Problem I. iSpyWithMyLittleEye

Time limit: 3 seconds
Memory limit: 1024 mebibytes

This is an interactive problem.

You and your friend have come up with a guessing game.

The game takes place on an $N \times M$ grid. The rows are numbered from 1 to N from top to bottom, and the columns are numbered from 1 to M from left to right. We denote the cell in row i and column j by (i, j).

Before the game begins, you write an integer between 1 and 4, inclusive, on every cell of the grid. You may choose these integers however you wish.

Then, while you are not looking, your friend starts moving around the grid according to the following rules:

- Your friend starts at an arbitrary cell.
- At each step, your friend moves to a cell sharing a side with the current cell.
- Your friend visits **every cell exactly once**.

Your friend then writes the numbers written on the cells in the order they visit them. This produces a sequence X of length NM.

After finishing the route, your friend tells you the sequence X, but does not reveal the chosen route itself.

Your task is to determine which cell your friend visited last.

The game is repeated Q times. You must use the same assignment of numbers for all rounds, but your friend may choose a different route each time.

Interaction Protocol

Your program must interact with the interactor through standard input and standard output.

The interaction consists of two phases.

Phase 1: Assigning numbers

First, read two integers N and M — the number of rows and the number of columns, respectively.

Print N lines, each containing a string of length M. Each character must be one of `1`, `2`, `3`, or `4`. The j-th character of the i-th line represents the number assigned to cell (i, j).

After printing all N lines, you must flush the output buffer.

The assignment cannot be changed after this phase.

Phase 2: Guessing the last cell

Read an integer Q — the number of rounds.

For each of the Q rounds:

1. Read a string X of length NM, consisting only of the characters `1`, `2`, `3`, and `4`. This is the sequence of numbers observed along the route chosen by the interactor in the current round.
2. Print two integers r and c separated by a space on a single line — the row and column of the last visited cell, respectively.

The interactor sends all Q strings without waiting for your answers. You may print each answer immediately after reading the corresponding string, or read all Q strings before printing your answers. In either case, you must print the answers in the order of the rounds and flush the output buffer after printing all Q answers.

After completing each phase, you must flush the output buffer to ensure that the interactor receives your output. Failing to do so can result in an unexpected verdict. You can flush the output by using the following methods:

- In `C++`, call `fflush(stdout)` or `cout.flush()`.
- In `Java`, call `System.out.flush()`.
- In `Python`, call `sys.stdout.flush()`.

For other languages, you should refer to the official documentation for your language.

Constraints

- $4 \leq N, M \leq 200$
- $1 \leq Q \leq 200$

Example

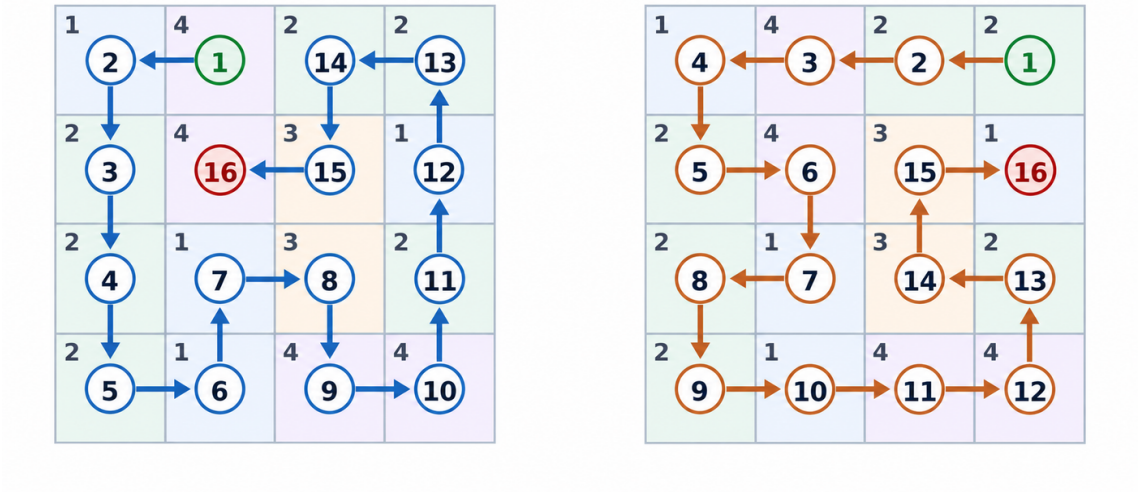
standard input	standard output
4 4 2 4122211344212234 2241241221442331	 1422 2431 2132 2144 2 2 2 4

Note

In the example, the program first assigns the following numbers to the 4×4 grid:

$$\begin{matrix} 1 & 4 & 2 & 2 \\ 2 & 4 & 3 & 1 \\ 2 & 1 & 3 & 2 \\ 2 & 1 & 4 & 4 \end{matrix}$$

The following figure illustrates the routes taken by your friend in the two rounds. In each cell, the small number in the top-left corner is the assigned number, and the number in the center indicates the order in which the cell is visited. The arrows indicate the direction of movement.



In the first round, your friend produces the sequence `4122211344212234`. The last visited cell is $(2, 2)$, so the program outputs `2 2`.

In the second round, your friend produces the sequence `2241241221442331`. The last visited cell is $(2, 4)$, so the program outputs `2 4`.