## The Goal

Your objective is to write a program capable of predicting the route Indy will take on his way down a tunnel. Indy is not in danger of getting trapped in this first mission.

## Rules

The tunnel consists of a patchwork of square rooms of different types.The rooms can be accessed and activated by computer using an ancient RS232 serial port (because Mayans aren't very technologically advanced, as is to be expected...).  
  
There is a total of 14 room types (6 base shapes extended to 14 through rotations).  
  
Upon entering a room, depending on the type of the room and Indy's entrance point (,, or ) he will either exit the room through a specific exit point, suffer a lethal collision or lose momentum and get stuck:

|  |  |  |  |
| --- | --- | --- | --- |
| Type 0 | This room type is not part of the tunnel per se as Indy cannot move across it. | | |
| Type 1 | The green arrows indicate Indy's possible paths through the room. | | |
| Type 2 | Type 3 | The room of type 3 is similar to the room of type 2 but rotated. | |
| Type 4 | Type 5 | A red arrow indicate a path that Indy cannot use to move across the room. | |
| Type 6 | Type 7 | Type 8 | Type 9 |
| Type 10 | Type 11 | Type 12 | Type 13 |

Indy is perpetually drawn downwards: he cannot leave a room through the top.  
  
At the start of the game, you are given the map of the tunnel in the form of a rectangular grid of rooms. Each room is represented by its type.  
  
For this first mission, you will familiarize yourself with the tunnel system, the rooms have all been arranged in such a way that Indy will have a safe continuous route between his starting point (top of the temple) and the exit area (bottom of the temple).  
  
Each game turn:

* You receive Indy's current position
* Then you specify what Indy's position will be next turn.
* Indy will then move from the current room to the next according to the shape of the current room.

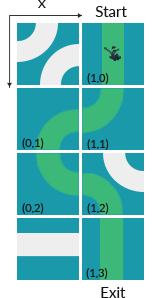
Victory Conditions

Indy reaches the exit

Lose Conditions

You assert an incorrect position for Indy

## Example



Indy starts in the room ,  
then, he falls down in and moves to .  
After that, he falls in and moves to .  
He finally reaches from where he can safely escape.

## Note

The tests provided are similar to the validation tests used to compute the final score but remain different.

## Game Input

The program must first read the initialization data from standard input. Then, within an infinite loop, read the data from the standard input related to the current position of Indy and provide to the standard output the expected data.

Initialization input

Line 1: 2 space separated integers W H specifying the width and height of the grid.

Next H lines: each line represents a line in the grid and contains W space separated integers T. T specifies the type of the room.

Last line: 1 integer EX specifying the coordinate along the X axis of the exit (this data is not useful for this first mission, it will be useful for the second level of this puzzle).

Input for one game turn

Line 1: XI YI POS

* (XI, YI) two integers to indicate Indy's current position on the grid.
* POS a single word indicating Indy's entrance point into the current room: if Indy enters from above, if Indy enters from the left and if Indy enters from the right.

Output for one game turn

A single line with 2 integers: X Y representing the (X, Y) coordinates of the room in which you believe Indy will be on the next turn.

Constraints

0 < W ≤ 20  
0 < H ≤ 20  
0 ≤ T ≤ 13  
0 ≤ EX < W  
0 ≤ XI, X < W  
0 ≤ YI, Y < H  
  
Response time for one game ≤ 150ms

Example

Initialization input

2 4 (W H)

4 3 (T T)

12 10 (T T)

11 5 (T T)

2 3 (T T)

1 (EX)

Grid with 2 columns, 4 lines.  
  


No output expected

Input for turn 1

1 0 TOP

(XI YI POS)

Indy enters room (1,0) from above  
  


Output for turn 1

1 1  
  
Determine that Indy will be in room (1,1) on the next turn

Input for turn 2

1 1 TOP

(XI YI POS)

Indy is now in (1,1)  
  


Output for turn 2

0 1  
  
Determine that Indy will be in room (0,1) on the next turn



And so on until Indy reaches the exit at (1, 3)...