/\*

\* Class: CMSC140\_30522

\* Instructor: Charles Naegeli

\* Project<3>

\* Description: You are required to write a program that calculates the occupancy rate and the total hotel income for one night and displays this information as well as some other information described below.

The program starts by asking the location where this hotel chain is located and the number of floors in the hotel. The number of floors may not exceed 5. The User then enters the total number of rooms for each floor. The program then asks specifically the number of occupied rooms for each room type on this floor. The total number of rooms on each floor may not exceed 30 and the program should check that the total number of occupied rooms on each floor does not exceed the total of rooms on that floor.

\* Due Date: 5/3/21

\* I pledge that I have completed the programming assignment independently.

I have not copied the code from a student or any source.

I have not given my code to any student.

Print your Name here: Chris Tark

\* Pseudocode or Algorithm for the program:

(be sure to indent items with control structure)

(need to match flow chart submitted in documentation)

1. using previous knowledge from chapters

2. control structure

3.do while loops

4.while statements

5.for while loops

\*/

#include <iostream>

#include <iomanip>

#include <string>

#include <cmath>

using namespace std;

int main() {

//string for location

string hotelLocation;

// rates for hotel

int singleRate = 60, doubleRate = 75, kingRate = 100, suiteRate = 150;

//variable

int sum = 0;

int rooms[30], single[10], doubleRoom[10], king[10], suite[10], floors, i, notOccupied, rateOcc = 0, min;

double occupied = 0, hotelIncome = 0, totalRoom = 0;

//output

cout << "=========================================================================" << endl;

cout << " BlueMont Hotel" << endl;

cout << "=========================================================================" << endl;

//asking for location

cout << "Enter the location of this hotel chain: ";

cin >> hotelLocation;

//asking for floors

cout << "Enter total number of floors of the hotel: ";

cin >> floors;

//loops for # of floors

while (floors <= 0 || floors >= 6) {

cout << "Number of floors should be between 1 and 5 !! Please try again\n";

cout << "\nEnter total number of floors of the hotel: ";

cin >> floors;

}

//loops for # of rooms

for (i = 1; i <= floors; i++) {

cout << "\nEnter total number of rooms in the " << i << "th Floor: ";

cin >> rooms[i];

//if user input is wrong

while (rooms[i] <= 0 || rooms[i] >= 31) {

cout << "Number of rooms should be between 1 and 30 !! Please try again\n";

cout << "\nEnter total number of rooms in the " << i << "th Floor: ";

cin >> rooms[i];

}

// occupied rooms

cout << "How many SINGLE rooms are occupied in the " << i << "th floor: ";

cin >> single[i];

cout << "How many DOUBLE rooms are occupied in the " << i << "th floor: ";

cin >> doubleRoom[i];

cout << "How many KING rooms are occupied in the " << i << "th floor: ";

cin >> king[i];

cout << "How many SUITE rooms are occupied in the " << i << "th floor: ";

cin >> suite[i];

sum = single[i] + doubleRoom[i] + king[i] + suite[i];

// when occupied rooms are greater than room

while (sum > rooms[i]) {

cout << "\nTotal number of occupied rooms exceeds the total number of rooms on this floor. Please try again!!" << endl;

cout << "Enter total number of rooms in the " << i << "th floor: ";

cin >> rooms[i];

cout << "How many SINGLE rooms are occupied in the " << i << "th floor: ";

cin >> single[i];

cout << "How many DOUBLE rooms are occupied in the " << i << "th floor: ";

cin >> doubleRoom[i];

cout << "How many KING rooms are occupied in the " << i << "th floor: ";

cin >> king[i];

cout << "How many SUITE rooms are occupied in the " << i << "th floor: ";

cin >> suite[i];

sum = single[i] + doubleRoom[i] + king[i] + suite[i];

}

}

// display results from user input(hotel cost for the user)

cout << "===========================================================================" << endl;

cout << " BlueMont Hotel located in " << hotelLocation << "\n\n";

cout << "===========================================================================" << endl;

cout << " TODAY'S ROOM RATES(US$/night)\n\n";

cout << setw(15) << "Single Room" << setw(15) << "Double Room" << setw(15) << "King Room" << setw(15) << "Suite Room\n\n";

cout << setw(15) << singleRate << setw(15) << doubleRate << setw(15) << kingRate << setw(15) << suiteRate << "\n\n";

cout << "\n=========================================================================" << endl;

//int singleRate = 60, doubleRate = 75, kingRate = 100, suiteRate = 150;

//int rooms[30], single[10], doubleRoom[10], king[10], suite[10], floors, i, notOccupied, rateOcc = 0, min;

//double occupied = 0, hotelIncome = 0, totalRoom = 0;

