Purpose

The purpose of this program is to load car data from a file, format and print it out to the console, save the data to a separate file, sort in order of ascending price, give the user cost estimates for all available cars for any number of days, rent a car, and exit all under user direction from a terminal on a Linux environment. I designed this using MinGW, nano, and clang-format.

Design

My program uses a state struct to pass around data to cli functions. It uses bubble sort to sort the data and used two switch statements and a few if else statements to dispatch the user’s request to the right cli action. It exists in a while loop that watches one of the app state variables, which the sit command modifies to exit the program. The car data is stored in an array of structures. The max filename that the program can process is 255 chars long. The program can recover from invalid inputs.

Problems

The instructions were ambiguous in a few instances so I guessed what I was supposed to do. I still didn’t know how long I should have made the filename buffer. I cloned the entire data array to sort for one of the cli commands because the instructions seemed to imply I was not to modify the original array. I had no other problems.

Changes

I would work on improving my formatting code to be more generic to reduce code duplication. I would also make cli option 5 more efficient since it copies the entire array before sorting, when I should just make it copy the available cars. I would also consider reworking the command dispatch code as it requires 2 switch statements when I can probably get it down to one and a smart use of a couple if else statements.