Radixsort

Description In this assignment you are requested to implement RadixSort .

For the test of your codes, copy "GradeMe03" into your working directory. You also need to copy the directory "testfiles" into your working directory along with all files under the "testfiles" directory: t1, t2, ..., t5 and o1, o2, ..., o5. For each testfile tx, the right answer is in file ox. All files are archived in "Lab03.tar" which can be found in the files tab in CatCourse. After compile, your execution file's name must be "RadixSort.exe". Perhaps you may need to change the permission of "GradeMe03". To do this, you can simply type

chmod 500 GradeMe03

Now if you run "./GradeMe03" and your code is correct, you will see the following messages.

Correct for 1 th example.
Correct for 2 th example.
...

In this assignment, there are 5 test files.
You can find the execution log in the "result" file.
You are required to compile your codes in front of the TA, and answer any questions you are asked about your codes. The idea is that you should fully understand what your programmed. If you want to test your code for just one test file, you can try: ./RadixSort.exe < ../testfiles/t1

Warning: Never ever change files under the "testfiles" folder or GradeMe03. If you do so, it will be considered as a SERIOUS CHEATING.

Input structure 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.

Examples of input and output

Example:
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, the above output example has 6 lines since I did "cout << endl;" at the end of each of the first 5 lines; those are the only white characters.

You can see the grader's output files, o1, o2, $\cdots$ in the testifies folder for more details. The above toy example is in t0 Perhaps you want to try

./RadixSort.exe < t0 > my0
diff my1 ./testfiles/o0

This will show the difference between your output and the output file that the grading toolkit uses. It is always a good idea to start with a simple example when testing your code.

Good luck!

Your solutions Before leaving the lab, submit a zipped tar archive of your program through the assignments page of CatCourse. Please use your UCMNetID as the filename for the zipped tar archive. Be careful since CatCourse strictly enforces the assignment deadlines (deadlines will be every lab date at 10:30am or 1:30pm depending your sessions).