PROBLEM 7: HOTEL RESERVATIONS

For this exercise you must write a program that helps the owner of a small bed and breakfast hotel manage their reservations. This small hotel has six rooms (conveniently numbered 1 through 6) and offers complementary breakfast for each guest. The owner has asked all guests to confirm which days they will take breakfast, so they can plan accordingly.

Your program will be given a series of reservation requests. Each request will include the check-in date, the number of nights the guests will be staying, the number of guests, the room requested (1–6), and which mornings the guests would like to have breakfast. Your program must then output for each request whether it was accepted or rejected. Requests must be automatically accepted, unless rejected for one or more of the following reasons: (1) the room requested is not available for one or more nights included in the request, (2) the number of guests is above the maximum for the room (rooms 1–4 can accomodate a maximum of 2 guests each while rooms 5 and 6 can accomodate 4 guests each), (3) the request is invalid in some way, such as not staying any nights, a non-existant room was requested, or breakfast was requested after they have checked out. Then, for each day covered by the requests, your program must output how many guests will be having breakfast that morning and which rooms will be occupied that night (in numerical order and separated by commas). If no rooms will be occupied on a given night, your program must output "n/a" in place of the list of rooms. If your program did not receive any valid requests, it must output "No valid requests."

Your program will receive input as soon as it launches. Each line of input will constitute a reservation request, with each part of the request separated by spaces. The first part of each line will be the check-in date, in the format `YYYY-MM-DD`. The second part will be the number of nights the guest(s) would like to stay. The third part will be the number of guests in the party. The fourth part will be the room being requested. The final section of input will be the mornings on which the party would like to have breakfast, with 1 indicating the first morning after check-in, 2 indicating the second morning after check-in, etc. The breakfast requests will be separated by hyphens (-). The requests will end when you receive a hyphen (-) on a line by itself. You may safely assume that you will all given dates will be valid and that all numbers given will be integers less than 32 bytes.

*Your program's output must match the examples given below **exactly**. Your program must not prompt for input. Note carefully the spelling, capitalization, punctuation, and spacing of the output. The input that will be given to your program is highlighted in **red**.*

EXAMPLE RUN 1

```
2023-06-27 5 2 1 1-2-3-4
-
Request 1: Accepted
2023-06-27 -> Breakfast: 0 & Rooms: 1
2023-06-28 -> Breakfast: 2 & Rooms: 1
2023-06-29 -> Breakfast: 2 & Rooms: 1
2023-06-30 -> Breakfast: 2 & Rooms: 1
2023-07-01 -> Breakfast: 2 & Rooms: 1
2023-07-02 -> Breakfast: 0 & Rooms: n/a
```

EXAMPLE RUN 2

```
2023-08-09 7 4 5 1-2-3-4-5-6-7
2023-08-11 2 2 2 1-2
-
Request 1: Accepted
Request 2: Accepted
2023-08-09 -> Breakfast: 0 & Rooms: 5
2023-08-10 -> Breakfast: 4 & Rooms: 5
2023-08-11 -> Breakfast: 4 & Rooms: 2,5
2023-08-12 -> Breakfast: 6 & Rooms: 2,5
2023-08-13 -> Breakfast: 6 & Rooms: 5
2023-08-14 -> Breakfast: 4 & Rooms: 5
2023-08-15 -> Breakfast: 4 & Rooms: 5
2023-08-16 -> Breakfast: 4 & Rooms: n/a
```

EXAMPLE RUN 3

```
2024-02-01 4 2 3 1-2-3-4
2024-02-02 4 2 3 1-3-4
2024-02-03 3 2 4 1-3
2024-02-03 3 4 1 1-2-3
2024-02-03 3 4 6 1-2-3
-
Request 1: Accepted
Request 2: Rejected
Request 3: Accepted
Request 4: Rejected
Request 5: Accepted
2024-02-01 -> Breakfast: 0 & Rooms: 3
2024-02-02 -> Breakfast: 2 & Rooms: 3
2024-02-03 -> Breakfast: 2 & Rooms: 3,4,6
2024-02-04 -> Breakfast: 8 & Rooms: 3,4,6
2024-02-05 -> Breakfast: 6 & Rooms: 4,6
2024-02-06 -> Breakfast: 6 & Rooms: n/a
```