Optional Lab

Deadline in lab on Nov 24

You are to write a program that keeps track of the occupancy of rooms in a hotel. Your hotel has 5 rooms, numbered 1 through 5, inclusive. Here are the tasks you must be able to handle:

1) Checking a customer in.
2) Checking a customer out.
3) Printing a list of all customers
4) Searching for a customer

Unlike previous programs, your program will not take ANY input from the user. Rather, your program will simply open the file "hotellog.txt", read in and process the input from this file and produce the output file "summary.txt".

Input file format

The following lines will contain the commands, one command per line. Each input command will have one of the four possible formats:

```
CHECKIN MM/DD/YYYY Last_Name Number_of_People Number of Days
PRINTOCCUPANCY
SEARCH Last_Name
```

For the first command, `Last_Name` represents the last name of the party to be checked in and `Number_of_People` represents the number of people in the party. For commands two and four, `Last_Name` represents the party being checked out and searched for, respectively.

You may assume that each last name is 19 characters or less and that the number of people in each party is an integer in between 1 and 10, inclusive. Also, the input files are in chronological order in terms of checkin date. Also, when a person checks out, the room cannot be checked in since it requires one day to clean. For example, if a person checks on Nov 4, 2022 from Room 1, it is available for check in on Nov 5, 2022. Rooms are checked in from rooms with lowest numbers. If Room1 and 5 are available, then Room 1 is always checked in first.

Output file format

To execute checking in a party, you must output one of the two following types of messages:

```
Last_Name was checked into room 1 on November 1, 2022.
```

Where `Last_Name` is the name of the party being checked in, and `Number` is the room number of where the party was checked into.

Or, if the hotel is full, print out a message with the following format:

```
Sorry, no room was available for Last_Name.
```

To execute checking out a party, you must output one of the two following types of messages:

```
Last_Name was checked out of room 1 on November 3, 2022.
```

Where `Last_Name` is the name of the party checking out, and `Number` is the room number of where the party had stayed.

When executing the command `PRINTOCCUPANCY`, simply print out a list with each occupied room, its occupant, and the number of people in their party at that specific time. Here's an example:

```
Name       Room Number    Number of Occupants
Smith      0              3
Walsh      1              10
Davis      2              7
```

You may adjust the format slighly, but you must print out one row of headers, followed by each occupant on a row by itself. Uniform spacing is not required, but the output should be readable. Also, you should only print out a row for each occupied room.

Finally, for a search, you should produce one of two types of output. If the occupant is found, output a statement of the form:

```
Last_Name is currently staying in room 1.
```

If no occupant is found in the hotel with the last name `Last_Name`, then print out the following message:

```
Sorry, there is no occupant named Last_Name.
```

Sample Input File

```
CHECKIN 11/01/2022 Smith 3 3
CHECKIN 11/02/2022 Owens 5 10
CHECKIN 11/03/2022 Davis 7 1
SEARCH Owens
SEARCH Ryoo
CHECKIN 11/04/2022 Lauren 7 1
PRINTOCCUPANCY
```

Sample Output

```
Smith was checked into room 1 on November 1, 2022.
```

```
Owens was checked into room 2 on November 2, 2022.
Davis was checked into room 3 on November 3, 2022.
Owens is currently staying in room 2.
Sorry, there is no occupant named Ryoo.
Smith was checked out of room 1 on November 4, 2022.
Lauren was checked into room 4 on November 7, 2022.
Name      Room Number    Number of Occupants
Owens     2              10
Davis     3              7
Lauren    4              7
```

Grading Criteria

Your program will be graded upon the following criteria:

1) Your correctness
2) Your programming style and use of white space. (Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you won't get the full mark.)
3) How points are awarded based on tiers
    a) Midterm grade <= 50%: midterm grade to 51%
    b) Midterm grade between 50-60%: 2 points added to midterm grade
    c) Midterm grade between 60-80%: 1 point added to midterm grade
    d) Midterm grade between 80%+: 1 point added to midterm grade and bonus points in one lab or assignment

Other additional submission requirements are:

1. The file should be named: lab9.c
2. You can include auxiliary files.
3. Coding style and modular function designs will be checked as well.
4. Run command will be ./a.out hotellog.txt.

Submission for this optional lab.

- Present the code and run it in the lab
- There will be no corner case testing