//

//math

for (i = 1; i <= floors; i++) {

occupied += single[i] + doubleRoom[i] + king[i] + suite[i];

hotelIncome += (single[i] \* singleRate) + (doubleRoom[i] \* doubleRate) + (king[i] \* kingRate) + (suite[i] \* suiteRate);

totalRoom += rooms[i];

}

// more math for occupancy

notOccupied = totalRoom - occupied;

rateOcc = (occupied / totalRoom) \* 100;

//display total fee for the user

cout << setw(30) << "Hotel Income: " << setw(30) << hotelIncome << endl;

cout << setw(30) << "Total # of rooms: " << setw(30) << totalRoom << endl;

cout << setw(30) << "Total # of Occupied rooms: " << setw(30) << occupied << endl;

cout << setw(30) << "Total # of Unoccupied rooms: " << setw(30) << notOccupied << endl;

cout << setw(30) << "Occupany rate: " << setw(30) << rateOcc << endl;

int room = 1;

min = rooms[i];

// min amount of rooms

for (i = 2; i <= floors; i++) {

while (rooms[i] > min) {

min = i;

room = i;

}

}

// stating least # of floor

cout << "\n" << room << "th Floor with " << rooms[room] << " rooms, has the least # of rooms.";

if (rateOcc < 60) {

cout << "\nNeed to improve Hotel occupancy rate!!\n";

}

cout << "\n\nThank you for testing my program!!";

cout << "\nPROGRAMEER : Chris Tark";

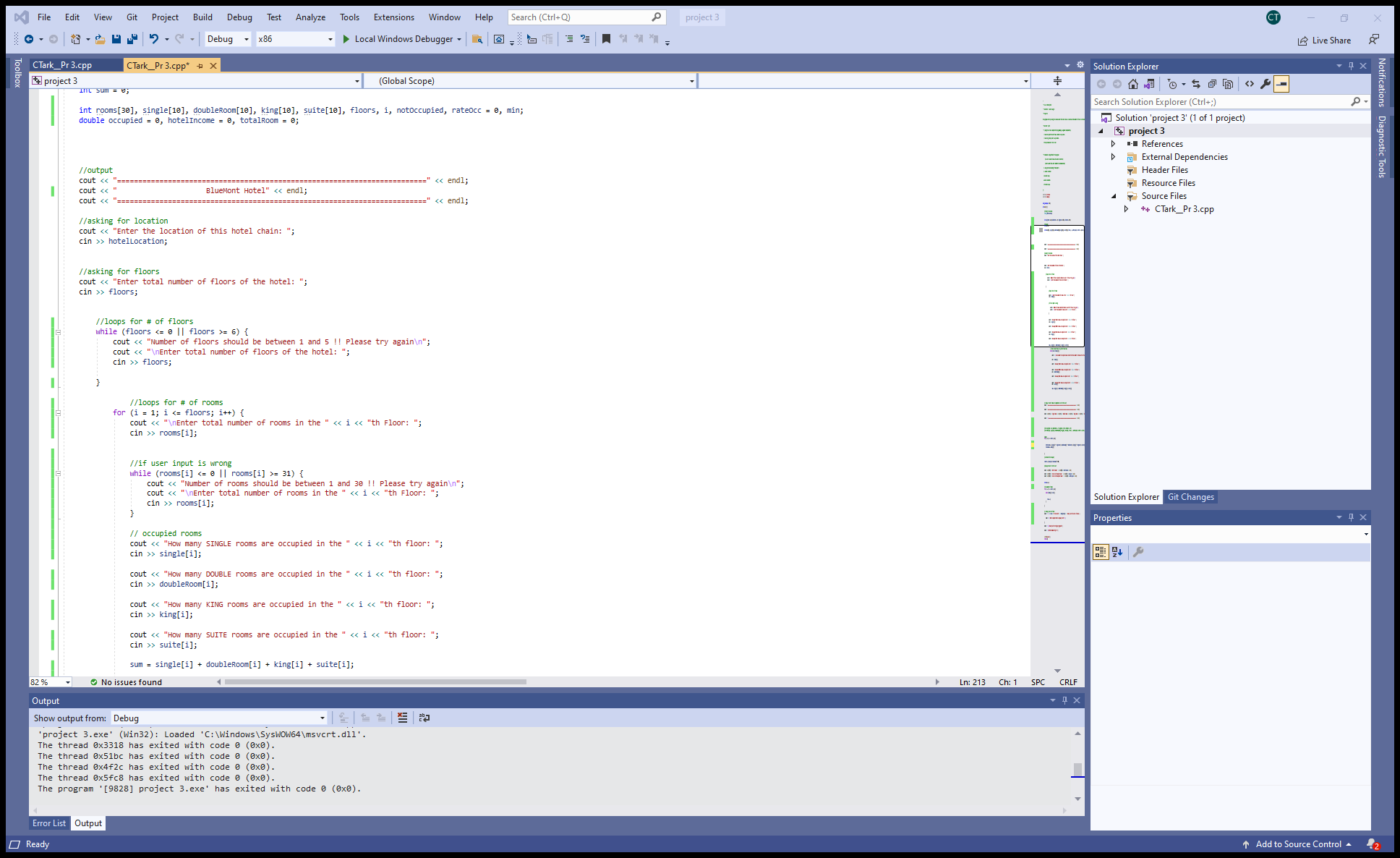
cout << "\nCMSC140 Common Project 3";

cout << "\nDue Date: 5/3\n\n";

