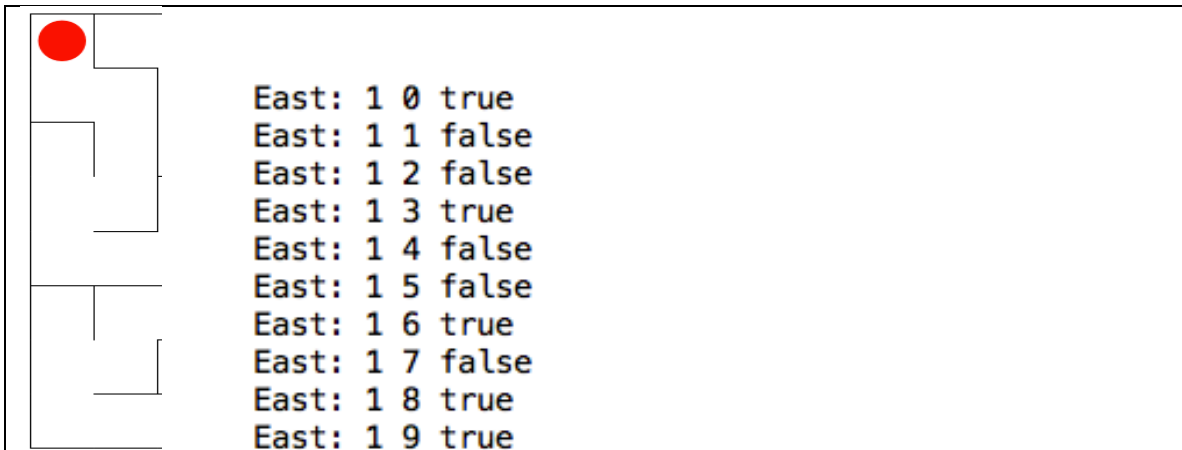# CS2300 Assignment 1 (50 points)

Due Date: 9/9 (11:59pm)

Description: In this assignment, you will write a Java program to determine a path from the start (1, n) of a maze to the finish (n, 1) of the maze where n is the size of the maze.

Requirement: In this assignment, you MUST use a stack to solve the problem. You MAY NOT use the stack class from the Java library (java.util.Stack) and you MUST write your own stack class.

Understand the Data: The maze is square and the size is n x n. The lower left position is (1, 1) and the upper right position is (n, n). The starting position is at position (1, n). When you reach the position (n, 1), you have found the exit and you are done. The maze is characterized by four boolean arrays (north, east, south, and west) indicating the walls where a wall is present in a particular direction if it is true. See the following figure as an example.


```
East: 1 0 true
East: 1 1 false
East: 1 2 false
East: 1 3 true
East: 1 4 false
East: 1 5 false
East: 1 6 true
East: 1 7 false
East: 1 8 true
East: 1 9 true
```

When you move to the next position, you should call "drawDot(x, y, "BLUE")". When you backtrack to the previous position, you should mark the backtracked position by calling "drawDot(x, y, "GRAY")".

In this assignment, you are provided an "StdDraw.jar" for drawing a maze and dots. To compile with the source file,
>> javac Maze.java –cp StdDraw.jar
If you are using Ellipse, you need to add the jar into your project.

In addition, your program should read a maze file from the command prompt.
>> java Maze maze16.txt

Grading:
Submit your source files along with a readme indicating how to compile your files via blackboard.

Successfully compile and link (10 points)
Solve the maze8.txt   (10 points)
Solve the maze16.txt (10 points)
Solve the maze32.txt (20 points)

** If you do not write and use your own stack to solve this problem, you will receive zero point.