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:27047

Max Score:20
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++14

```cpp
#include <bits/stdc++.h>
using namespace std;

bool Comp(string now, string next)
{
   int a=now.length();
   int b=next.length();
   if(a==b) return now<next;
   return a<b;
}

int main()
{
    int n=0, Len=0;
    string temp;
    cin >> n;
    vector<string> Vec(n);
    for(int i = 0; i < n; i++)
       cin >> Vec[i];
   sort(Vec.begin(), Vec.end(), Comp);
    /*Len=Vec.size();
    for(int Times=0; Times<Len-1; ++Times)
       for(int i = 0; i <=Len-1-Times; i++)
          if(Vec[i]>Vec[i+1])
          {
               string temp=Vec[i];
               Vec[i]=Vec[i+1];
               Vec[i+1]=temp;
               cout<<Vec[i]<<"  "<<Vec[i+1]<<endl;
          }*/
   for(vector<string>::iterator itr=Vec.begin();
                                itr!=Vec.end(); ++itr)
        cout<<*itr<<endl;
    return 0;
}

```

Line: 36 Col: 1

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

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run you code against all the test cases.

Input (stdin)

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

Your Output (stdout)

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

Expected Output

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