

## CS 361 – Homework #2 – School Sort – Due October 22, 2018

The purpose of this assignment is to give you practical experience sorting a large amount of information. Write a computer program in either Java or Python that reads the file `school.txt`, which you can find on the class Web site. Each line of this file refers to a public school in the United States. Store the schools in some kind of list. Sort this list in two different ways. The output of each sort should go into a separate file.

You may assume the following format about each line of the input file. The table indicates the width of each field by indicating the caret position of the beginning and end of each field. Note that the fields are separated by a single space.

Meaning of field	Justification	Starting column	Ending column
State and county	Left	0	18
ZIP code	Right	19	24
Low grade (e.g. kindergarten)	Right	25	27
High grade (e.g. 12)	Right	28	30
Name of school	Left	31	54
Enrollment	Right	55	60
Latitude	Right	61	69
Longitude	Right	70	78

Your first sort will be written to a file called `zip.out`. In this file, the schools are to be sorted in the following manner:

- By state and county ascending
- Then, within a county, by ZIP code ascending
- Then, within a ZIP code, by High grade descending
- Then, within High grade, by enrollment descending

Your second sort will be written to a file called `lat.out`. In this file, the schools are to be sorted the following way:

- By state and county ascending
- Then, within a county, by High grade descending
- Then, within High grade, by latitude descending

Some stipulations and hints:

- a. Your program will be tested in a Linux environment, e.g. the server `cs.furman.edu`. You need to make sure that your program will compile and run in this environment. If you are using Java, then your program should compile with the command `javac *.java`. Do not create a Java package.
- b. Your program should adhere to good programming style. Do not indent excessively. Limit source lines to 80 columns. There should be a comment at the top of each source file, and above each

function. Other comments in the code should indicate the important choices you are making in your algorithm, and indicate to the reader “where am I” in the code. Comments should also take note of assumptions that the program is making, and what steps in the algorithm are being accomplished. If your program does not exhibit acceptable programming style, it will not be evaluated.

- c. You may assume that state and county may be combined into a single field, as shown in the table above, since the input file already has the schools sorted by state abbreviation and county.
- d. I encourage you to use your programming language’s built-in sorting routine in the run-time library, so that you don’t actually write the sorting algorithm yourself.
- e. On each line of the output, the school needs to appear exactly how it appears in the input file. In other words, there should still be one space between each field of data, and the line lengths should not have changed.
- f. In order to accomplish the sorts, your program should follow an approach akin to radix sort. Since you need to sort on multiple fields, begin sorting by the least significant field, and finish by sorting on the most significant field.
- g. When your program is finished writing to the two files, it should print a message on the screen announcing that it’s done.