Problem Statement :

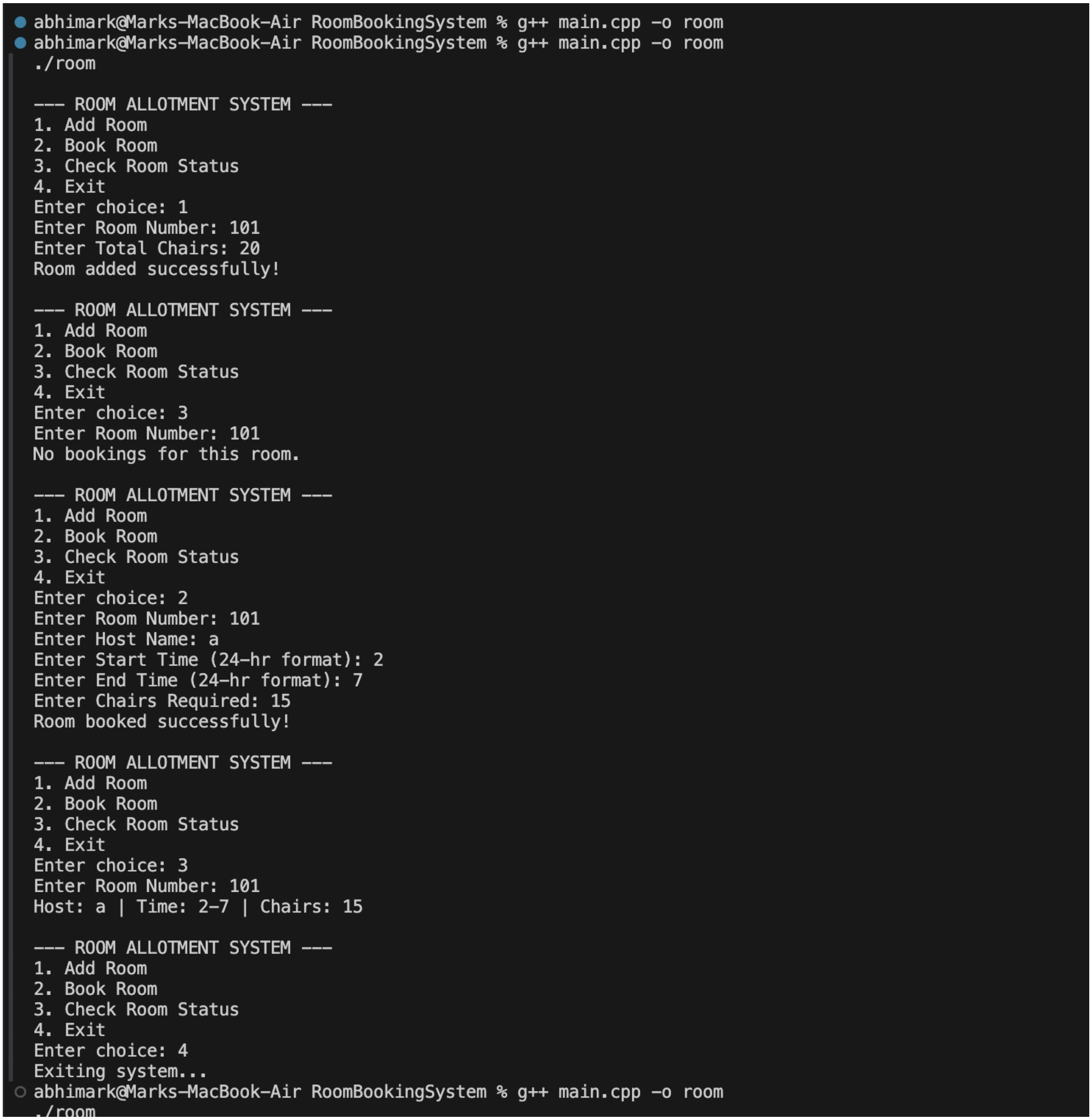
Room Allotment and booking system using advance DSA Concepts and Multiple parameters in C++

