

Assignment #2

1. Problems

1) Implement postfix notation solver.

Postfix notation is defined " $(pexp_1)(pexp_2)op$ ". It's an unambiguous way of writing an arithmetic expression without parentheses. **For example the postfix version of " $((8 + 9) * (5 - 3)) / 4$ " is " $89+53-*4/$ "**

The conditions given are:

- Numbers are natural numbers below 10 (e.g., 1,2,3, ... 9).
- Operators are +, -, *, and /
- There is no white space in the expression.
- Return the result value of the given equation.
- Use Stack to solve this problem.

Example of input

- 89+53-*4/

The return value of the above equation

- 8.5

2) Implement a solver of gem collector from a dungeon

You entered a dungeon to collect gems. The good news is that you have a dungeon map. **You have to calculate the maximum number of gems you can get while moving from dungeon entrance to exit.** Dungeons consist of a path, obstacles, gems, exit, and entrance.

The conditions given are:

- There is only one exit and one entrance each.

- You start with the entrance.
- You can move to an area excluding obstacles.
- You can move up, down, left, and right by one slot. It counts one movement.
- When you arrive where the gem is, you can get the gem once.
- **You cannot revisit** a place you have visited once, including the entrance, path, exit, and gem.
- **Map information= Entrance: s, Exit: e, Gem *, Path: ., Obstacle: !**
- The map size does not exceed 15x15.
- Use the given Stack or Queue to solve this problem.
- The first line of dungeon map input has the size of the dungeon map (y x)
- If there is no path between the entrance and exist, print '-1'.
- There is a timeout for each test case. timeout is 10 seconds.
- Do not use recursion methods.

Input example 1

```
3 4
s.!
..*.
.e..
```

The return value of example 1

```
1
```

Input example 2

```
7 6
s.!.*
..!...
..!...
..!...
..!*!*
.....
*!....
**!!!e
```

The return value of example 2

3

Input example 3

4 4

s.!*

...!

..!e

The return value of example 3

-1

3) Print the path of your answer.

Print any path of your answer in java stdout (e.g., system.out.print)

Input example

7 6

s.!...*

..!...

...!...

..!*!*

.....

*!....

**!!!e

Output example

--- maze ---

1..111

1..1.1

1..1.1

1..1.1

1111.1

....1

....1

Maze answer: 3