In this project, you will implement a system, that is based on Linked List, to assign passengers to various busses to help them commute based on their scheduled times.

Each student (i.e., passenger) has the following information: traveler ID (the student ID), travelling date, travelling time, from, and to (destination). Following is an example of the input file called (passengers.txt):

```
1190001#18042022#14:00#Birzeit#Nablus

1190112#18042022#11:00#Beit Rima#Birzeit

1190554#19042022#13:00#Birzeit#Atara

1190201#20042022#14:00#Birzeit#Ramallah
```

Another file containing the registered bus lines (busses.txt) contains: a bus number, date, departure time, from, to, price of ticket, and capacity. Example of the input file:

```
1#18042022#14:30#Birzeit#Ramallah#6#15

2#18042022#11:45# Beit Rima#Birzeit#6#1

13#19042022#14:30# Birzeit#Atara#6#20

53#20042022#14:00#Birzeit#Nablus#6#7
```

Your application should do the following:

1. Read the drivers file and load it into an array, then read the second file (passengers) which should be added to the busses in a linked list (array of linked lists). Before assigning passengers to busses you should make sure that:
   a. Each passenger is added to a bus based on the date and departure time. Meaning, each passenger should match the bus based on his/her travelling time and the departure time of the bus
   b. The length of each linked list should be decided based on the capacity of the available bus
   c. If a student has no matching travelling date/time, there should be another linked list for those

Your application should be able to show the following information through a proper menu of the application:

1. Load the bus information file
2. Load the passenger information file
3. Assign passengers and print assignment information of all busses
4. Print a specific bus information along with its passengers information (names and IDs)
5. Print unmatched passengers
6. Add new passenger
7. Delete passenger
8. Delete bus number
9. Exit

Grading policy:

1. Your application should have all functionalities working properly. **Twenty** marks will be graded for the functionality of the project;
2. The following notes will make up the remaining 10 marks of the grade:

a. There has to be adequate documentation and comments in the code (i.e., functions, loops, etc.);

b. Your code should follow the code convention (i.e., spaces, indentations, etc.); and

c. Your application should contain a menu to allow the user to select which option (s) he would like to run.

Notes and submission instructions:

1. **This is individual work**. It should represent your own efforts. It is fine to discuss your work and to ask your colleagues, but you are not allowed to copy/paste the work of others or give your work to anyone else. You are not allowed to post/copy from other websites and/or social media and this will be considered as cheating.

2. Any **plagiarized** code will not be marked.

3. **Document format**. Please submit only the code file (**c** file) containing the code of your project. Please rename it as follows:
"**P1_YourStudentID_FirstNameLastName_SectionNo.c**".

4. **Input/output file name**. Make sure that the input/output file names are the same as in the specifications.

5. Include your full name, student ID, and section number in the beginning of your file.

6. Please do not compress the file, only the C-file is needed.

7. Files not following the convention in point 2 will not be marked.