Counting Sort 2

Often, when a list is sorted, the elements being sorted are just keys to other values. For example, if you are sorting files by their size, the sizes need to stay connected to their respective files. You cannot just take the size numbers and output them in order, you need to output all the required file information.

However, if you are not concerned about any other information, then you can simply sort those numbers alone. This makes counting sort very simple. If you already counted the values in the list, you don't need to access the original list again. This challenge involves a simple counting sort where the elements being sorted are all that matter.

Challenge
Given an unsorted list of integers, output the integers in order.

Hint: You can use your previous code that counted the items to print out the actual values in-order.

Input Format
There will be two lines of input:

- n - the size of the list
- ar - *n* space separated numbers that belong to the list

Output Format
Output all the numbers of the list in-order.

Constraints
1 <= n <= 1000000
0 <= x < 100 , x ∈ ar

Sample Input

```
100
63 25 73 1 98 73 56 84 86 57 16 83 8 25 81 56 9 53 98 67 99 12 83 89 80 91 39 86 76 85 74 39 25 90 59 10 94 32 44 3 89 30 27 79 46 96 27 32 18 21 92 6 9 81 40 40 34 68 78 24 87 42 69 23 41 78 22 6 90 99 89 50 30 20 1 43 3 70 95 33 46 44 9 69 48 33 60 65 16 82 67 61 32 21 79 75 75 13 87 70 33
```

Sample Output

```
1 1 3 3 6 8 9 9 10 12 13 16 16 18 20 21 21 22 23 24 25 25 25 27 27 30 30 32 32 32 33 33 33 34 39 39 40 40 41 42 43 44 44 46 46 48 50 53 56 56 57 59 60 61 63 65 67 67 68 69 69 69 70 70 73 73 74 75 75 76 78 78 79 79 80 81 81 82 83 83 84 85 86 86 87 87 89 89 89 90 90 91 92 94 95 96 98 98 99 99
```

Explanation

In the output you can see the numbers sorted in ascending order. You can also see that numbers appearing multiple times are printed accordingly.