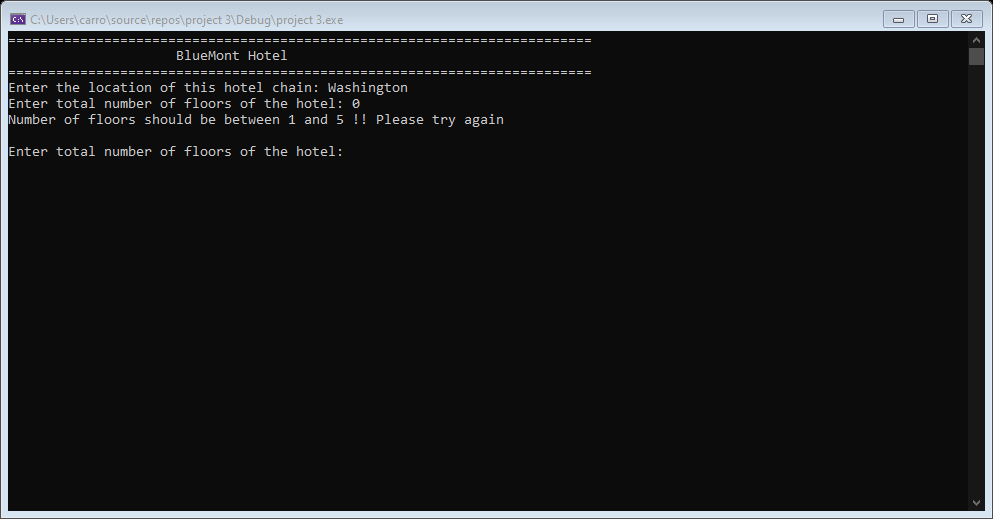
system("pause");

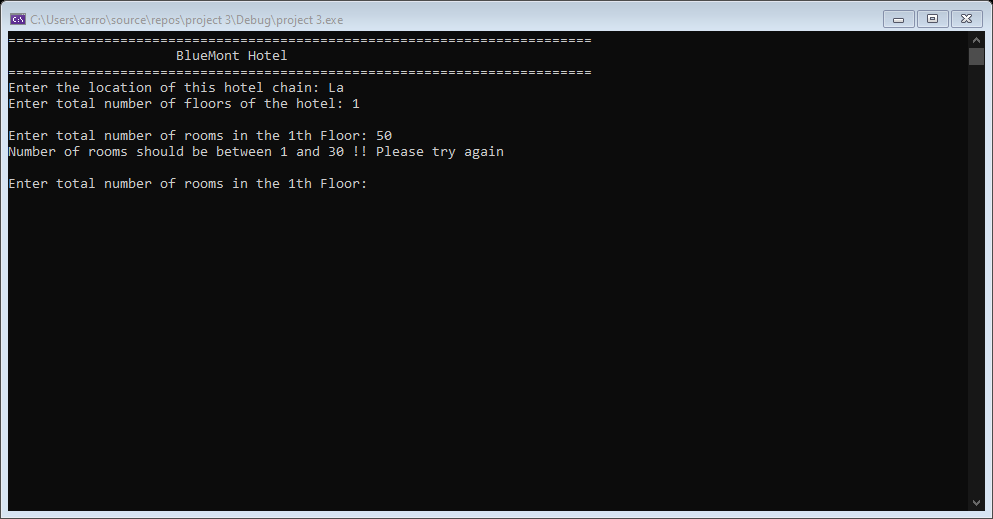
return 0;

