CS 211 Data Structures and Algorithms Lab
Autumn 2022

Assignment no.	3
Objective	To implement Radix sort
Total marks	10
Due date (without penalty)	5th September (Monday) 11:59 pm
Cut-off date (with penalty - 5%)	8th September (Thursday)11:59 pm
Penalty for violating naming convention(s)	5%

Command-line argument:

Your program should receive a file (input file) and an integer as command line arguments. A typical execution will ./a.out input.txt 5

Input file:

The input file will be a text file where each line contains a non-negative integer which has exactly **d** digits where **d** is the second command-line argument.

Task:

Implement Radix sort to sort the integers in the input file in ascending order. The output must be a file named ***'radix.txt'***. Every line should contain exactly one integer (the first line contains a smallest integer, and so on). The sorting technique used inside the Radix sort is preferred to be Counting sort, i.e., **Counting sort should be used to sort every set of significant digits.**

Submission:

- The program you submit should output ***radix.txt*** when run.
- The main file of your program should be named as <roll no>.<extension>, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C is mandatory for this assignment). Ex: 210010001.c.
- Do the stress test of your program well before submission.
  (i) You may use the attached sample input files for testing; the corresponding output files are also attached;
  (ii) We have some hidden inputs with us to test your program. *The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs*.
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is important that you follow the ***input/output***

conventions exactly (including the naming scheme) as we may be doing an automated evaluation. ***There will be a penalty of 5% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.***

- Follow some coding style uniformly. Provide proper comments in your code.
- *Submit only through moodle*. ***Submit well in advance***. Any hiccups in the moodle at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. ***Copying others' programs is a serious offense and a deserving penalty will be imposed if found.***

Evaluation:

- To consider for the first evaluation without penalty, you have to submit your program by the due date. If you submit after the ***due date*** but on or before the ***cut-off date***, ***there will be a penalty of 5% on the marks you deserve otherwise***.
- If you do not submit by the ***cut-off date,*** your program will not be considered for the first evaluation.
- We will do the first evaluation after the cut-off date. *The marks you obtain will be proportional to the number of correct lines in the output files*. We will use the ***'diff'*** program to check the differences between the correct output file and the output file generated by your program. So, you may verify the correctness of the output file by using the diff program with a sample output file before submission. (See the man page of diff for more info).
- We will do the second evaluation later. This is for those who want to improve their marks obtained in the first evaluation or who do not submit for the first evaluation. There will be ***a penalty of 20%*** for those who are submitting for the second evaluation. The details of the second evaluation will be shared later.