CS367 Lab 8

For this assignment you will write a program that determines which of a set of rental cars the user can rent for a user-specified rental rate. Your submission should follow the instructions for program organization and **Moodle** submission given in Lab **7**.

Your program must be written entirely using pointer syntax; i.e., it cannot contain ***any*** **[]**'s (*hint:* the declaration **char* argv[]** can be written **char** argv**). Remember to follow all of the dynamic memory allocation/deallocation "rules" given in class, including remembering to free any dynamically allocated memory.

Your program must meet the following requirements:

1. Define a **Car** structure type in your header file. The **Car** structure should have three fields: a model (a string), a year (an integer), and a daily rental rate (a floating point number). The model field string will be dynamically allocated (see function **2(b)** below).

2. Your program should define the following functions:

a. A function that takes a single **Car** argument. The function should output the fields of the **Car** in a sentence. The rate should be formatted as money.

b. A function that takes a single **Car** argument. The function should first dynamically allocate the **Car** model field large enough to hold a string of at most **50** characters in length. It should then use a single **scanf** to read the next line from **stdin**, consisting of:

modelName year rentalRate

and place the values into the appropriate fields in the **Car** argument. The function should return **0** if successful and **1** if an error occurred in reading the line. This function should not perform any output.

c. A function that takes as arguments an array of cars and the maximum number of cars that array can hold. The function should fill the array with the indicated number of cars using calls to your function **2(b)**. The function should perform reasonable error checking and should return **0** if successful and **1** if an error occurred. This function should not perform any output.

d. A function that takes as arguments an array of cars, the number of cars in the array, and a desired rental rate. The function should print-out all cars in the array with a rental rate less than or equal to the specified rate, calling your function **2(a)** to perform any car output.

3. Your **main** function should do the following:

a. Your program should take one command line argument which represents the desired rental rate.

b. Read the number of cars to be input from **stdin**. Exit if the value is invalid or negative.

c. Dynamically allocate an array of **Car**s exactly large enough to hold the specified number of cars (see **3(b)** above).

d. Fill the array with cars from **stdin** by calling your function **2(c)**. If an error occurs give an informative error message and exit the program with a status value of **1**.

e. Call your function **2(d)** with the desired rental rate (which was specified as a command line argument).

You can test your program by feeding it a properly formatted text file using **stdin** redirection (or a pipe, as you prefer). Three such files are available on cs367.sou.edu in **/usr/local/cs367/cars.tar.gz**: **cars1.txt**, **cars2.txt**, and **cars3.txt**.

Run your program three times with a command line argument rental rate of **$50.00**, and feeding it each of the rental car data files **cars1.txt**, **cars2.txt**, and **cars3.txt.** (*hint:* one or more of these input files may contain an intentional error that should cause your program to exit with an appropriate error message during input processing). Submit your output from these runs along with your source code files and Makefile.