

# Hotel Reservations

---

## Background

Imagine you own a hotel with a certain number of identical rooms - let's call this number the **size** of your hotel. Your clients would like to make reservations for rooms: They e.g. call to check if they can make a reservation for one room from a certain start date **start** until a certain end date **end**. For each booking request you will have to check if there is a room available and either confirm the booking or reject it.

## Goal

Your task is to write a small interactive application to check the availability of rooms. In addition: Please implement the test cases described below to automatically test your solution.

- Feel free to also include necessary libraries as you would normally do.
- Please submit a zip archive containing all your source code artifacts in a way that it can run at least on a standard terminal as a small text-based client (no UI necessary).
- Please note that, along with the functionality of your code, we will also evaluate its quality and style.

## Details

Assume you have a hotel with 3 rooms:

	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5
Room 1						
Room 2						
Room 3						

Let's represent bookings as tuple: **(startDay, endDay)**, e.g. **(0, 2)**. Guests always book full days: the booking **(0, 0)** blocks one room only on day 0. The booking **(1, 3)** will block a room for days 1/2/3.

After bookings **(0, 0)**, **(0, 2)**, **(2, 4)** and **(2, 2)** (which we all accept) our hotel might for example look like this.

	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5
Room 1	X		X	X	X	
Room 2	X	X	X			
Room 3			X			

We would now have to decline another booking **(2, 3)**.

Notes:

- Days are represented as the number of days from a certain date, e.g. day 0, day 1, day 2, ...; We will limit bookings to 1 year, that is 365 days.
- All rooms are identical and may be assumed to be numbered. The size of your hotel is `size <= 1000`.
- Guest do not change the room during their stay, but always stay within the room they moved in initially.
- If a booking request arrives and we can accept it, we accept it directly. We do not wait for later requests (e.g. to maximize the utilization of our rooms).
- If there are multiple rooms available for booking, the one with the smallest room number is chosen first.

In case of additional questions: Feel free to make necessary assumptions. If you do so, please state and shortly explain your assumptions in your code.

## Test Cases

Please inspect the following test cases and implement them accordingly. You may also consider additional test cases, if you think they are meaningful.

1a/1b: Requests outside our planning period are declined (Size=1)

	StartDate	EndDate	Result: Accept / Decline
Booking 1	-4	2	Decline

and

	StartDate	EndDate	Result: Accept / Decline
Booking 1	200	400	Decline

2: Requests are accepted (Size=3)

	StartDate	EndDate	Result: Accept / Decline
Booking 1	0	5	Accept
Booking 2	7	13	Accept
Booking 3	3	9	Accept
Booking 4	5	7	Accept
Booking 5	6	6	Accept
Booking 6	0	4	Accept

3: Requests are declined (Size=3)

	StartDate	EndDate	Result: Accept / Decline
Booking 1	1	3	Accept
Booking 2	2	5	Accept
Booking 3	1	9	Accept

	<b>StartDate</b>	<b>EndDate</b>	<b>Result: Accept / Decline</b>
Booking 4	0	15	Decline

4: Requests can be accepted after a decline (Size=3)

	<b>StartDate</b>	<b>EndDate</b>	<b>Result: Accept / Decline</b>
Booking 1	1	3	Accept
Booking 2	0	15	Accept
Booking 3	1	9	Accept
Booking 4	2	5	Decline
Booking 5	4	9	Accept

5: Complex Requests (Size=2)

	<b>StartDate</b>	<b>EndDate</b>	<b>Result: Accept / Decline</b>
Booking 1	1	3	Accept
Booking 2	0	4	Accept
Booking 3	2	3	Decline
Booking 4	5	5	Accept
Booking 5	4	10	Decline
Booking 6	10	10	Accept
Booking 7	6	7	Accept
Booking 8	8	10	Accept
Booking 9	8	9	Accept