Mirrors and Lasers

We would like to see how you solve the following short, contained project.

- You have **one** week to complete this project.
- Expect to spend between 2-4 hours to produce a working solution.
- Unless you have been asked to code in any language, solve the task in C++.
- Keep your code clean, organized and well written!
- Include a short report or readme file which explains your design decisions and instructions on how to compile and run your code.
- Please submit your solution as an attachment by email. Please keep this assignment confidential.
- Be sure to be able to talk about your solution, its complexity, and scalability.

Overview

You will be solving an algorithmic problem. We present you a computation challenge which you should not only solve optimally but you also should be able to analyze its **memory** usage and **runtime** performance.

Sensor System Implementation

Safe Ltd. is a company that manufactures high-quality safes. Its latest invention is an optical closure mechanism that uses a laser beam passing through a rectangular grid with several mirrors.

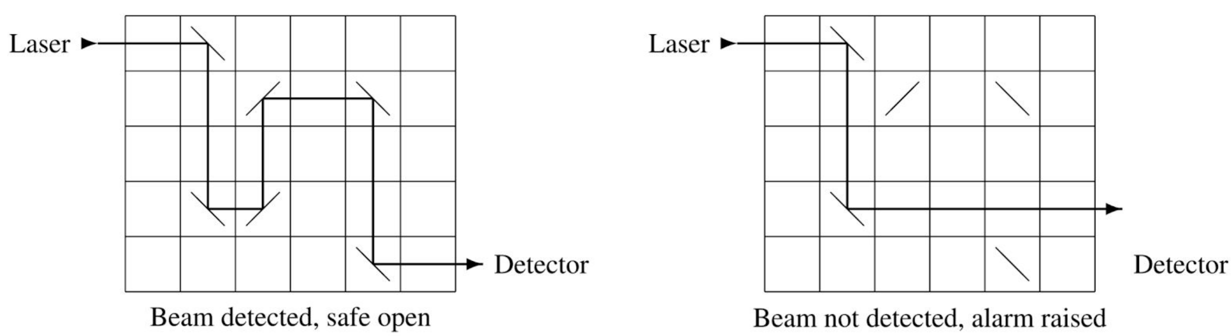

Figure 1: A sample mirror-laser-safe setup.

When the laser is activated, a beam enters the top row of the grid horizontally from the left. The beam is reflected by every mirror that it hits. Each mirror has a 45 degree diagonal orientation, either / or \. If the beam exits the bottom row of the grid horizontally to the right, it is detected

and the safe opens (see the left side of the figure above). Otherwise, the safe remains closed and an alarm is raised.
Each safe has a missing mirror, which prevents the laser beam from traveling successfully through the grid (see the right side of the figure above). The safe has a mechanism that enables the user to drop a single mirror into any empty grid cell. A legitimate user knows the correct position and orientation of the missing mirror (/ in row 4 column 3 above) and can thus open the safe. Without this knowledge, the user has to guess correctly, which can be difficult for safes with large grids.
Your job is to determine if particular safes are actually secure. A secure safe does not open right away without inserting a mirror, and there is at least one valid location and orientation for the missing mirror. There may indeed be multiple such locations and orientations.

Input

Each test case describes a single safe and starts with a line containing four integer numbers r, c, m, and n. The mechanisms grid has r rows and c columns.
Each of the next m lines contains two integer numbers ri and ci($1 \le ri \le r$ and $1 \le ci \le c$) specifying that there is a / mirror in row ri column ci . The following n lines specify the positions of the \ mirrors in the same way. The m + n positions of the mirrors are pairwise distinct.

Constraints:

- Category I (0-2 years of experience) : $1 \le r, c \le$ **1000** and $0 \le m, n \le$ **2000**.
- Category II (2+ years of experience) : $1 \le r, c \le$ **1000000** and $0 \le m, n \le$ **200000**).

Output

For each test case, display its case number followed by:

- 0 if the safe opens without inserting a mirror.
- *k r c* if the safe does not open without inserting a mirror, there are exactly k positions where inserting a mirror opens the safe, and (r, c) is the lexicographically smallest such row, column position. A position where both a / and a \ mirror open the safe counts just once.
- *impossible* if the safe cannot be opened with or without inserting a mirror.

Sample Input	Output for Sample Input

`5 6 1 4`
`2 3`
`1 2`
`2 5`
`4 2`
`5 5`
`100 100 0 2`
`1 77`
`100 77`
`100 100 0 0`
	`Case 1: 2 4 3`
	`Case 2: 0`
	`Case 3: impossible`

Figure 2: A sample input-output.