Scope

Writing an algorithm, for the Hotel Management software, so as to decide if a room can be booked as per the various conditions.

Business Scenario:

A hotel has number of rooms. It rents out those rooms. For better and easier management and for maximizing the profit, they set some conditions for the rooms. For example, if guest wants to stay in the room 101, the condition says that he should stay at least for 3 nights. If guest is booking that room for 2 days, he is then not allowed to book it.

There are more such conditions, we call them *rentability*.

Rentability

Arrival From – Departure To

This indicates that guest can check in and check out in this date range (both dates inclusive).

Example – Hotel wants to rent out its rooms from 1st Nov to 14th Nov and 21st Nov to 2nd Dec 2016. The remaining period (15th to 20th Nov) is closed and should not be allowed to be rented/booked by the customer.

Check-in Day and Check-out Day

This condition means that guest can check-in / check-out only on specific day(s) of the week.

Example –Guest should be able to check in only on Friday and check out only on Monday or Friday.

Minimum / Maximum Length of Stay

This condition means that a guest must stay for minimum number of nights (minimum length of stay) and guest cannot stay for more than maximum length of stay.

Example – Guest should stay a for minimum of 2 days and maximum stay period allowed is 10 days

The combination of all the three above mentioned conditions make up **Rentability.**

Rentability Example

Let's see few examples to understand the concept more clearly.

Look at the table below:

Arrival From	Departure To	Check-in Day	Check-out Day	Minimum Length of Stay	Maximum Length of Stay
1-11-2016	14-11-2016	Monday	Friday, Monday	3 nights	7 nights

This data can be interpreted in following way:

- The guest can check-in from 1 Nov to 14 Nov.
- The guest can check-out from 1 Nov to 14 Nov.
- The guest can check-in only on Mondays.
- The guest can check-out only on Fridays and Mondays.
- The guest has to stay for minimum 3 nights and he cannot stay more than 7 nights.

You can have multiple such configurations (criteria) for same room. We will see it in below section where we are explaining what all possible test cases should be covered.

Test Cases

Case 1

We will start with a very basic test case:

Arrival From	Departure To	Check-in Day	Check-out Day
1-11-2016	14-11-2016	Friday, Monday	Monday, Friday

- Guest can check-in between 01 Nov 2016 till 14 Nov 2016
- Guest can check-out between 01 Nov 2016 till 14 Nov 2016
- Guest can check-in only on Friday and Monday
- Guest can check-out only on Monday and Friday

So, for example, the guest can check-in on 07 Nov 2016 which is Monday and check-out on 11 Nov 2016 which is Friday.

Case 2

Arrival From	Departure To	Check-in Day	Check-out Day	Minimum Length of Stay	Maximum Length of Stay
1-11-2016	14-11-2016	Friday, Monday	Monday, Friday	3 nights	ts

As compared to Case 1, we have added two more criteria in this case:

- Guest must stay for minimum 3 nights
- Guest cannot stay more than 7 nights

So, for example, the guest can check-in on 04 Nov 2016 (Friday) and check-out on 11 Nov 2016 (Friday) with stay of 7 nights.

But, guest cannot check-out on 14 Nov 2016 (Monday) as it will result into stay of 10 nights which is not permissible (max limit is 7 nights).

Note: The values in Minimum or Maximum stay can be NULL as well which means that there is no limitation of minimum/Maximum stay for renting the rooms.

Case 3

This is the 'real' case. Please read and understand it carefully.

Arrival From	Departure To	Check-in Day	Check-out Day	Minimum Length of Stay	Maximum Length of Stay
1-11-2016	14-11-2016	Friday, Monday	Friday, Monday	3 nights	14 nights
21-11-2016	02-12-2016	Monday	Friday	3 nights	14 nights
03-12-2016	02-01-2017	Monday, Friday	Monday, Friday	7 nights	21 nights

You can see multiple criteria. **Note: The values for Arrival From and Departure To will never overlap.**

Check-in and Check-out lie in different records and the records are continuous

Look at second and third records. You will observe that they are continuous i.e. one record is from 21 Nov to 02 Dec and next one is from 03 Dec to 02 Jan. There is no gap between 'departure to' of second record and 'arrival from' of third record.

We can say that – guest can check-in on 28 Nov 2016 (Monday) *(from second record)* and check-out on 05 Dec 2016 (Monday) *(from third record)* with duration of 7 nights. 7 nights duration is allowed as per first record.

But, if guest checks-in on 28 Nov 2016 (Monday) and checks-out on 16 Dec 2016 (Friday) i.e. with duration of 18 nights, then it is not allowed because as per the first record where check-in is happening, maximum stay is of 14 nights.

Check-in and Check-out lie in different records and the records are discontinuous

Look at first and second records. You will observe that they are discontinuous i.e. one record is from 01 Nov to 14 Nov and next one is from 21Nov to 02 Dec. There is a gap between 'departure to' of first record and 'arrival from' of second record.

In this case – guest cannot check-in on 14 Nov 2016 (Friday) *(from first record)* and check-out on 25 Nov 2016 (Friday) *(from second record)* with duration of 11 nights. This is only because the rentability is not continuous i.e. first and second records are not continuous.

What you have to do?

We have a ROOM table:

ROOM
room_id (number)
room_code (varchar2)

The column room_id is just a primary key, and the column room_code is code given to a room.

Design the database table for storing rentability per room. It should be possible to have multiple rentability record for a given room.

Once the database table is ready, you will need to write a Java program having a function as below:

public booleanisBookable(Long room_id, Date arrival_date, Date departure_date)

This function should look into the rentability data for the given room_id and on the basis of rentability, it should decide whether the room can be booked for given arrival_date and departure_date. It should return true if room is bookable, else false.

To make this assignment simple, instead of using any database and JDBC, you can store the rentability in a file and read that file or you can also hardcode the data inside your program using some data structure.

The program should support all the test cases mentioned above.