In this assignment, you will add some facilities to hotel reservation application you developed for Labwork 10.

You will convert your existing structures to the ones given below:

Room: roomNumber, typeOfRoom, availableOrNot[7], priceOfRoom

Customer: name, arrivalTime, departureTime, age, reservedRoom (of type **Room**), arrayOfOtherCustomers (array of customer names)

Hotel: myCustomerArray (array of Customer structures), myRoomArray (array of Room structures)

You will update the menu so that the new menu becomes the following:

1. Create a new reservation
2. Set departure time of a customer
3. Display total fee for a customer
4. Offer the available rooms.
5. Delete reservation.
6. Display the status of the rooms.
7. Display the number of people staying at the hotel.
0. Exit

When the application starts, you will read the current status of the rooms from a file, then populate myRoomArray accordingly. The format of the file is exemplified below:

1,1Person,1111111,20
2,1Person,1110001,20
3,2Person,0001111,50
...

The comma seperated file includes 4 columns, which specify the number, type, availibility and price of the room from left to right. The column designating the availability is a seven bit string. A bit of 1 indicates that the room is available whereas zero indicates that the room is in use. Because a reservation can last at most 7 days, the size of bit string and *availableOrNot* array of Room structure is 7.

In options 1, 2 and 3, operations should be updated conforming to the modifications on structures and their fields. While setting the departure time in option 2, you should check the difference between arrival time and departure time. If it exceeds 7 days, you should warn the user with a message saying that a reservation cannot exceed 7 days, then ask the user to enter the departure time again.

In option 4, you will ask the user to enter the number of people who will stay (**n**), then traversing *myRoomArray*, you will display the free rooms satisfying the user's request.

- If there are free rooms whose capacity (the number of people specified in the room type) is equal to n, display all of these rooms.

- If the capacity of free rooms is not sufficient to accomodate n people such that allocating 1 room is enough for the customer, then offer some combinations of the rooms so that the total offered capacity becomes sufficient to accomodate n people.

  Ex. If n is 8, you might offer *2 family rooms*.

  If n is 2 and there are no free *2 person rooms*, you might offer 2 *1 person room*s.

  If n is 3, you might offer 1 *2 person room* and 1 *1 person room.*

  (A room cannot be shared by people of different reservations.)

After the free rooms are displayed, the user will be required to enter his/her choice of room(s), s/he will be asked to enter his/her name. The reservedRoom field of the customer corresponding to that name will be set using the room information, updating the availableOrNot field so that it now becomes "not".

In option 5, you will delete the customer, who previously made a reservation. The customer will be removed from *myCustomerArray* and his/her *reservedRoom* will be marked as "available".

Whenever the availability of a room changes, the file containing the room information will be updated accordingly.

There are 20 rooms in the hotel.

In option 6, you will display how many rooms are in use. If the room information of an example hotel is given as in Table 1, you will display the message as:

1 person room: 4/8 are in use
2 person room: 3/7 are in use
family room: 2/5 are in use

Room type	1 person	2 person	family
Number of rooms	8	7	5
Number of rooms which are not available	4	3	2

Table 1. Room information of an example hotel

In option 7, you will display the number of people staying at the hotel.

Note:

➢ **You will demonstrate your assignment on 4.1.2017. The time and location of demonstrations will be announced.**

➢ **This assignment is worth 200 points.**