# I • A Mazing!

A maze consists of a collection of equal sized square cells, where any or all of the sides may be a *wall* or a door.   The maze may have no exit or multiple exits. Cells are typically arranged so that they may share sides with other cells as shown in the four sample mazes below:
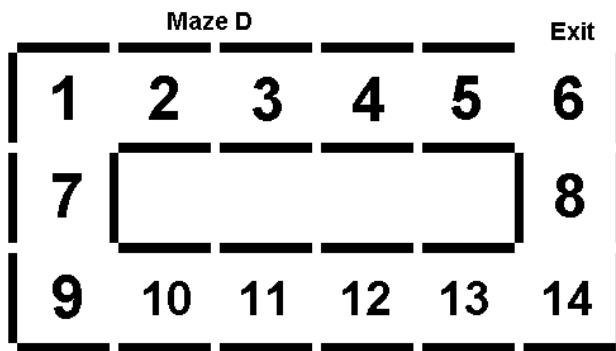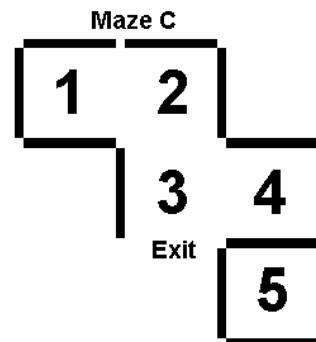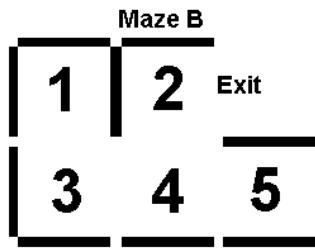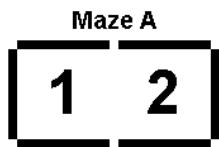
(The numbers in the cells above are for illustrative purposes only.)

Each side of a cell is labeled with a direction to allow navigation:
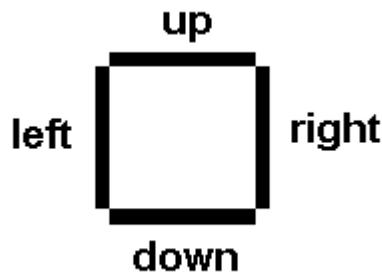
For each maze, the starting point is someplace in the maze (for example, any of the numbered cells in the samples above). In the samples above:

- Maze A has *no way out*.
- Maze B has an exit (solution) to the *right* of cell 2.
- Maze C has an exit *down* from cell 3, unless the starting point is cell 5, in which case there is *no way out*.

- Maze D has an exit *up* from cell 6.

For example, using *Maze D* above, if the starting point is cell 9, one possible set of directions to get to the exit would be: `right, right,` right`, right, right, up, up, up`.

For this problem, you will write a program that finds an exit to a maze. Your program must operate interactively. That is, your program will make a move by providing a direction (right, down, left or up), and the judging software will send back one of four responses:

1) **wall** – indicates that a wall is there and you cannot proceed in that direction
2) **ok** – indicates that there is door there and you may proceed in that direction to the neighboring cell.
3) **solved** – indicates that you have successfully found an exit to the maze
4) ***<EOF>*** - an end of file on the stream indicates that your program should exit. To be clear, this is **not** the literal string ***<EOF>***, rather it is the end of file indication that is provided by the input stream when there is no more data to read.

After each **solved** response, a new maze will be generated and your program should start over again. This process repeats until your program receives an ***<EOF>*** indication.

If your program determines there is no way out of the maze, you should send the precise string "`no way out`" instead of a direction. If there is in fact no way out, a new maze will be generated and your program should start over.

Your program will receive an ***<EOF>*** indication if any of the following occur:

1) When the judging program determines that your solution works. This can occur at any point during the run!
2) Your program sends "`no way out`", even though there is a way out.
3) Your program makes the same move (direction) from the same cell twice.

Input/Output

This is an interactive program. The input you receive is a function of the output you generate. All input and output strings must end in a new-line character. You should never send extra blank lines.

The first thing your program must do when it starts up is to send its first move (up, down, right or left), followed by a new-line character. It will then wait for a new-line terminated response on the standard input. The response will be one of **wall**, **ok**, **solved** or an ***<EOF>*** (end-of-file) indication. Your program will then make another move based on the response it received as discussed above. This process will repeat until your program receives an ***<EOF>*** indication.

Example (User output in **Bold**, Computer judge output in *Italics*). (This sample run has no relationship to the samples shown above).

Sample Run
down
wall
right
wall
left
wall
up
ok
right
ok
down
ok
down
wall
right
wall
left
wall
up
ok
right
solved
right
ok
up
ok
up
ok
up
wall
right
solved
right
<EOF>