Big Sorting

by _mfv_

Consider an array of numeric strings, $unsorted$, where each string is a positive number with anywhere from 1 to 10^6 digits. Sort the array's elements in *non-decreasing* (i.e., ascending) order of their real-world integer values and print each element of the sorted array on a new line.

Input Format

The first line contains an integer, n, denoting the number of strings in $unsorted$.
Each of the n subsequent lines contains a string of integers describing an element of the array.

Constraints

- $1 \leq n \leq 2 \times 10^5$
- Each string is guaranteed to represent a positive integer without leading zeros.
- The total number of digits across all strings in $unsorted$ is between 1 and 10^6 (inclusive).

Output Format

Print each element of the sorted array on a new line.

Sample Input 0

```
6
31415926535897932384626433832795
1
3
10
3
5
```

Sample Output 0

```
1
3
3
5
10
31415926535897932384626433832795
```

Explanation 0

The initial array of strings is $unsorted = [31415926535897932384626433832795, 1, 3, 10, 3, 5]$. When we order each string by the real-world integer value it represents, we get:

$$1 \leq 3 \leq 3 \leq 5 \leq 10 \leq 31415926535897932384626433832795$$

We then print each value on a new line, from smallest to largest.

Submissions:23775

Max Score:20
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Java 7

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11         int n = in.nextInt();
12         String[] unsorted = new String[n];
13         for(int unsorted_i=0; unsorted_i < n; unsorted_i++){
14             unsorted[unsorted_i] = in.next();
15         }
16         // your code goes here
17     }
18 }
19
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code