Big Sorting

by _mfv_

Problem Submissions Leaderboard Discussions Editorial

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.

```cpp
#include <bits/stdc++.h>

using namespace std;

bool sort_string(string a, string b) {
    if (a.length() < b.length()) {
       return true;
    } else if (a.length() > b.length()) {
       return false;
    }
    for (int i = 0; i < a.length(); i++) {
        if (a[i] < b[i]) {
            return true;
        } else if (a[i] > b[i]) {
            return false;
        }
    }
    return false;
}

int main(){
    int n;
    cin >> n;
    vector<string> unsorted(n);
    for(int unsorted_i = 0; unsorted_i < n; unsorted_i++){
       cin >> unsorted[unsorted_i];
    }
    sort(unsorted.begin(), unsorted.end(), sort_string);
    for(int unsorted_i = 0; unsorted_i < n; unsorted_i++){
       cout << unsorted[unsorted_i] << "\n";
    }
    // your code goes here
    return 0;
}

```

Line: 1 Col: 1

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