

This assignment is to alphabetically sort n strings, each with maximum-length of $k = 21$ characters. (You may assume $n \leq 1000$.) The algorithm to be used is LSD (Least-Significant-Digit-First) radixsort. The algorithm must run in $O(n)$ time, where one character operation takes one unit of time. You must handle the strings on a *character-by-character basis*, one character at a time. (Each character operation takes $O(1)$ time.)

The strings consist of only upper-case letters A-Z, with no space. The strings need not be distinct. (For example, there may be several PATEL in a class list!)

Each string has at most k characters. Shorter strings must be padded with blank characters (as they are read from a file) to reach the fixed length of k .

First, read the strings from a text file “f.txt”. Assume that each string starts at the beginning of a new line and ends with one or more white-space-characters. (An end-of-line character is a white-space.) Store the strings in a two-dimesional array S with n rows and k columns. That is, string i is stored in $S[i, 1..k]$.

For efficiency, once the strings have been read into array S , they must stay in that order and not be moved during the sorting. Instead, we use a pointer array (array of string indices) $P[0..n-1]$. Initially, set $P[i] = i$ for all i . During the sorting, we move these indices. At the end of sorting, $P[0]$ will be the index of the string which is the first in the sorted order, $P[1]$ the second string in the sorted order, and so on.

At the end, output the strings in sorted order to a text file “g.txt”. Use the same format as the input file, with each string as one line. (Each string must begin at the beginning of a line and end with an end-of-line character.)

Notes on naming files: For ease of grading, your program must receive the names of input-file and output-file from the command line. And these file names must have default names “f.txt” and “g.txt”, respectively. Then the user will be able to either specify a file name, or decide to go with the default name.

That is, your program will first receive the name of input file.

Please specify the input file (default = f.txt):

After receiving the name of input file, your program will then receive the name of output file.

Please specify the output file (default = g.txt):

Make sure you adhere strictly to this protocol so the TA will be able to easily specify his/her own file name to run your program.