CS 1050 Homework Assignment 3

MIZZOU ACCOMODATIONS RESERVATION SYSTEM 2.0 (MARS 2.0)

Spring-2019

Due: Friday, April 26th at 5:00 PM (No Extension). Submission system shuts off AT 5pm.

****START EARLY! ****

Directions

Complete the following homework assignment using the description given in each section. You may want to review the solution to Homework #2 that has been provided on Canvas, as this assignment is more or less an extension of Homework #2.

Submission information:

Submit the assignment by the due date using the similar submission command as used for labs
Please include your lab section in the file name.

Filename must be: *sectionletter-hw3.c* (Include your respective lab section)

e.g. k-hw3.c

```
$ submit <class> <assignment> <filename>
$ submit CS1050 HW3 k-hw3.c
```

Purpose:

- Practice doing file processing.
- Practice using command-line arguments.
- Practice using dynamic memory allocation with malloc (if doing the BONUS)

Description

Your client from HW1 and HW2 (Interpedia) continues to be impressed with your work. They have come back to ask that you build a solution that is entirely data-driven. This time, their analysts will provide files that contain all of the hotel, room, and cost data. Your system will need to load that data at runtime and use it for their hotel booking solution.

Your program should accept 3 file names as arguments on the command-line. If you do not know how to use command-line arguments, please refer to the lecture/Panopto material on this subject.

The first file will specify the names of the hotels. The first line of this file will contain a number representing how many hotel names will be in the rest of the file. After this number, there will be a hotel name on each line until all of the names have been given.

Similarly, the second file will specify the names of the room types. The first line of this file will contain a number representing how many room type names will be in the rest of the file. After this number, there will be a room type name on each line until all of the names have been given.

Finally, the third file will have lines that represent a hotel, and on each line will be a space-separated list of integers that represent the prices of each room within that hotel. The first line will correspond to the first hotel named in the hotel names file, and so on. The first integer on each line will correspond to the first room type specified in the room type file, and so on. Altogether, this file represents the cost matrix which is similar to homework #2.

Your program must print a menu for the user to select what he/she needs to do with the application. The application only displays hotel brands, room types, and prices in various ways. Note that it does not actually allow the user to select a hotel/room combination to book a room.

A file hw3Template.c is attached on Canvas which will guide you to how to start the program and what global variables you need to use. Comments and prototypes have been given to help you in the process. You will also find 3 sample data files on Canvas corresponding to the hotel names file, the room type names file, and the cost matrix file. **Note that when your program is graded, the TAs may use files that are different (longer, shorter, different data, etc.) than these sample files**!

Functions to implement similar to HW2

1. const char * const GetHotelName(int hotel)
2. const char * const GetRoomName(int room)
3. void PrintHotelOptions()
4. void PrintRoomOptions()
5. void PrintHotelRoomsLessThan(int iLimit)
6. int GetMainMenuChoice()
7. int GetRoomPrice(int hotel, int room)
8. void PrintHotelRoomsByHotel(int hotel)
9. void PrintHotelsByRoom(int room);
10. int main(int argc, char *argv[]);
    Yep, you are gonna need a main. The main should be prototyped as shown, so that you can retrieve command-line arguments.

Functions new to HW3

1. int LoadHotelNames(char * sFileName);
   Loads hotel names from the data file into memory.
2. int LoadRoomNames(char * sFileName);
   Loads room type names from the data file into memory.
3. int LoadCostMatrix(char * sFileName);
   Loads the cost matrix from the data file into memory.
4. void PrintCostMatrix();
   Prints out the cost matrix to the screen.
5. void PrintHotelNames();
   Prints out the hotel name list to the screen.
6. void PrintRoomNames();
   Prints out the room type list to the screen.

Important Notes

- You must implement all 16 of the above-described functions! Failure to create and use these functions in your program will result in a **50% reduction in points**!
- You may write additional functions if you wish.

