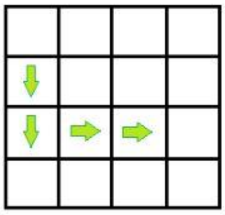
Problem 2 – Parking System

The parking lot in front of SoftUni is one of the busiest in the country, and it's a common cause for conflicts between the doorkeeper Bai Tzetzo and the students. The SoftUni team wants to proactively resolve all conflicts, so an automated parking system should be implemented. They are organizing a competition – Parkoniada – and the author of the best parking system will win a romantic dinner with RoYaL. That's **exactly** what you've been dreaming of, so you decide to join in.

The parking lot is a **rectangular** matrix where the **first** column is **always** free and all other cells are parking spots. A car can enter from any cell of the first column and then decides to go to a specific spot. If that spot is **not** free, the car searches for the **closest** free spot on the **same** row. If **all** the cells on that specific row are used, the car cannot park and leaves. If **two** free cells are located at the **same** distance from the **initial** parking spot, the cell which is **closer** to the entrance is preferred. A car can **pass** through a used parking spot.

Your task is to calculate the distance travelled by each car to its parking spot.



A car enters the parking at row 1. It wants to go to cell 2, 2 so it moves through **exactly four** cells to reach its parking spot.

Input

- On the first line of input, you are given the integers **R** and **C**, defining the dimensions of the parking lot.
- On the next several lines, you are given the integers **Z, X, Y** where **Z** is the entry row and **X, Y** are the coordinates of the desired parking spot.
- The input stops with the command '**stop**'. All integers are separated by a **single** space.

Output

- For each car, print the distance travelled to the desired spot or the first free spot.
- If a car cannot park on its desired row, print the message '**Row {row number} full**'

Constraints

- 2 ≤ R,C ≤ 10000
- Z, X, Y are inside the dimensions of the matrix. Y is never on the first column.
- There are no more than 1000 input lines
- Allowed time/space: 0.1s (C#) 0.25s (Java)/16MB

Input	Output
4 4 1 2 2 2 2 2 2 2 2 3 2 2 stop	4 2 4 Row 2 full

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.