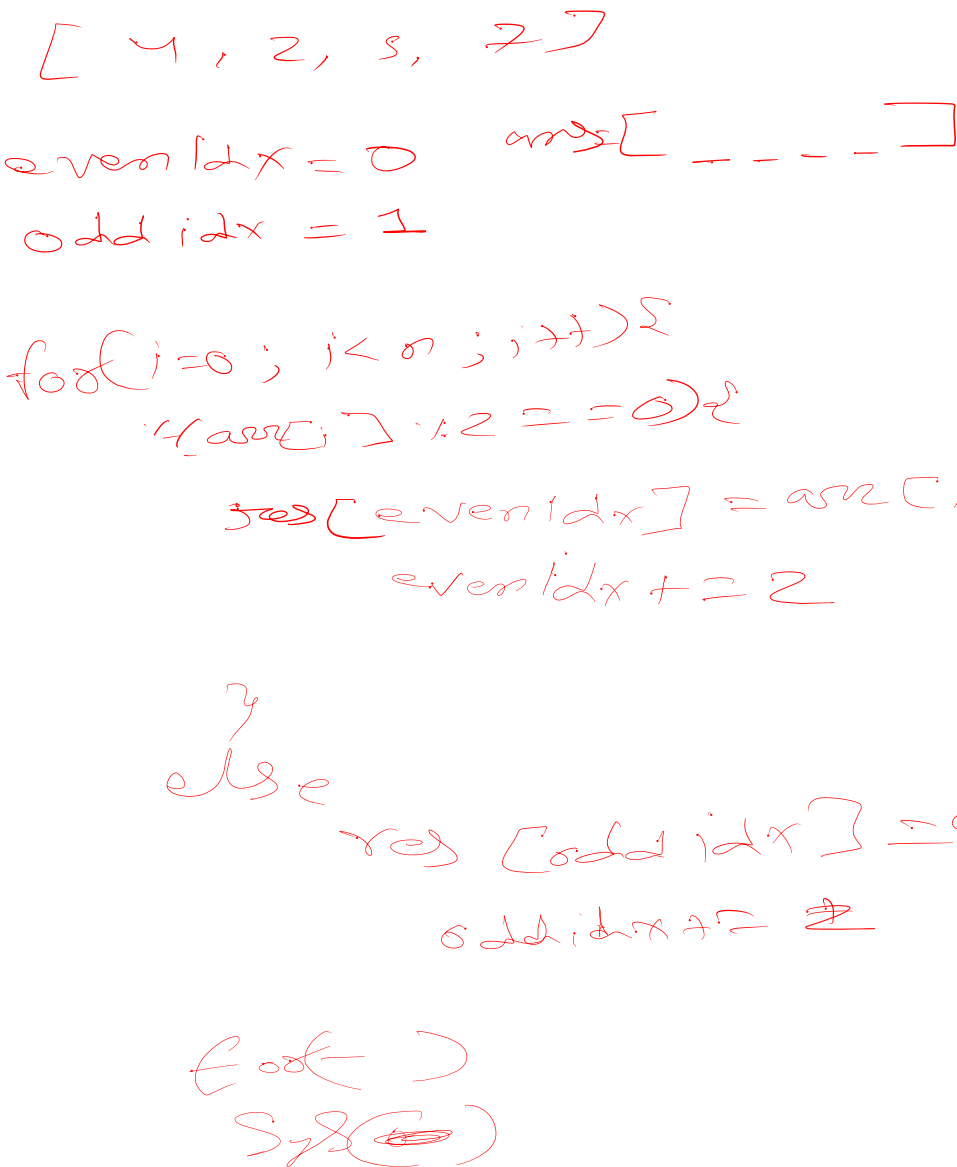
Sample Input 0

```
4
4
2
5
7
```

Sample Output 0

```
2 5 4 7
```





Submitted Code

Language: Java 15

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

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();
        Arrays.sort(arr);
        int res[] = new int[n];
        int evenidx=0,oddidx=1;

        for(int i=0;i<n;i++){
            if(arr[i]%2==0){
                res[evenidx]= arr[i];
                evenidx+=2;
            }
            else {
                res[oddidx]= arr[i];
                oddidx+=2;
            }
        }
         for(int i=0;i<n;i++)System.out.print(res[i]+" ");
    }
}
```

Meet Sarah, an enthusiastic programmer who loves to solve challenging problems. She was recently given an array of non-negative integers and was asked to arrange its elements in such a way that they form the **Smallest** possible number.

Solve the problem by comparing the values of the elements in a way that produced the **Smallest** possible number.

Input Format

First line take an integer input from user as **N** , where **N** is the size of array.

Second line takes **N** elements as non-negative integers input in array.

Constraints

```
1<=N<=100

1<=arr[i]<=10^4
```

Output Format

Return the smallest number.

Sample Input 0

```
6
5 6 2 9 21 1
```

Sample Output 0

```
1212569
```

Explanation 0

Print the smallest number.

Submitted Code

Language: Java 15

```
1  import java.io.*;
2  import java.util.*;
3
4  public class Solution {
5
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8          int n = sc.nextInt();
9          int arr[]= new int[n];
10         for(int i=0;i<n;i++)arr[i]=sc.nextInt();
11
12          String arr1[] = new String[n];
13         for(int i=0;i<n;i++)arr1[i]=Integer.toString(arr[i]);
14         Arrays.sort(arr1,(a,b)->{
15             String val1 = a+b;
16             String val2 = b+a;
17             return val1.compareTo(val2);
18         });
19
20
21         String ans = "";
22         for(int i=0;i<arr1.length;i++)ans+=arr1[i];
23
24
25         System.out.print(ans);
26
27     }
28 }
```