Max Ashkenazi
Kyle Shin

Design
To tackle this assignment, we stored the information as an array of structs consisting of only strings. We then perform merge sort using two methods that allow us to recursively tackle the problem. To do this, we kept track of the first, middle, and end of the initial array and split it at the middle into two arrays of structs. Then when it's separated into arrays of size two, we would compare the type we're searching by and then reorganize based off that. For certain columns, we use atoi or atof in order to compare by int or double as strcmp doesn't work in certain cases.

mergesorthelper/main - to code merge sort, we went through it recursively while keeping track of the initial array and the first, middle, and end index for the start and end points of the array of structs we split the original array into. At the end of the splitting process, it'll compare with the array next to it and we essentially rebuild the original array with the sub arrays. This should be O(nlogn) where n is the amount of rows we're dealing with.

main - This was done by making an array of structs for each line (n) and then assigning strings to each column (m) so it should be O(mn). Then it performs mergesort which as stated before is O(nlogn) and then it goes through the array of structs to print to the file O(mn). Total run time of this program should be O(mn + nlogn).

Assumptions
We're assuming that all the files in the test csv's used for grading will be in the same order as the mock one we're given.

Difficulties
There were some difficulties with accessing forbidden memory in merge sort which is why the main method needs to have the amount of rows - 1. Dynamically sizing the array of structs was also an issue as we hard coded a large size for the array. We bumped into a couple trimming issues too such as with quotes in movie titles and some unexpected spacing after sorting.

Testing procedure
We ran the code initially with the mock csv and director_name. Then we run again with that output and another type to sort by until we've gone through all the types to sort by.

How to use
In terminal:
gcc sorter.c mergesort.c -o sorter
cat <filename> | ./sorter -c <type to sort by>