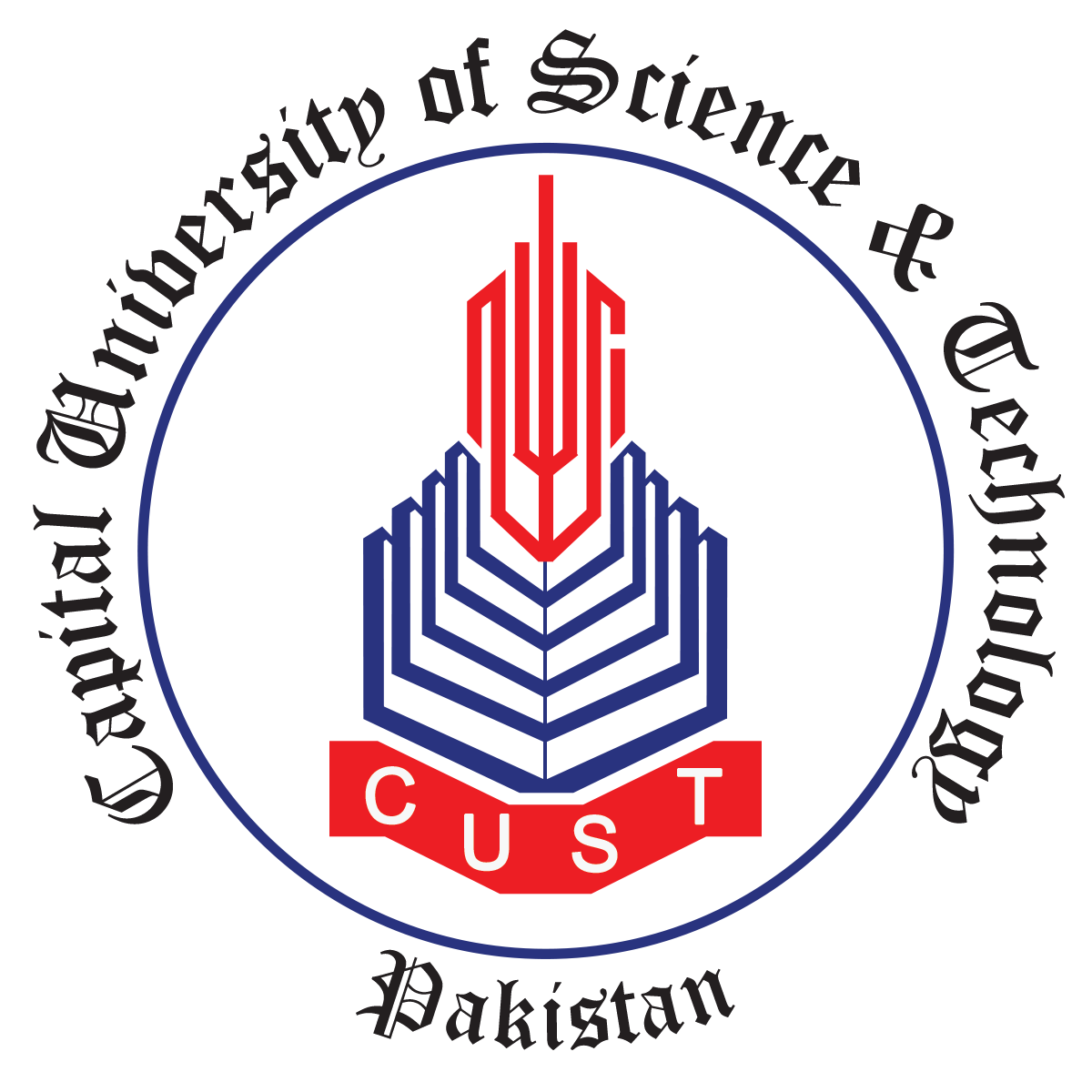
****

**Name: Muhammad Ammar Janjuha**

**Registration No: BCS213104**

**Section : 02**

**Subject: Data Structure**

**Project : Hotel Management System**

**Submitted to : Sir Syed Saqib**

**Date : 13-June-2023**

**Department of Computer Science**

**Abstract:**

The Hotel Management System is a program that allows hotel staff to manage room allocations, guest information, and billing. The program is implemented using a doubly linked list data structure to efficiently store and organize room records.

