

#####READ ME#####

Project 0: Sorter  
-sorter.c  
-mergesort.c  
-Sorter.h

This program takes in a CSV file containing IMBD data,  
and sorts it by a field of the user's choice.

All inputs must be in the form:

```
cat input.file | ./sorter -c FIELD
```

where field may be:

color  
director\_name  
num\_critic\_for\_reviews  
duration  
director\_Facebook\_likes  
actor\_3\_facebook\_likes  
actor\_2\_name  
actor\_1\_facebook\_likes  
gross  
genres  
actor\_1\_name  
movie\_title  
num\_voted\_users  
cast\_total\_facebook\_likes  
actor\_3\_name  
facenumber\_in\_poster  
plot\_keywords  
movie\_imdb\_link  
num\_user\_for\_reviews  
language  
country  
content\_rating  
budget  
title\_year  
actor\_2\_facebook\_likes  
imdb\_score  
aspect\_ratio  
movie\_facebook\_likes

All outputs are printed to the stdout in CSV format.  
All fields with a ',' inside are printed out surrounded  
by quotes to protect proper CSV formatting.

Design:

We chose to use an array of struct pointers to store our  
data, as this made for an easier time swapping the arrays.

We chose to use `strcasecmp`, as using `strcmp` put lowercase letters after all uppercase letters, which put them out of dictionary order place.

Memory leak free according to `valgrind`.

We reject all improperly formatted arguments.

All empty number fields are set as `"-1"`, to differentiate between an empty input value, and a value of 0.

All empty strings are left as is.

Mergesort was put into it's own C file, as we felt it could be retooled for some later purpose.

`Sorter.h` contains the struct needed to store our movie data, as well as a number of function declarations. The structs and functions are the ones that are called in both files, `sorter.c` and `mergesort.c` like `mergesort` or `mergesortHelper`, or ones that could be useful for other projects, such as `trim`, `compareStrings`, or `printCSV`.

Other:

There is a problem with our given input file, that causes unintuitive results. Movie titles have a weird hexadecimal symbol following them, that is sometimes invisible, which causes things like `"Zoolander "` and `"Zoolander 2"` to be sorted in an incorrect order.

This is due to the `strcmp` reading the extra symbol, and placing it before the regular space located in `"Zoolander 2"`. In an input file without this weird symbol, these would print as intended.

Using `print` statements, we were able to print out only the sorted fields, which made testing for this program relatively easy. We could very clearly see if our fields were sorted in proper order.