fileSort.C - A User Manual

“fileSort.C" was a collaborative effort by Advith Chegu and Savan Patel that allows for a user to input one of two sorting methods and a comma-separated text file consisting of either numbers or words whilst outputting the values of the text file in its sorted order.

Sorting

The two sorting methods that are implemented in this program are insertion sort and quicksort, both having different runtimes depending on the organization of the text file. Insertion sort was implemented iteratively and with the use of a helper method while quicksort was done recursively. Quicksort chose the first node of a linked list to be the pivot and to be sorted in its proper spot depending on the situation. The function prototypes for both are given here.

```
int insertionSort(void* toSort, int(*comparator)(void*, void*));
int quickSort(void* toSort, int(*comparator)(void*, void*));
```

Reading the File

The text file itself had to be comma-separated and accounted for trailing spaces or tabs. Capital letters, degenerative dashes in numbers, or mixed text files were not considered valid text files in the assignment description so those attributes were not accounted for. For this assignment, the “f functions” were not allowed as they are intended for larger file streams. Instead, functions like “open()”, “read()”, and “close()” were implemented and those specific functions utilize the file descriptor nature of Linux.

Helper Functions

As our program read these files, it placed them in a character array buffer which was statically allocated and expanded if needed to. Helper methods were then used to clean out the buffer and remove trailing spaces.

```
void cleanRead(char*, int);
```

Then the buffer was read and loaded each comma separated value into nodes which formed a linked list of how many values there were in the original text file. Then, depending on the sorting algorithm selected, that was done on the linked list which utilized a string or integer comparator method that was passed into the sorting algorithm as a function pointer.

```
int loader(struct node*, char*, int);
```

Each sorting method also took in a void* which we interpreted type casted to be the first node in the general unsorted linked list. The linked list was then sorted and each value was printed onto the screen followed by a new line. Certain errors were also accounted for depending on the assignment description.

Usage

The program should be compiled with the following flags found in our makefile. We compiled using the gcc compiler.

```
gcc -g -o fileSort fileSort.c
```

It can then be run using the following commands

```
./fileSort -q testFile.txt
```

Where -q is the quicksort flag (-i) for insertion sort and the testFile is the name of the file you would like to sort.

Contributing

This C program was written by Advith Chegu and Savan Patel, please adhere to the Rutgers University Academic Integrity Policy while using this program.

License

MIT