The system provides various functionalities such as allocating different types of rooms (family, VIP, other), displaying allocated rooms, calculating bills based on room rates, searching for specific rooms, updating room information, and deleting room records.

Upon initialization, the user specifies the number of rooms available in the hotel. The program then presents a menu of options for the user to select from. Users can allocate rooms to guests, display the details of allocated rooms, calculate bills based on the number of allocated rooms and room rates, search for rooms based on room numbers, update guest information for specific rooms, and delete room records.

The program ensures data integrity by validating user inputs, such as room numbers, and provides appropriate error messages if required. It also keeps track of the number of available rooms and the total number of allocated rooms, allowing users to monitor the occupancy status of the hotel.

Overall, the Hotel Management System provides a user-friendly interface for efficiently managing room allocations, guest information, and billing in a hotel setting using a doubly linked list data structure

The Hotel Room Management System provides a basic framework for managing hotel rooms and keeping track of their occupancy status in real-time.

**Working Process of Code:**

The code is an implementation of a hotel management system using a doubly linked list

The code is a hotel management system implemented in C++. It allows the user to allocate rooms to guests, calculate bills, and manage room availability. Here's an overview of the code:

1. The code defines a **node** struct to represent each guest, including information such as room number, name, phone number, CNIC, and age. Each node also has pointers to the previous and next nodes.
2. The **Hotel** class is defined to manage the hotel operations.
3. The **Hotel** class has private member variables to keep track of various counts, such as the number of rooms, available family rooms, available VIP rooms, etc. It also has variables to store user inputs like the number of rooms to be rented and the number of days of stay.
4. The constructor of the **Hotel** class initializes the member variables to their default values.
5. The **family\_room** function is responsible for handling the rental process for family rooms. It takes user inputs for the number of rooms and days of stay. It then checks if the requested number of rooms is available and proceeds accordingly.
6. The **vip\_room** function is similar to **family\_room** but handles the rental process for VIP rooms.
7. The **first\_allot**, **after\_allot**, **before\_allot**, and **last\_allot** functions are used to allocate a room to a guest based on their preference (first, after, before, or last). These functions create a new **node** for the guest, take input for guest details, and place the node in the appropriate position in the linked list.
8. The **working** function is not included in the code snippet you provided. It is assumed to be another function that continues the hotel management operations after a room has been allocated.

Overall, the code provides a basic implementation of a hotel management system using linked lists. However, without the **working** function, it's not clear how the rest of the hotel operations are handled or how the linked list is utilized beyond room allocation.

**CODE IN C++**

#include <iostream>

#include <string>

using namespace std;

struct node

{

// Room Number

int room\_no;

// Name of Person

string name;

// Phone number of Person

long int Phone\_no;

// CNIC number of Person

long int CNIC;

// Age of Person

int age;

// Pointer for previous and next node

struct node\* prev, \* next;

};

class Hotel

{

private:

int counter, num, b;

int familyroom, viproom, totalroom;

struct node\* r, \* start;

int choice, room, day;

int rupees;

int bill;

int allocation;

int y;

public:

Hotel()

{

counter = 100;

start = NULL;

familyroom = 20;

viproom = 30;

totalroom = 50;

rupees = bill = 0;

allocation = day = b = room = num = choice = 0;

y = 0;

}

// Class for Family room

void family\_room()

{

cout << "\n\t\*\*\*WELCOME TO 1st floor for family Rooms...!\*\*\*\t" << endl;

cout << "\n\tRent of the room is 15000 Rs/day\t" << endl;

cout << "\n\n\thow many rooms do you want for rent :\t";

cin >> room;

cout << "\n\thow many days do you want to stay :\t";

cin >> day;

if (room <= 20 && familyroom >= room)

{

cout << "\n\tFamily Rooms range from 1 - 20\t" << endl;

}

else

{

cout << "\n\tSORRY ROOMS ARE CURRENTLY UNAVAILABLE" << endl;

cout << "\tAvailable Family Rooms: " << familyroom << endl;

family\_room();

}

if (room <= 20)

