BİL 113/012 Computer Programming I

HOMEWORK 5 [50 Points]

To be demonstrated in December 4, 2022

1 [30 POINTS] MR. BIDIK IS BECOMING A BUSINESSMAN

In the last 15 months, Mr. Bıdık majored in finance and made series of very profitable trades, and now he is a millionaire. His new interest is in the transportation industry and he is now considering to buy Mersin Vif Tourism, a bus company. If he buys the company he will change its name to Bıdık Tourism. Before making his decision he wants to increase his knowledge in the transportation industry and wants to have a kiosk machine to play with. He wants you to write the necessary software for the kiosk machine in Java, and all he will pay you for this job, 35 points provided that he likes your program. The specifications are as follows:

The kiosk machine only have a touch-screen which can be used by the users to input alpha-numeric characters. Machine is to be used by the customers for bus reservation and cancellations. Bıdık Tourism will have 5 buses at the start. Mr. Bıdık wants you to write the two classes *Bus* and *Trip*, data members of which are given in the table below. You can define additional variables.

Bus Class

Acces Level	Type	Name
private	String	type
private	int	model
private	int	age
private	int	capacity
private	int	remainingCapacity
private	boolean	type
private	boolean[][]	seats

Trip Class

Acces Level	Type	Name
private	String	origin
private	String	destination
private	String	departureTime
private	String	arrivalTime
private	String	date
private	Bus	assignedBus

The program should have a third class for user interface purposes and work according to the following description. Program should display a menu that allows the user to choose from the following two alternatives i) make a bus reservation, ii) cancel a bus reservation. Once the user entered his/her choice the program should list the five trips for customer to choose from as in below.

```
Available trips are shown below.
1)Trip Information:
        Date: 27/11/2022
        From: Ankara to Istanbul
        Trip time: 00:15 to 06:30
                Bus Information:
                Bus: Setra
                Type: 2+2
                Age: 8
                Capacity: 40 seats
                RemainingCapacity: 23
2)Trip Information:
        Date: 27/11/2022
        From: Ankara to Mersin
        Trip time: 13:00 to 20:00
                Bus Information:
                Bus: Setra
                Type: 2+1
                Age: 5
                Capacity: 36 seats
                RemainingCapacity: 11
3)Trip Information:
        Date: 27/11/2022
        From: Istanbul to Ankara
        Trip time: 07:00 to 13:30
                Bus Information:
                Bus: Neoplan
                Type: 2+1
                Age: 2
                Capacity: 48 seats
                RemainingCapacity: 38
4)Trip Information:
        Date: 27/11/2022
        From: Mersin to Ankara
        Trip time: 17:00 to 00:00
                Bus Information:
                Bus: Travego
                Type: 2+2
```

```
            Age: 15
            Capacity: 36 seats
            RemainingCapacity: 14
5)Trip Information:
    Date: 27/11/2022
    From: Istanbul to Mersin
    Trip time: 00:15 to 12:30
    Bus Information:
            Bus: Setra
            Type: 2+1
            Age: 15
            Capacity: 36 seats
            RemainingCapacity: 8
```

After a trip is selected, your program should display the status of the seats as shown in below for different types of buses and ask the user if he/she wants to make or cancel a reservation.

If the user is making a new trip reservation, the program should print the available seats for the user to choose from. After making the appropriate checks, the program should make a new reservation, create and print a unique PNR code, which will be needed for canceling the reservation, and finally, print the status of the seats again.

If the user is canceling a reservation, the program should prompts the user to enter the already reserved seat and a PNR code, and then cancel the reservation, provided that the PNR code matches with your records and the seat is actually reserved. Your program should print informative error messages for incorrect reservation and cancellations.

Your program should print the status of the seats as shown in the below for buses with 40 and 30 seats. The buses either have 4 seats in each row, type 2+2, or have 3 sets, type 2+1. You can assume that the number of seats in all the trips is either a multiple of 4 or 3 and the number of rows in all the trips is even.
An example of an type 2 + 2 bus with 40 seats, 5 of them which is reserved.

```
    AB  CD
1   T.  ..
2   ..  .T
3   ..  ..
4   .T  ..
5   ..  ..

6   ..  ..
7   ..  ..
8   ..  .T
9   ..  ..
10  .T  ..
```

In the bus above, seats 1A, 2D, 4B, 8D and 10B are already reserved, and the remaining ones are currently empty. Customers should be able to enter seat numbers in the printed format such as 6A, 7C etc.

