Data Structures
Term Project 2
Due: December 2nd Tuesday by 23:59 in LMS

This is the second application of a series of applications that you will implement as your term projects. You are more than welcome to discuss the project with your friends and actually encouraged to do that. But, this is an individual project so that code sharing is not allowed.
No late projects will be accepted.
You can **only** use constructs and data structures we discussed in class.
You might be interviewed briefly about your implementation. Failure to answer any question will result in no credit for the project.

This is the same problem you had with the first term project with one important difference: You will the **shortest** path.

The maze is the same as before. You are provided with a maze in the maze.txt file with **two openings**: one **entrance** and one **exit**. It is a **15×15 maze** with that setup (entrance on the top row, exit on the bottom row). The walls of the maze are encoded as “1” and openings/halls are with a “0”. Your task is read this file and find **the shortest path** if exists (from the entrance to the exit) using **the queue data structure. You can either implement** of your own or use the LinkedList class in Java. If there is a (shortest) path from the entrance to the exit, your application should print the locations/coordinates of the openings (“0”) all the way from the entrance to the exit. You can assume that the (0,0) coordinate is the top left corner of the maze, that is, the top left corner “1” has the coordinate (0,0), and the (14,14) is the bottom right corner of the maze.

Steps to take:

1. Be able to read the file and store the maze in an appropriate data structure. (35 points)

2. Implement/Use your queue data structure to store your moves. (15 points)

3. Remember which openings you have been to. (15 points)

4. Add all neighbors of the front of the queue to the end of the queue (if necessary, that is, the front of the queue is not the exit). (20 points).

5. Print all the coordinates of in the shortest path from the entrance to the exit once you hit the exit. (15 points)

Deliverables:

Your source code (not the whole project folder!) only. The output of your program showing the coordinates of the openings (“0”) of the path all the way from the entrance to the exit appended to your source code as comments.

Put as comments after your output that briefly explain which steps you finished.

Name your source file as: YourName_project1.java.