Bonus

For bonus points, instead of assuming a maximum number of hotels, room types, and cost matrix size, you should dynamically allocate these variables using malloc(). That is, you will need to read the number of hotel names before allocating the memory to store the hotel names. Similarly, you will need to have read the number of room types as well in order to allocate storage for the room types and cost matrix. The sample template file (hw3Template.c) should help you understand how your global variables might change depending on whether you are doing the bonus or not.

Sample Output

Your output should look virtually identical to the following, when using the sample data files:

```
JimR@JimRArea51:~/$ compile hw3.c
JimR@JimRArea51:~/$ ./a.out
./a.out hotels_file rooms_file cost_file
JimR@JimRArea51:~/$ ./a.out hotels.dat rooms.dat
./a.out hotels_file rooms_file cost_file
JimR@JimRArea51:~/$ ./a.out hotels.dat rooms.dat cost.dat
***** Welcome to the Mizzou Accomodations Reservation System 2.0 (MARS 2.0) *****
Data Loading.....
Hotel Names:
------------
Marriott
BestWestern
Motel6


Room Names:
-----------
Standard
Queen
King
Suite


Cost Matrix:
------------
Marriott,Standard = 100
Marriott,Queen = 200
Marriott,King = 300
Marriott,Suite = 400
BestWestern,Standard = 110
BestWestern,Queen = 120
BestWestern,King = 130
BestWestern,Suite = 140
Motel6,Standard = 210
Motel6,Queen = 220
Motel6,King = 250
Motel6,Suite = 525


Data Loaded!!




*** Main Menu ***
1. Display by hotel
2. Display by room type
3. Display by price limit
4. Exit program

Please make a selection: 1
1. Marriott
2. BestWestern
3. Motel6
Select a hotel: 3

*** Rooms at Motel6:
Motel6 - Standard ($210)
Motel6 - Queen ($220)
```

```
Motel6 - King ($250)
Motel6 - Suite ($525)


*** Main Menu ***
1. Display by hotel
2. Display by room type
3. Display by price limit
4. Exit program

Please make a selection: 2
1. Standard
2. Queen
3. King
4. Suite
Select a room: 2

*** Hotels with room Queen:
Marriott - Queen ($200)
BestWestern - Queen ($120)
Motel6 - Queen ($220)


*** Main Menu ***
1. Display by hotel
2. Display by room type
3. Display by price limit
4. Exit program

Please make a selection: 3
Enter the maximum price in dollars you will pay: 125

*** Rooms less than $125:
Marriott - Standard ($100)
BestWestern - Standard ($110)
BestWestern - Queen ($120)


*** Main Menu ***
1. Display by hotel
2. Display by room type
3. Display by price limit
4. Exit program

Please make a selection: 4
***** Thank you for using MARS 2.0! *****
```

Guidelines for Grading Homework-3
60 Points Possible (+5 bonus)

General

Assignment will not be given any credit (0/60) if your program does not compile or fail to produce any valid output. Submit the homework assignment using the submission command specified. **ADDITIONALLY, if your program does not compile you will receive NO credit!!!!**

10 points – Proper prototypes, headers, comments, and general coding style with good indentation.
15 points –All of the 10 functions from HW2 still function and work correctly in this version. No data is hard-coded in the source code (as it was in HW2).
5 points – Hotel names are loaded from the data file into g_hotelNames properly.
5 points – Room type names are loaded from the data file into g_roomNames properly.
10 points – The cost matrix is loaded from the data file into g_costMatrix properly.
5 points – Hotel names are printed to the screen properly using PrintHotelNames().
5 points –Room type names are printed to the screen properly using PrintRoomNames().
5 points – The cost matrix is printed to the screen properly using PrintCostMatrix().
5 points – (BONUS) - g_hotelNames, g_roomNames, and g_costMatrix are defined as shown in the #ifdef BONUS section. These variables are dynamically allocated using malloc() as needed, and use no more memory than necessary based on the data in the data files read at runtime.