Question - 1
Can You Sort?

An array of integers, *arr*, of size N is defined as $\{a_0, a_1, ..., a_{N-1}\}$.

Complete the *customSort* function declared in your editor. It must take *arr* as a parameter, sort its elements in order of ascending frequency, and then print each element of the sorted array as a new line of output. If *2* or more elements have the same frequency, this subset of elements should be sorted in non-decreasing order.

Input Format
The locked stub code in the editor handles reading input from stdin, assembling it into an array of integers (*arr*), and calling the *sort* function. The first line of input contains an integer, N (the number of elements). Each line *i* of the N subsequent lines describes array element *arr[i]*.

Constraints
- $1 \le N \le 2 \times 10^6$
- $1 \le a_i \le 10^6$

Output Format
Your *customSort* function should print the sorted (in order of *non-decreasing* frequency) elements of array *arr*. If *2* or more elements have the same frequency, this subset of elements should be sorted in non-decreasing order. Each element must be printed on a new line.
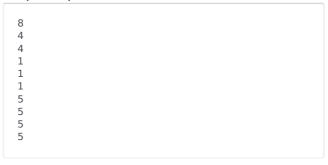
Sample Input 1

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

Sample Output 1

```
1
3
4
2
2
```

Sample Input 2

```
10
8
5
5
5
5
```

```
1
4
4
```

Sample Output 2

```
8
4
4
1
1
1
5
5
5
5
```

Explanation

Sample Case 1

N = 5, arr = {3, 1, 2, 2, 4}

First, we separate our numbers by *frequency.*

The subset of numbers having frequency *1* is *{3, 1, 4}*.

The subset of numbers having frequency *2* is *{2, 2}*.

Our partially sorted data (with respect to and in ascending order of frequency) can be expressed as *{{3, 1, 4}, {2, 2}}*.

Then we sort each subset of elements having the same frequency in non-decreasing order, resulting in *{{1, 3, 4}, {2, 2}}*.

Sample Case 2

N = 10, arr = {8, 5, 5, 5, 5, 1, 1, 1, 4, 4}

First, we separate our numbers by *frequency.*

The subset of numbers having frequency *1* is *{8}*.

The subset of numbers having frequency *2* is *{4}*.

The subset of numbers having frequency *3* is *{1}*.

The subset of numbers having frequency *4* is *{5}*.

Our partially sorted data (with respect to and in ascending order of frequency) can be expressed as *{{8},{4, 4},{1, 1, 1},{5, 5, 5, 5}}*.