{

a:

day = day \* 15000;

bill = day \* room;

cout << "\n\tyour bill is\t" << bill << endl;

cout << "\n\tROOM CAN BE ALLOCATED" << endl;

cout << "\n\tpress 1 for first allot" << endl;

cout << "\n\tpress 2 for after allot" << endl;

cout << "\n\tpress 3 for before allot" << endl;

cout << "\n\tpress 4 for last allot" << endl;

cout << "\nENTER YOUR CHOICE! :\t";

cin >> allocation;

if (allocation == 1)

{

while (room != 0)

{

y = first\_allot();

if (y == 1) {

familyroom--;

counter--;

room--;

}

}

cout << "available Family Rooms: " << familyroom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 2)

{

while (room != 0)

{

y = after\_allot();

if (y == 1)

{

familyroom--;

counter--;

room--;

}

}

cout << "available Family Rooms: " << familyroom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 3)

{

while (room != 0)

{

y = before\_allot();

if (y == 1)

{

familyroom--;

counter--;

room--;

}

}

cout << "available Family Rooms: " << familyroom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 4)

{

while (room != 0)

{

y = last\_allot();

if (y == 1)

{

familyroom--;

counter--;

room--;

}

}

cout << "available Family Rooms: " << familyroom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else

{

cout << "enter a valid option" << endl;

goto a;

}

}

}

// Class for VIP room

void vip\_room()

{

cout << "\n\t\*\*\*WELCOME TO 2nd floor for vip Rooms...!\*\*\*\t" << endl;

cout << "\n\tRent of the room is 25000 Rs/day\t" << endl;

cout << "\n\n\thow many rooms do you want for rent :\t";

cin >> room;

cout << "\n\thow many days do you want to stay :\t";

cin >> day;

if (room <= 30 && viproom >= room)

{

cout << "\n\tVIP Rooms range from 21 - 50\t" << endl;

}

else

{

cout << "\n\tSORRY ROOMS ARE CURRENTLY UNAVAILABLE" << endl;

cout << "\tAvailable VIP Rooms: " << viproom << endl;

vip\_room();

}

if (room <= 30)

{

b:

day = day \* 25000;

bill = day \* room;

cout << "\n\tyour bill is\t" << bill << endl;

cout << "\n\tROOM CAN BE ALLOCATED" << endl;

cout << "\n\tpress 1 for first allot" << endl;

cout << "\n\tpress 2 for after allot" << endl;

cout << "\n\tpress 3 for before allot" << endl;

cout << "\n\tpress 4 for last allot" << endl;

cout << "\nENTER YOUR CHOICE! :\t";

cin >> allocation;

if (allocation == 1)

{

while (room != 0)

{

y = first\_allot();

if (y == 1)

{

viproom--;

counter--;

room--;

}

}

cout << "available VIP Rooms: " << viproom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 2)

{

while (room != 0)

{

y = after\_allot();

if (y == 1)

{

viproom--;

counter--;

room--;

}

}

cout << "available VIP Rooms: " << viproom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 3)

{

while (room != 0)

{

y = before\_allot();

if (y == 1)

{

viproom--;

counter--;

room--;

}

}

cout << "available VIP Rooms: " << viproom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else if (allocation == 4)

{

while (room != 0)

{

y = last\_allot();

if (y == 1)

{

viproom--;

counter--;

room--;

}

}

cout << "available VIP Rooms: " << viproom << endl;

cout << "Total available Rooms: " << counter << endl;

working();

}

else

{

cout << "enter a valid option" << endl;

goto b;

}

}

}

// Function for First Allot and Allots a room to the guest at the beginning of the linked list.

int first\_allot()

{

node\* p, \* tmp;

int check = 1;

p = new node[sizeof(node)];

p->prev = NULL;

p->next = NULL;

if (start == NULL)

{

start = r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

else

{

tmp = start;

while (tmp->next != NULL)

{

tmp = tmp->next;

}

tmp->next = p;

p->prev = tmp;

r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

}

// Function for After Allot and Allots a room to the guest after a specified room number in the linked list.

int after\_allot()