Code :

```
#include <iostream>
#include <vector>
#include <map>
#include <string>
using namespace std;

struct Booking {
    string hostName;
    int startTime;
    int endTime;
    int chairsRequired;
};

class Room {
public:
    int roomNumber;
    int totalChairs;
    vector<Booking> bookings;

    Room() {
        roomNumber = 0;
        totalChairs = 0;
    }

    Room(int num, int chairs) {
        roomNumber = num;
        totalChairs = chairs;
    }

    bool isAvailable(int start, int end) {
        for (auto &b : bookings) {
            if (!(end <= b.startTime || start >= b.endTime)) {
                return false;
            }
        }
        return true;
    }

    bool addBooking(Booking b) {
        if (b.chairsRequired > totalChairs)
            return false;

        if (!isAvailable(b.startTime, b.endTime))
            return false;

        bookings.push_back(b);
        return true;
    }

    void displayBookings() {
        if (bookings.empty()) {
            cout << "No bookings for this room.\n";
            return;
        }

        for (auto &b : bookings) {
            cout << "Host: " << b.hostName
                 << " | Time: " << b.startTime << "-" << b.endTime
                 << " | Chairs: " << b.chairsRequired << endl;
        }
    }
};

class RoomBookingSystem {
    map<int, Room> rooms;

public:
    void addRoom() {
        int roomNo, chairs;
        cout << "Enter Room Number: ";
        cin >> roomNo;
        cout << "Enter Total Chairs: ";
    void bookRoom() {
        int roomNo;
        rooms.emplace(roomNo, Room(roomNo, chairs));
        cout << "Room added successfully!\n";
    }

    void bookRoom() {
        int roomNo;
        Booking b;

        cout << "Enter Room Number: ";
        cin >> roomNo;

        if (rooms.find(roomNo) == rooms.end()) {
            cout << "Room not found!\n";
            return;
        }

        cout << "Enter Host Name: ";
        cin.ignore();
        getline(cin, b.hostName);

        cout << "Enter Start Time (24-hr format): ";
        cin >> b.startTime;

        cout << "Enter End Time (24-hr format): ";
        cin >> b.endTime;

        cout << "Enter Chairs Required: ";
        cin >> b.chairsRequired;

        if (rooms[roomNo].addBooking(b))
            cout << "Room booked successfully!\n";
        else
            cout << "Booking failed (Time conflict or insufficient chairs).\n"
    }

    void checkRoomStatus() {
        int roomNo;
        cout << "Enter Room Number: ";
        cin >> roomNo;

        if (rooms.find(roomNo) == rooms.end()) {
            cout << "Room not found!\n";
            return;
```

```cpp
        if (rooms.find(roomNo) == rooms.end()) {
            cout << "Room not found!\n";
            return;
        }

        rooms[roomNo].displayBookings();
    }

    void menu() {
        int choice;
        do {
            cout << "\n--- ROOM ALLOTMENT SYSTEM ---\n";
            cout << "1. Add Room\n";
            cout << "2. Book Room\n";
            cout << "3. Check Room Status\n";
            cout << "4. Exit\n";
            cout << "Enter choice: ";
            cin >> choice;

            switch (choice) {
                case 1: addRoom(); break;
                case 2: bookRoom(); break;
                case 3: checkRoomStatus(); break;
                case 4: cout << "Exiting system...\n"; break;
                default: cout << "Invalid choice!\n";
            }
        } while (choice != 4);
    }
};

int main() {
    RoomBookingSystem system;
    system.menu();
    return 0;
}
```

Output:



```
● abhimark@Marks-MacBook-Air RoomBookingSystem % g++ main.cpp -o room
● abhimark@Marks-MacBook-Air RoomBookingSystem % g++ main.cpp -o room
  ./room

  --- ROOM ALLOTMENT SYSTEM ---
  1. Add Room
  2. Book Room
  3. Check Room Status
  4. Exit
  Enter choice: 1
  Enter Room Number: 101
  Enter Total Chairs: 20
  Room added successfully!

  --- ROOM ALLOTMENT SYSTEM ---
  1. Add Room
  2. Book Room
  3. Check Room Status
  4. Exit
  Enter choice: 3
  Enter Room Number: 101
  No bookings for this room.

  --- ROOM ALLOTMENT SYSTEM ---
  1. Add Room
  2. Book Room
  3. Check Room Status
  4. Exit
  Enter choice: 2
  Enter Room Number: 101
  Enter Host Name: a
  Enter Start Time (24-hr format): 2
  Enter End Time (24-hr format): 7
  Enter Chairs Required: 15
  Room booked successfully!

  --- ROOM ALLOTMENT SYSTEM ---
  1. Add Room
  2. Book Room
  3. Check Room Status
  4. Exit
  Enter choice: 3
  Enter Room Number: 101
  Host: a | Time: 2-7 | Chairs: 15

  --- ROOM ALLOTMENT SYSTEM ---
  1. Add Room
  2. Book Room
  3. Check Room Status
  4. Exit
  Enter choice: 4
  Exiting system...
○ abhimark@Marks-MacBook-Air RoomBookingSystem % g++ main.cpp -o room
  ./room
```