Tanvi Mankame & Dhruv Duggal
Systems Programming
Assignment 1

Files in the Project:

- SimpleCSVsorter.c: The simpleCSVsorter is broken down into several smaller methods that help it run efficiently. The main responsibility of the simpleCSVsorter.c file is to sort the CSV files, fork through directories and identify which files are CSVs and which ones are not.
    - main() - The main function reads in the input in the command terminal and picks up the parameters that will be used for the whole program. After the main() function picks up the column that will be sorted, the file input directory, and the desired output directory, it sends those values into the ReadDirectory() function.
    - ReadDirectory() - This method traverses through the directory given by the user to find CSV files to sort. It also accounts for directories within directories which may also have CSV files. In short, it has two parts to it: the first part being that it finds a directory and the second being it finds a CSV file. If it finds a directory, it will either recursively call the method again until it finds a CSV or gets to the end of the file. If it finds a CSV, it will quickly check if it is a CSV and then run the sorting algorithm on it.
    - isCSV() - a simple method that returns 1 if the file is a CSV and returns a 0 if it is not.
    - Sorting() - This method takes the column that each file needs to be sorted with as a parameter and proceeds to go through every CSV file that is passed through it and stores it into a linked list to be sorted either alphabetically or numerically.

- Mergesort.c: Mergesort has three helper methods for the main mergesort method. It has a Split() method, a Mergesort() method which is the actual sorting method, it has three subcategories: SortingInts() which sorts integer values, SortingStrings() which alphabetically sorts strings, and SortingDoubles() which sorts double/float values. The last helper method is a simple print method called PrintStrings() that allows us to see the output after mergesort, split, and sorting methods are called.
    - Split() - This method splits the linked list into two halves to make it easier to sort using the Mergesort() method.
    - Mergesort()- This method actually implements mergesort and calls upon Split() and SortingStrings() to help break down and build the sorted linked list up again.
    - SortingStrings/SortingInts/SortingDoubles - This method goes node by node to sort according to the specific category.
    - PrintStrings(): This method takes in the CSV files, and outputs them into newly named CSV files in the desired output directory.

- SimpleCSVsorter.h: SimpleCSVsorter.h is a header file that holds all the methods we use in the Merge our project, summarized by a short blurb of what each function is supposed to do.
- Makefile: Makefile compiles both mergesort and simpleCSVsorter at the same time and allows a more efficient way to test our code. In order to compile, just type ‘make’ into the terminal.

DESIGN ELEMENTS:

- One of our design elements consisted of us using realpath() to make the directory and file paths correct and up to date.
- We also used strrchr, strcat, strcpy to keep track of and manipulate certain paths to fit our agenda.

Assumptions:

- At first we assumed that the user would always input a specific input/output directory but later we changed our code to reflect the possibility that the user does not input any specific details about the directories. Because of this we also had to update our error messages regarding incorrect inputs because there were several cases that would now work as correct inputs that might not have previously.
- We also assumed that because the user would input an output directory location, it would create a folder by itself. We later learned that there is a function in c called makdir() that creates a directory for you in the code so that we could output our sorted CSV files in it.

Difficulties:

- The main difficulty we had was with forking without creating a fork bomb. After we figured out where to fork to get the output we wanted, the next concern was how to fork within several directories.
- We had a problem with our sorting method when it came to outputting sorted CSV files that were found in other directories. We solved it by specifying the path the sorted file would be outputted to.

How to Run the Program:

- The “make” command will compile the program by implementing “gcc -Wall -Werror -fsanitize=address <filename> -o filename.c. Because our Makefile has several c files in it, the make command will compile all of them simultaneously.
- The next Command will be
    - ./simpleCSVsorter -c <column_name>
        - This is the command for just sorting all of the CSV files in a current directory with the output being the same directory
    - ./simpleCSVsorter -c <column_name> -d <directory_path>
        - This command is for sorting all of the CSV files within a specific directory to be sorted by the column given, with the output being the same specific directory.
    - ./simpleCSVsorter -c <column_name> -d <directory_path> -o <output_directory_path>
        - This is the command for sorting all the CSV files in a specific directory by the specified column name and outputting them into a specific output directory. If you input just a output directory name, the directory will be created in the source directory, unless specified otherwise.