

Homework 2 – Stack

Romeo & Juliet

TA: Sine (z970485888@gmail.com)

Deadline: 11:59pm, Oct. 21, 2020

1. Objective

Romeo and Juliet want to meet each other, but Count Paris sets up a maze to prevent them. Please implement a program to help Romeo and Juliet have a reunion. Let them live happily ever after.

2. Task Direction

First, Romeo and Juliet are at the initial position. Second, your program drives them and follows the rules to explore the given maze. Finally, terminate your program according to the termination conditions and output the moving path of them.

3. Maze Specification

- Size: $2 \times 101 \times 101$ (a 2-story 101 by 101 maze space including walls)
- Direction: X-axis, Y-axis, and Z-axis.
- Initial position:
 - ◆ Romeo: (0,1,1) (i.e., 1F position (1,1))
 - ◆ Juliet: (1,99,99) (i.e., 2F position (99,99))
- Symbol representation
 - ◆ X: a wall
 - ◆ • : a road
 - ◆ O: a stair

4. Walk Rules

- Romeo: start from (0,1,1) and stop at (1,99,99) of the maze, and it walks according to the rule that *Upstairs* > *Right* > *Down* > *Up* > *Left*. Additionally, Romeo will not go downstairs when he has already gone upstairs.
- Juliet: start from (1,99,99) and stop at (0,1,1) of the maze, and it walks according to the rule that *Downstairs* > *Left* > *Up* > *Down*

- *Right.* Additionally, Juliet will not go upstairs when she has already gone downstairs.
- Romeo goes one step further before Juliet goes one step.
- The road traveled will not be visited again unless there is no way to go, which avoids the infinite loop.
- The stair is a portal, that is, when somebody reaches the stair, goes straight up or down one floor.

Example:

The position of the stair is (0,2,3). When Romeo reaches the stair, his position directly becomes (1,2,3). In other words, going upstairs does not require an extra step.

5. Termination Condition:

- They meet each other.
- One of them arrives at its destination first.

For example:

Romeo : (0,1,1) → (0,2,1) → ... → (1,99,99)

Juliet : (1,99,99) → (1,99,98) → ... → (0,23,55)

The program is terminated since Romeo arrives at its destination first.

6. Input and Output

- Input:
 - ◆ Notably, the inputs adopt the standard I/O, instead of reading files.
 - ◆ Format
 - ❖ The number before the maze is the floor of the maze.
 - ❖ Next, 101 rows represent the map of the maze.
 - ❖ See more details in the test files.
- Output:
 - ◆ The path including all the rounds of Romeo and Juliet move.
 - ◆ The format is as shown in the example and the **test files**.
 - ◆ The number of rounds starts at one because it does not include the initial states.
 - ◆ Finally, output the one of the following states whether Romeo

and Juliet meet each other or not.

- ❖ They encounter each other in (z,x,y).
- ❖ They do not encounter each other in this maze.

➤ Example of a two-floor 11 by 11 maze :

Symbol: underline is the space, \n is the newline.

Input	<pre> 1 XXXXXXXXXXXXX X.....X.XXX X.X.XXX.XXX X.X...XXXXX X.XXX...XXX X.X.X.X.XXX X.X.XXX.XXX XoX...XX.XX X.....X XXXX.XXXX.X XXXXXXXXXXXXX 2 XXXXXXXXXXXXX X. ...X.XXX X.X.XXX.XXX X.X...XXXXX X.XXX...XXX X.X.X.X.XXX X.X.XXX.XXX XoX.XXXX.XX X.....X XXXX.XXXX.X XXXXXXXXXXXXX </pre>
Output	<pre> ==Round: _1==\n Romeo:_(0,2,1)\n Juliet:_(1,9,8)\n ==Round: _15==\n Romeo:_(0,7,6)\n Juliet:_(0,1,2)\n ==Round: _16==\n Romeo:_(0,7,5)\n </pre>

	Juliet:_(0,1,1)\n They do not encounter each other in this maze.
--	---

7. Limitation Condition

- Please utilize stacks to implement your program.
- Do not copy the code of your classmates.

8. How to Submit

To submit your files electronically, login DomJudge website through the following url:

<http://140.123.102.98:12345/>

Press the submit button and choose the homework questions you want to submit. After submitting your code, DomJudge will give you a result to tell you whether your code is correct or not. However, during the demo time, your code will be evaluated by different sets of test cases. Please make sure your code can work correctly based on the description above. Additionally, you must compress your code and the README file into a **zip** file and upload it to Ecourse2.

9. Grading policies

- 10% - Readme file, code style, and comments in the source code
- 90% - Source code of each part can be compiled without any errors and the results are correct.