An example of an type 2 + 1 Bus with 30 seats, 3 of them which is reserved.

```
    AB  C
1   T.  .
2   ..  .
3   ..  .
4   .T  .
5   ..  .

6   ..  .
7   ..  .
8   ..  .
9   ..  .
10  .T  .
```

Suggested methods for the class Bus are as follows:

```
public int numberOfEmptySeats(): Returns the number of empty seats
public boolean doesSeatExist(String): Checks if the given seat exist
public boolean isSeatEmpty(String): Check if the given seat is empty
public boolean fillSeat(String): Tries to fill a seat, returns true unless fails
public boolean emptySeat(String): Tries to empty a seat, returns true unless fails
public void printSeats(): Prints all seats
public void fillSeatsRandomly(int): Randomly fills number of seats given
                                    as parameter
public String toString(): Displays bus information
private int calculateSeatNumber(String): Converts a string in the proper format
        such as 1A to the corresponding seat index
```

You must design a user-friendly interface.

2 [15 POINTS] FINDING NASH EQUILIBRIA

In this question, you will find and print all *Nash Equilibria* for a given two player *game*, if they exist, or print a appropriate message if no Nash Equilibrium exists.

Game theory is the study of mathematical models of strategic interactions among rational players. It has variety of applications and mostly studied by economists, and more recently by computer scientist. There are many types of games, but in this question, we will focus on a special types of games called two player non-cooperative games.

In a two player non-cooperative game, each player has a set of strategies that he/she can choose. We will assume that first player has n strategies, whereas the second player has m strategies. A two player non-cooperative game can be represented by a matrix, called the payoff matrix, that has n rows and m columns, which represent the all possible strategy pairs of players. Each pair is called a strategy profile, and the cell of the matrix corresponding to each strategy profile is tuple x, y, which denotes the expected utility of the first and second players, respectively.

A strategy profile is called a Nash Equilibrium (NE) if no player can unilaterally deviate to strictly increase his/her utility. In other words, if no player can strictly increase his/her utility by changing his/her strategy while the other player sticking his/her strategy, then this strategy profile is a NE. Each game can have a single or multiple NEs, but there are also games that have no NE.

Consider the rock-paper-scissor game where each player can choose rock (R), paper (P), and scissor (S). I am sure that you are familiar with this game, but if you are not, you can look it up on wikipedia. In this game, each player has the same strategies. Rules are like this: rock beats scissors, scissors beat paper, and paper beats rock. If the players choose the same strategy, game ends in a draw. The winner obtains a utility of 1 whereas the loser obtains a utility of -1. If the game ends in a draw, both players obtains a utility of 0. The payoff matrix of this game can be model by the following payoff matrix. Notice that this game has no Nash Equilibrium.

	R	P	S
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

Consider the following game where each player has four strategies. From now on, we will not name the strategies but denote them with numbers. Rows and columns show the strategies of player 1 and player 2, respectively. This game has a unique Nash Equilibrium $(1,1)$.

	1	2	3	4
1	0,0	2, -2	2,-2	2, -2
2	2, -2	1,1	3,-1	3, -1
3	2, -2	-1, 3	2,2	4,0
4	2, -2	-1,3	0,4	3,3

Finally, consider the following game where each player has two strategies. This game has two Nash Equilibria: $(1,1)$ and $(2,2)$.

	1	2
1	0,0	0,0
2	0,0	1,1

In this question you will write a method that takes a two player non-cooperative game as input, and print all Nash Equilibria. If no Nash Equilibrium exists, then print that the given game has no NE.

Along with this, create a Demo class prompts the user for a game input and call the method you written. Input format for this demo is as follow. First line contains two integers n and m. Then the following n lines contains m two integer pairs a,b and there is a single space after each pair.
For instance, the following input corresponds to the last game.

```
2 2
0,0 0,0
0,0 1,0
```

Your method should print (1,1) and (2,2) for this input in a single line.