**King Saud University**
**College of Computer and Information Sciences**
**Computer Science Department**
**CSC215: Procedural Programming with C**
**Second Semester 2022-2023**

**Project**

---

## Project description:

Write a C program that represents tracking the availability of cars taxies. The system keeps track of the following trip car's information: ***id, driver, category, plate, color, rate, mincharge*** **and** ***state.***

The program should create a dynamic linked list that keeps track of all the taxi cars with their information.

Example of the list of information for cars taxies:

| ***id*** | **driver** | **category** | **plate** | **color** | **rate** | **minCharge** | **state** |
|---|---|---|---|---|---|---|---|
| 111 | Ali | Business | WED500 | Silver | 4.5 | 50.00 | R |
| 222 | Omr | Family | ABC234 | Black | 5.0 | 20.00 | A |

…… and so on.

## The requirements:

1. Download the txt file ***"Taxies.txt"*** which contains the cars trip information.
2. Create a structure called: ***Taxi,*** that contains the following information:
   - ***id****:* represents the driver's id as integer.
   - ***driver:*** represents the name of driver (first name and last name) as string with length 20.
   - ***category:*** represents the category of the car as string with length 20. The category could be one of the following: "***standard", "Business" or "Family".***
   - ***plate:*** represents the plate number of the car as string with length 7. The format of the plate number: 3 alphabet characters and 3 digits.
   - ***color:*** represents the car's color as string with length 20.
   - ***rate:*** represents the driver's rate as float.
   - ***minCharge:*** represents the minimum charge of the trip car according to the car category as float.
   - ***state***: represents the trip car's state as character**.** The state could be "***A"*** which indicates the driver is available and ready for a trip, or ***"R"*** which indicates the driver is in a ride currently.
   - ***next****:* which is a self-referential structure pointer.
3. Declare a pointer: ***list*** of type *struct Taxi***,** which points to the created linked list.
4. Implement the following functions:

- ***void addTripCar()*** to add all trip cars information to the created linked list by reading all the content of the given text file: ***"Taxies.txt",*** where each line in the text represents a single trip car record as shown in ***Figure.1***. Then, it adds the read records to the linked list. While adding trip cars to the linked list, set the variable ***state*** in each car record to **'A'** to indicate that all taxies are available and ready for a trip.

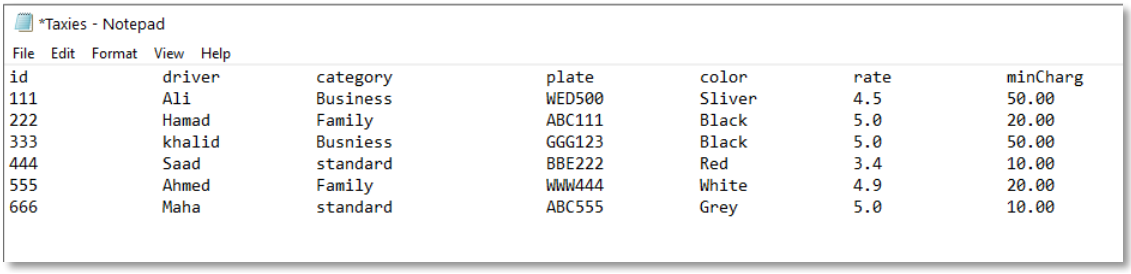


Figure 1: The content of Taxies.txt

- ***void setTripCar (char* category, float* rate):*** The function searches for a specific trip car given the car's category and the driver's rate. If the car is found, then it sets the state of trip cars to ***"R"*** to indicates the car is currently in a trip.
- ***void writeCarsInRide (char* fileName):*** The function updates the file ***"Taxies.txt"*** by appending the following information of the trip cars ( ***id, driver, category, plate, rate and state)*** which are in a trip with state ***"R"*** as shown in the ***Figure 2*** below.
- **Note:** follow the same format as shown in Figure 2: add the header of each column (i.e, ***id, driver, category, plate, rate and state)*** and add each record in a separate line. Also add the header*:" The trip Cars"* and notice each column( each data in a line) separated by tabs.

