CSE100 Algorithm Design and Analysis — Lab 06, Spring 2024
Deadline: See Catcourses

Radixsort

Description In this assignment you will implement RadixSort. See the textbook for the pseudocode. You must use *counting sort* as a stable sort algorithm.

Input structure You are going to apply RadixSort to sort vectors. The input starts with an integer number which indicates the number of vectors to be sorted. Then vectors follow, one vector per line. Each vector consists of 10 numbers where each number has a value in $\{0, 1, 2, 3\}$. Entries of a vector are separated by a space.

Output structure Output the sorted sequence of vectors, one per line. Vector i must appear before vector j in your output if and only if for some $d \in \{1, 2, ..., 10\}$, vector i is smaller than or equal to vector j on the dth entry, and the two vectors are equal for any of the first $d - 1$ entries. So, vectors should be ordered in a lexicographically increasing order.

Examples of input and output

Input

```
5
3 3 3 3 3 2 2 2 2 2
2 3 2 2 2 2 2 2 2 2
1 3 0 0 2 1 0 0 0 0
1 3 0 0 2 2 0 0 0 0
2 3 2 1 2 2 2 2 2 2
```

Output

```
1;3;0;0;2;1;0;0;0;0;
1;3;0;0;2;2;0;0;0;0;
2;3;2;1;2;2;2;2;2;2;
2;3;2;2;2;2;2;2;2;2;
3;3;3;3;3;2;2;2;2;2;
```

More precisely, this output example has 6 lines since a “cout << endl;” call was made at the end of each of the first 5 lines; those are the only white characters.

See the lab guidelines for submission/grading, etc., which can be found in Files/Labs.

Note: if it takes more than 10 seconds to run your code for all the 10 examples, you will get 0 points. It’s because it would mean that you implemented something wrong – in most cases, it shouldn’t take more than 2 seconds.