Big Sorting ☆

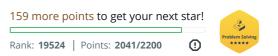**You have successfully solved Big Sorting**   [Share]   Tweet

You are now 159 points away from the 6th star for your problem solving badge.

Try the next challenge | **Try a Random Challenge**

Problem   Submissions   Leaderboard   Editorial

RATE THIS CHALLENGE

★ ★ ★ ★ ★

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

Function Description

Complete the bigSorting function in the editor below. It should return the sorted string array.

bigSorting has the following parameter(s):

- unsorted: an unsorted array of integers as strings

Input Format

The first line contains an integer, n, denoting the number of strings in $unsorted$.
Each of the n subsequent lines contains an integer string $unsorted[i]$.

Constraints

- $1 \le n \le 2 \times 10^5$
- 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.

Sample Input 1

```
8
1
2
100
12303479849857341718340192371
3084193741082937
3084193741082938
111
200
```

Sample Output 1

```
1
2
100
111
200
3084193741082937
3084193741082938
12303479849857341718340192371
```

C++14

```cpp
1    #include <algorithm>
2    #include <functional>
3    #include <iostream>
4    #include <string>
5    #include <vector>
6
7    int main() {
8      int n{}, Len{};
9      std::string temp{};
10     std::cin >> n;
11     std::vector<std::string> vec(n);
12     for (std::string &element : vec)
13       std::cin >> element;
14
15     static const auto compare = [](const std::string &now,
```

```
16                                  const std::string &next) noexcept->bool {
17        return now.size() == next.size() ? now < next : now.size() < next.size();
18      };
19      std::sort(vec.begin(), vec.end(), std::cref(compare));
20      for (const std::string &element : vec)
21        std::cout << element << '\n';
22      return 0;
23    }
24
```

Line: 24 Col: 1

↑ **Upload Code as File**   ☐ **Test against custom input**   **Run Code**   **Submit Code**

Congratulations
You solved this challenge. Would you like to challenge your friends? [f] [t] [in]

Next Challenge

Test case 0 ⊘

Test case 1 ⊘

Test case 2 ⊘

Test case 3 ⊘

Test case 4 ⊘

Test case 5 ⊘

Test case 6 ⊘

Compiler Message

Success

Input (stdin)   Download

    6
    31415926535897932384626433832795
    1
    3
    10
    3
    5

Expected Output   Download

    1
    3
    3
    5

Request a Feature

Request a Feature