Suva Shahria-ss2479, Jerry Zhang -jz570

Difficulties:
We made a trimming code that successfully trimmed whitespace from the header file. When we tried to trim the whitespace from the rest of the csv we were unable to. After many attempts we were unsuccessful. We commented out the attempts.

Our merge algorithm works but fails to compare the middle rows to each other. To compensate for this in main after we call mergesort we compare our middle rows together resulting in proper arrangement.

Sorter.c:
Our main file extracts the csv file, calls tokens calls which calls tokenizer to make tokens of csv file for each row. The tokens are separated by “,”. If there are “” we remove the “” and take the text as 1 whole token.
Our code takes in the user input of category and finds the column that is being referred to.
It then calls mergesort()

mergesort:
Our mergesort recursively cuts our rows in half and compares them to each other. When comparing the text there is an int check,isNumber is called to see if the string is an int. We store the int as a double, our code allows decimal values such as 7.4.
We than compare which int is bigger and put the higher one on the bottom.
If the text is not an int we compare the strings instead.

Header:
The header allows the user to sort by whichever category the user wants. The header also gives information about what to expect to find from this list. It’s like a trailer to the movies.

Assumptions:
We assumed we should print the 1st column.

To run our code we went to the folder the files were located in, with the csv file in same folder.

```
Gcc -o test sorter.c mergesort.c | cat m2.csv | ./test -c color >tmp2.csv
```

To test we made a csv from the main file. It’s called m2.csv and provided.

Extra Credit #2:
We believe our code is generalized for any csv file. We didn’t hard code the number of rows, columns, or the header categories. Our code gathers this information on its own. Since we collect this data we can sort any csv file based on columns of any size. Our code doesn’t check for negative numbers. We could easily implement this feature in the isNum class in mergesort by adding an if statement to check if the 1st char of the string is “-” and then proceed from there.