CIS*2520— Assignment #2

Fall 2024

Due: Friday, October 18, 2024@ 23:59

Please submit your assignment solutions as **one zip file** (named as YOUR/UoG/ID_a2.zip, e.g. 1234567_a2.zip) to Dropbox under Assignment 2 before the due date.

You are granted a penalty-free grace period for 48-hours. The grace period ends on Sunday, October 20 23:59. After this time, you cannot submit the assignment. The late assignment (no submission after the grace period) will be marked as ZERO.

If you require a longer grace period (than the default 48 hours) for your assignment, the request must be sent to the course email: cis2520@socs.uoguelph.ca (using your UoG account) BEFORE the due date (Friday October 18, 2024@ 23:59). Please refer to the communication guidelines in week 0 on writing emails.

Please refer to the Course Outline and Academic Integrity Video to ensure you understand and comply with the University's Academic Integrity Standards.

1 Rental Car Linked Lists

Write a program in which linked lists are used to maintain the data structure for a car rental company. The program allows several types of transactions to be applied to the data structure in order to keep the lists up to date.

In the program three linked lists are maintained for cars:

- available-for-rent,
- rented,
- in-repair

Each car contains three pieces of information: the plate number, the current total mileage, and the expected return date (only valid for the rented list, the other two have -1 as the value). The cars on the available-for-rent list are ordered by mileage, with the car having the least miles at the front of the list. The cars on the rented list are ordered by the expected return dates, with the first car on this list having the earliest expected return date.

The program prompts the user for a transaction code (integer) as follows:

1) add a new car to the available-for-rent list,

2) add a returned car to the available-for-rent list (the rented list needs to be updated as well, and you may view it as transferring the car from the rented to the available-for-rent),

3) add a returned car to the in-repair list (may view as transferring the car from the rented to the in-repair),

4) transfer a car from the in-repair list to the available-for-rent list,

5) rent the first available car (may view as transferring the car from the available-for-rent to the rented),

6) print all the lists,

7) quit.

When users select a code, the prompts with different transaction codes shall be:

- For the new car addition and the return transactions (codes 1 - 3), the program should then prompt for a plate number (character string) and a mileage (integer). Users will input the plate number and the current total mileage.

- For the transfer transaction (code 4), the program should then prompt for a plate number (character string).

- For the rent transaction (code 5), the program should then prompt for an expected return date (integer: yymmdd).

- For the print transaction (code 6), the program should not prompt any additional information.

- For the quit transaction (code 7), the program should quit the current and not prompt any additional information.

Different messages should be printed along with the selected transaction code,

- For each transaction processed, a message should be printed indicating what action is taken, for example, which car is transferred from which list to which list.

- For each return transaction (codes 2&3), a **charge** is computed and **printed**. The charge is calculated as follows:

    - ❖ a flat rate of $80.00 for up to 200 km,

    - ❖ 15 cents per km for the additional (i.e. beyond 200) kilometres.

Other requirements for the program:

A. The program shall check if the input **mileage** from the user is a **valid** number, e.g., when a

care is returned, the current total mileage should be larger than the previous stored mileage of this car.

B. The program shall inform the user with a printed message on the required format of a car plate number, and also check if the user input satisfies the requirement. The given **requirement for plate number** is a length of 2-8 characters with only numbers and letters (capitalization does not matter).

C. The program shall check if the user input is a **duplicate plate** and print proper messages.

D. The program shall check if the **return date is valid** in the rent transaction (code 5).

E. When a quit transaction is completed, the program shall **store** the data in the three lists into the three relevant **text files (available.txt, repair.txt, and rented.txt)**, and when the program restarts, the program should read the data from the files and restore the lists. The filenames can be hardcoded in the program as we do not expect changes of the file names.

F. The program should be able to reasonably detect other error conditions (such as invalid transaction code, and an attempt to return a non-existent car) and print appropriate error messages.

2 Reverse Polish notation (RPN)

An expression is in postfix form (**Reverse Polish Notation**) if the operators appear after their operands, for example, "1 2+" means "1 + 2", and "1 2 + 5 3 − *" means "(1 + 2) *(5 − 3)". Write a program that evaluates postfix expressions composed of **single digits** and binary operators of +, −, *, and /, for addition, subtraction, multiplication, and division respectively.

The program takes an expression as a command-line argument, with no space in it, for example, *q2 12+53-**, where *q2* is the name of the executable (Note: there is a space between *q2* and the expression *12+53-**). The program prints a result that has two digits after the decimal point.

You must use **linked list implementation of stack** to solve this question. Your program should be able to reasonably detect error conditions (such as invalid input string) and print appropriate error messages.

Assignment 2 Guideline

- You must implement **singly** linked lists for **both questions**.
- Template files are provided for you to work on this assignment. You **cannot** change the names of the template files, and the names of the functions in the template files.
    - *q1.c, q1.h,* and *q1_functions.c* are for question 1.
    - *q2.c, q2.h,* and *q2_functions.c* are for question 2.
    - *readme.md* is the template for the readme file. You can add more info to it as you want, such as the testing input/output from the samples given to you.
- You must create a **makefile** (one single makefile) for compiling the programs. The makefile should be able to generate two executable files, for example, *a2q1* and *a2q2*.
- For q1, sample files of available.txt, repair.txt, and rented.txt are given to you. For q2, testCaseQ2.txt include some testing cases. But please note that for q2, the program shall take an expression as a command-line argument, **not** reading from the given file.
- For this and future assignments, you must compile with "*gcc -Wall -std=c99 -pedantic*". **The code must be tested on the school server before submission**.
- **Submission requirement**:
    - Only the following files are required and should be submitted. All the files should be under the folder named YOUR/UoG/ID_a2.
        - *q1.c, q1.h,* and *q1_functions.c*
        - *q2.c, q2.h,* and *q2_functions.c*
        - *readme.md*
        - *makefile*
    - No other files should be included in your submission, such as .o files, txt files, etc..
    - Once you have everything, zip the folder and submit it via CourseLink Dropbox.
- For grading:
    - q1 is weighted 54%, q2 is weighted 36%, and 10% is for style, comments, documentation, readme, and makefile. For example, in each of the C files, any

function should have a brief comment describing its purpose. Also, any section of code where it is not easily apparent what the code does should have a short comment. Don’t forget indentation.

    - Any compilation error or warning will result in a mark deduction appropriate to the severity of the error. Memory leaks and memory errors will also result mark deductions. A maximum of 10% could be applied to deductions.

- The given *CIS2520-F24-A2.md* is prepared by the instruction support team and can be viewed as an additional source of guidance.