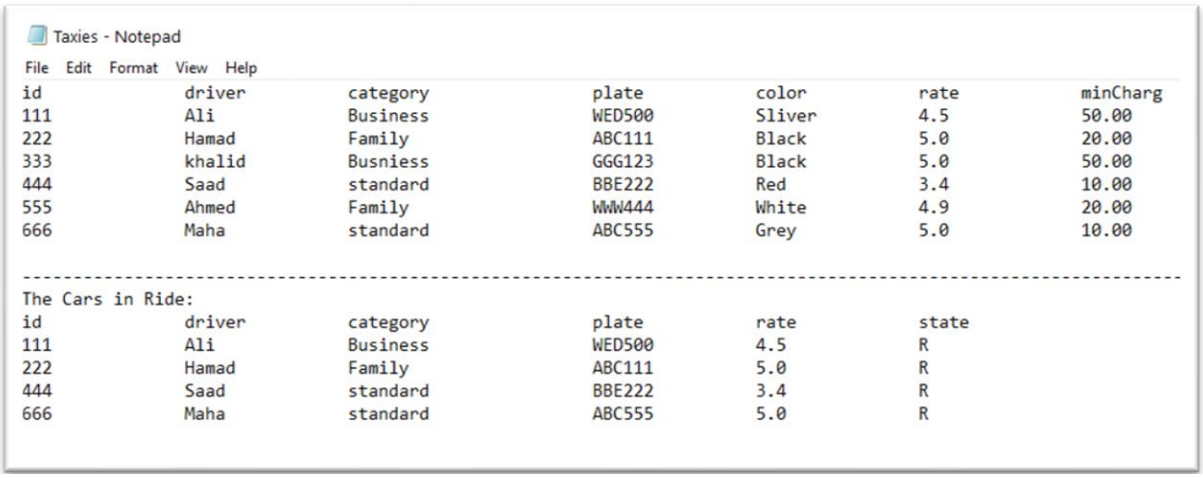


Figure 2: The content of Taxies.txt after appending trip cars

- ***printList():*** The function prints on the screen the trip cars information that stored in the linked list as shown in ***Figure 3***. It prints each records in a single line whith their the header of each columns(***ex, id, driver, category … )*** separated by tabs (i.e, data in a line separated by tab).

**5.** In the main function do the following:

- Call the function ***addTripCar*** to read the content of written trip cars information.
- Call the function ***setTripCar*** with the following given shaded information:

| **Category** | **rate** |
|---|---|
| Business | 4.5 |
| Family | 5.0 |
| Family | 4.0 |
| standard | 3.4 |
| standard | 5.0 |

- Call the function ***writeCarsInRide*** to update the txt file with cars in trip.
- Call the function ***printList*** to print the cars information as early described, (i.e, after reading the initial information in the txt file and after set some cars to be in a ride) as shown in ***Figure 3***.

➢ **Notes:**

- Do not create any structures other than: the ***Taxi*** structure**.**
- All possible validation are required that are related to dynamic memory allocation, or file.
- You have to consider all possible cases related to the creating data structure linked list.
- Do not use the array [] brackets when dealing with pointer. Your solution should be with pure pointer notation.
- use the same given identifier name for the functions, structures, and variables.
- Your code should be clear and neat.
- Use indentation in your code.
- Use an appropriate identifier name.
- Add any needed standard libraries you needed.
- Your program should be free of compilation errors.
- Your program should be in **one file with c or txt extension.**
- **Do not zipped your file.**
- Any violation to mentioned instruction above will affect your project grade by deducing 1 point for each violation.

- **The Expected Output:**

```
The Available Trip cars:
111  Ali       Business  WED500    4.5  50.00     A
222  Hamad     Family    ABC111    5.0  20.00     A
333  khalid    Busniess  GGG123    5.0  50.00     A
444  Saad      standard  BBE222    3.4  10.00     A
555  Ahmed     Family    WWW444    4.9  20.00     A
666  Maha      standard  ABC555    5.0  10.00     A
-------------------------------------------------------------------------------------------
The Cars in Ride:
111  Ali       Business       WED500    4.5  50.00     R
222  Hamad     Family         ABC111    5.0  20.00     R
333  khalid    Busniess       GGG123    5.0  50.00     A
444  Saad  standard      BBE222    3.4  10.00     R
555  Ahmed     Family    WWW444    4.9  20.00     A
666  Maha standard       ABC555    5.0  10.00     R
```

Figure 3: The expected output of the content of linked list on the screen