Enrichment Center (glados)

After playing *Portal*, Giorgio decided to open his own *Enrichment Center* and be like GLaDOS in real life. In this kind of research facility, test subjects are locked into chambers and faced with several dangerous puzzles, which need to be solved to gain access to the exit (and to another even riskier chamber). Obviously, there are annoying laws about human rights that would interfere with a faithful implementation of this concept; but Giorgio already planned to get around that by building a very small (and cheap!) Enrichment Center designed for lab rats.

Of course, lab rats are not exceptionally smart so the puzzles need to be fairly straightforward. In the first test chamber, Giorgio is planning to arrange into a grid $H \times W$ some blocks of the following types:

- plain floor, represented by `'.'`;
- walls, represented by `'#'`;
- exits, represented by `'O'`;
- transparent glue, represented by `'@'`.

Figure 1: A prototypical chamber in the *Aperture Science*™ Enrichment Center.

Giorgio put the first lab rat in row-column coordinates $(R; C)$ into the grid, facing north. Quite surprisingly, he noticed that the rat is using one of the most ancient techniques for getting out of a maze: *stick to your right hand*.

In other words, the rat is following the wall that was on his right at the beginning, as if always sticking one hand on it.

Giorgio can't wait to know whether the rat will end up stuck in the glue, or if it will keep cycling forever or eventually reach an exit. Satisfy Giorgio's curiosity by calculating the rat's fate!

☞ Among the attachments of this task you may find a template file `glados.*` with a sample incomplete implementation.

Input

The first line contains the four integers H, W, R, C. Other H lines follow, each containing a string consisting of W characters among `'.'`, `'#'`, `'O'`, `'@'`.

Output

You need to write a single line with a single word:

- `stuck` if the rat will first reach a block with the glue,
- `free` if the rat will first reach a block with an exit,
- `cycling` if the rat will keep cycling forever.

Constraints

- $3 \leq H, W \leq 100$.
- $(R; C)$ vary from $(0; 0)$ — top left to $(H - 1, W - 1)$ — bottom right.
- All blocks on the perimeter are walls (so the rat cannot start on a border block).
- There is a wall on the right of the rat at the beginning (i.e., in position $(R, C + 1)$).

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (50 points) The answer is not `cycling`.

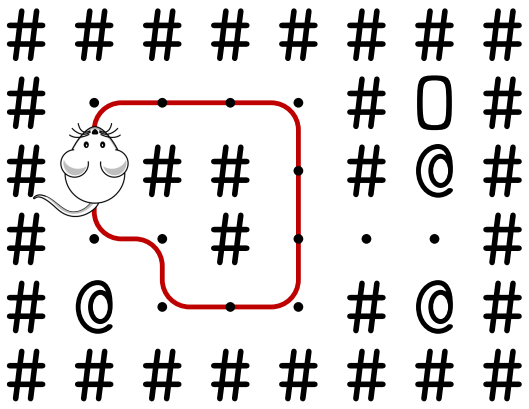
– **Subtask 3** (50 points) No additional limitations.

Examples

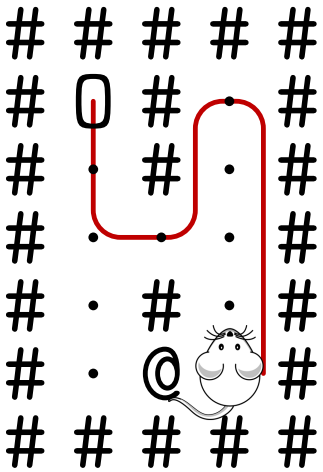
input.txt	output.txt
6 8 2 1 ######## #....#O# #.##.#@# #..#...# #@...#@# ########	cycling
7 5 5 3 ##### #O#.# #.#.# #...# #.#.# #.@.# #####	free
6 8 2 1 ######## #..@.#O# #.##.#@# #..#...# #@...#@# ########	stuck

Explanation

In the **first sample case**, the path followed by the rat is the following:



In the **second sample case**, the path followed by the rat is the following:



In the **third sample case**, the path followed by the rat is the following: