**CSCD 300 Homework 3**

**Solve Maze Puzzle with Recursion (Total 100 points)**

**Turn in:** On EWU Canvas CSCD300🡪Assignments🡪Hw3🡪Submit. Please put all your source code(.java) files and input files and output files together into a zip file. Name the zip file with your last name followed by first initial plus hw3.zip. For example, smithjhw3.zip is for John Smith.

If you forget to include your source code in the zip file, you get a zero credit for this homework. If your code shows a compile-time error, you get a zero credit.

**Problems Description**

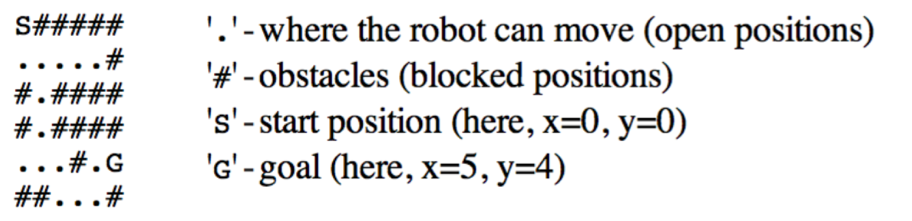
Based on the materials we discussed in classroom, you are required to implement and solve a given maze puzzle using recursion. You will implement two algorithms. The first algorithm is **required**, while the second algorithm is **optional** and offers bonus points.

Algorithm 1) You are required to write a **recursive** program that will print out the first path from the Start to Goal, the first path you program can find in a maze puzzle. Please write this path along with the puzzle representation into a text file, named as **path1.txt. See below for the format of the output file. ( 100 points )**

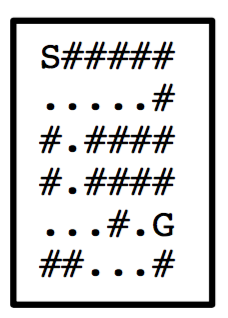
Algorithm 2) You can optionally choose to write a **recursive** program that will print out all possible paths from the Start point to Goal in the puzzle. Please write all paths along with its puzzle representation into another text file, named **pathAll.txt**. The format of the output text files is provided in section below. **(20 bonus points for algorithm 2**.)

**Rules:**

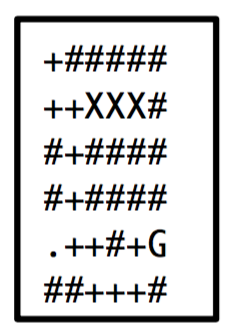
1. You have to use the required input format. That is, the maze puzzle is represented as a matrix of characters as following.

****

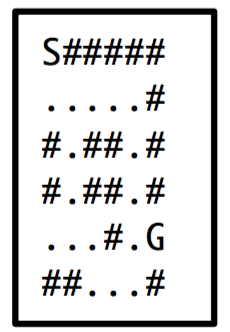
Your program should read in the maze puzzle from a text file and saves it into a 2D array,



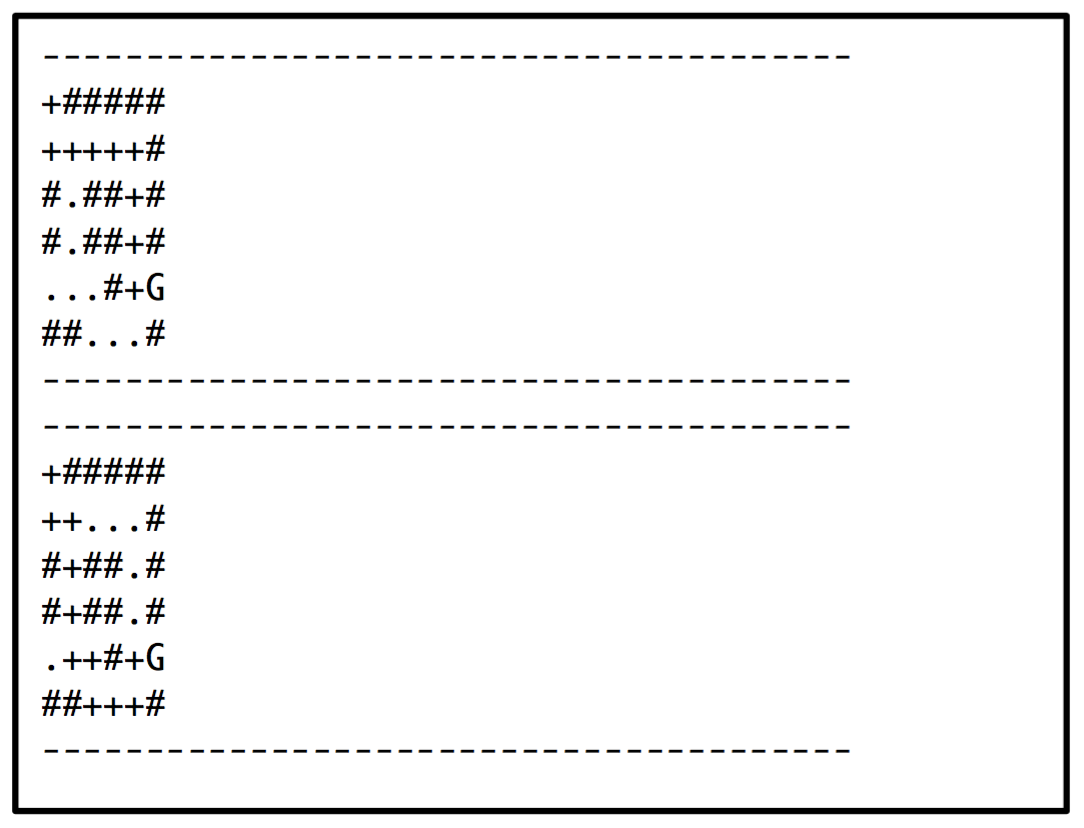
1. When output the first path that your program finds (algorithm one), the output file is formatted as shown in the following. A **‘+’** is used to mark a path from Start to Goal, **‘X’** shows your program went that cell, but no path is found through that cell. An example is shown in the below.



1. If you like to challenge yourself, you will extend the idea we learned in classroom and print out all paths in the puzzle. If your input text file is like the following,



The output file in algorithm 2 would look like below, (output all paths into a file). Different solutions are stored in a single text file, but are separated by dashed line (one line or two lines).



**What is provided?**

Two input text files are provided for test purpose, **inputPuzzle1** and **inputPuzzle2**.