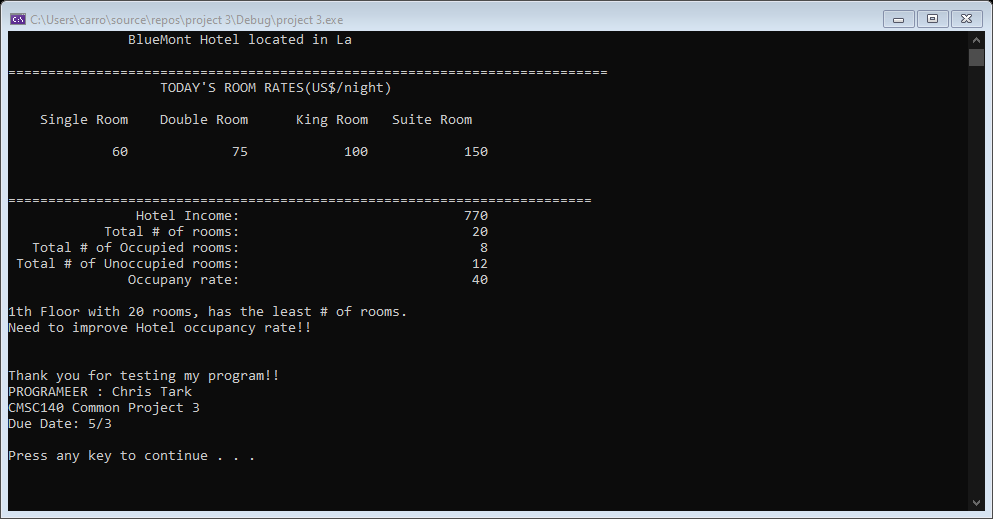
}

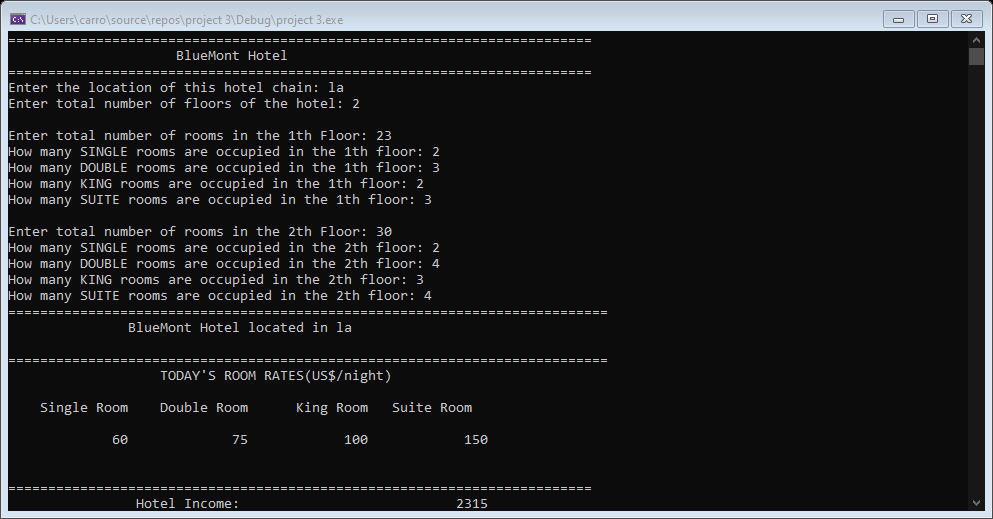


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | loc: Washington  #of floors: 0 | Washington  #of floors: 0 | Number of floors should be between 1 and 5. Please try again!! | Number of floors should be between 1 and 5. Please try again!! | yes |
| 2 | LA 1 floor and 50 rooms | LA 1 floor and 50 rooms | # of rooms should be between 1 and 30 | # of rooms should be between 1 and 30 | yes |
| 3 | Enter total number of rooms in the 1th Floor: 20  How many SINGLE rooms are occupied in the 1th floor: 2  How many DOUBLE rooms are occupied in the 1th floor: 2  How many KING rooms are occupied in the 1th floor: 2  How many SUITE rooms are occupied in the 1th floor: 2 | Enter total number of rooms in the 1th Floor: 20  How many SINGLE rooms are occupied in the 1th floor: 2  How many DOUBLE rooms are occupied in the 1th floor: 2  How many KING rooms are occupied in the 1th floor: 2  How many SUITE rooms are occupied in the 1th floor: 2 | Gives us the total for everything | Number of rooms should be between 1 and 30 !! Please try again | yes |
| 4 | Enter 2 floors | Enter 2 floors | Gives us the option to put info about the 2 floors and there room | Enter total number of rooms in the 1th Floor: 23  How many SINGLE rooms are occupied in the 1th floor: 2  How many DOUBLE rooms are occupied in the 1th floor: 3  How many KING rooms are occupied in the 1th floor: 2  How many SUITE rooms are occupied in the 1th floor: 3  Enter total number of rooms in the 2th Floor: 30  How many SINGLE rooms are occupied in the 2th floor: 2  How many DOUBLE rooms are occupied in the 2th floor: 4  How many KING rooms are occupied in the 2th floor: 3  How many SUITE rooms are occupied in the 2th floor: 4 | Yes |









The picture is in chronological order from top to bottom.

The last box in the flow chart is the end.

Diagram

Description automatically generated