|  |  |  |
| --- | --- | --- |
| **LAB101 Assignment** | **Type:** | **Short Assignment** |
| **Code:** | **C.S.P0019** |
| **LOC:** | **49** |
| **Slot(s):** | **1** |

**Title**

Management Hotel Room management.

**Background Context**

The structure stores an array that keeps track of a room's availability and an integer that keeps track of how many more rooms are available.

**Program Specifications**

You are to write a program that keeps track of the occupancy of rooms in a hotel. Your hotel has 10 rooms, numbered 0 through 9, 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

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".

***Function details:***

* Function 1: The input file, "hotellog.txt" will have a single integer, n, on the first line, indicating the number of commands you will have to process. The following n lines will contain the commands, one command per line. Each input command will have one of the four possible formats:

*CHECKIN Last\_Name Number\_of\_People*

*CHECKOUT Last\_Name*

*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.

* Function 2: To execute checking in a party, you must output one of the two following types of messages:

*Last\_Name was checked into room Number.*

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 Number.*

Where Last\_Name is the name of the party checking out, and Number is the room number of where the party had stayed.

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.*

When executing the command PRINTOCCUPANCY, simply print out a list with each occupied room, its occupant, and the number of people in their party. 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 Number.*

If the occupant is not found, simply output

*Sorry, there is no occupant named Last\_Name.*

In between the output for each command executed, leave a blank line.

***Expectation of User interface:***