{

node\* p, \* tmp;

int loc, check = 1;

p = new node[sizeof(node)];

p->prev = NULL;

p->next = NULL;

cout << "\n\tAFTER ALLOT FUNCTION\n" << endl;

cout << "\tEnter the location after which you want to allot the room: ";

cin >> loc;

if (start == NULL)

{

start = r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

else

{

tmp = start;

while (tmp->room\_no != loc)

{

tmp = tmp->next;

if (tmp == NULL)

{

cout << "Room number does not exist!" << endl;

return 0;

}

}

p->next = tmp->next;

p->prev = tmp;

if (tmp->next != NULL)

tmp->next->prev = p;

tmp->next = p;

r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

}

// Function for Before Allot and Allots a room to the guest before a specified room number in the linked list.

int before\_allot()

{

node\* p, \* tmp;

int loc, check = 1;

p = new node[sizeof(node)];

p->prev = NULL;

p->next = NULL;

cout << "\n\tBEFORE ALLOT FUNCTION\n" << endl;

cout << "\tEnter the location before which you want to allot the room: ";

cin >> loc;

if (start == NULL)

{

start = r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

else

{

tmp = start;

while (tmp->room\_no != loc)

{

tmp = tmp->next;

if (tmp == NULL)

{

cout << "Room number does not exist!" << endl;

return 0;

}

}

p->next = tmp;

p->prev = tmp->prev;

if (tmp->prev != NULL)

tmp->prev->next = p;

else

start = p;

tmp->prev = p;

r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

}

// Function for Last Allot and Allots a room to the guest at the end of the linked list.

int last\_allot()

{

node\* p, \* tmp;

int check = 1;

p = new node[sizeof(node)];

p->prev = NULL;

p->next = NULL;

if (start == NULL)

{

start = r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

else

{

tmp = start;

while (tmp->next != NULL)

{

tmp = tmp->next;

}

tmp->next = p;

p->prev = tmp;

r = p;

cout << "Please enter your name: ";

cin.ignore();

getline(cin, p->name);

cout << "Please enter your room number: ";

cin >> p->room\_no;

cout << "Please enter your phone number: ";

cin >> p->Phone\_no;

cout << "Please enter your CNIC number: ";

cin >> p->CNIC;

cout << "Please enter your age: ";

cin >> p->age;

cout << "Congratulations! Room has been allotted to you." << endl;

check = 1;

return check;

}

}

// Function to Display the List

void display()

{

node\* tmp;

int count = 0;

tmp = start;

cout << "\n\*\*\*List of Allotted Rooms\*\*\*" << endl;

cout << "-----------------------------------------" << endl;

while (tmp != NULL)

{

cout << "Room Number: " << tmp->room\_no << endl;

cout << "Name: " << tmp->name << endl;

cout << "Phone Number: " << tmp->Phone\_no << endl;

cout << "CNIC Number: " << tmp->CNIC << endl;

cout << "Age: " << tmp->age << endl;

cout << "-----------------------------------------" << endl;

tmp = tmp->next;

count++;

}

cout << "Total Allotted Rooms: " << count << endl;

}

// Function to Calculate Total Revenue

void total\_revenue()

{

int revenue;

revenue = (20 - familyroom) \* 15000 + (30 - viproom) \* 25000;

cout << "Total Revenue: " << revenue << " Rs" << endl;

}

// Main Function

void working()

{

cout << "\t\*\*\*\*\* WELCOM TO MARIAN HOTEL\*\*\*\*\* \n\t" << endl;

cout << "\t\*\*\* 1st floor for Family Rooms...!\*\*\* \n" << endl;

cout << "\t\*\*\* 2nd floor for VIP Rooms...!\*\*\* \n" << endl;

cout << "\n\t1. Allot Family Room" << endl;

cout << "\t2. Allot VIP Room" << endl;

cout << "\t3. Display Allotted Rooms" << endl;

cout << "\t4. Calculate Total Revenue" << endl;

cout << "\t5. Exit" << endl;

cout << "\nEnter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

family\_room();

break;

case 2:

vip\_room();

break;

case 3:

display();

working();

break;

case 4:

total\_revenue();

working();

break;

case 5:

break;

default:

cout << "Invalid choice! Please enter a valid option." << endl;

working();

break;

}

}

};

int main()

{

Hotel h;

h.working();

return 0;

}