
5. Maze Solver

Program Name: Maze.java

Input File: maze.dat

The local newspaper wants to start running daily mazes for children. They contracted a software development firm that wrote software that can randomly generate mazes. Unfortunately the software is buggy and not every maze is functional. What is even worse is that the company has now gone under! To save money, the editor has contacted you to write a program that can determine if a maze is solvable or not. A maze is solvable if you can get from the starting position to the end position by moving horizontally and vertically (no diagonal movement) though the maze.

Input

The first line of input will contain a single integer n that indicates the number of mazes to follow. Each maze is a 20 by 20 matrix, occupying 20 characters per line for 20 consecutive lines. Each maze is guaranteed to have one beginning spot, denoted by a capital S, and one ending point, denoted by a capital E. Wall characters are denoted by pound signs (#) and pathable area characters are denoted by periods (.). There is exactly one blank line separating matrices.

Output

For each maze you should print "Maze #N: " where N is the number of the current maze, followed by a space and then by YES if the maze is solvable or NO if the maze is not solvable.

Example Input File

```
2
#S#####
#.....#####
####.#####
####.#####.####
####.#####.####
####.#####.####
####.....#####
#####.#####.####
#####.###.#####
#.....#####
##.##.#####
##.##.....#####
##.#####.#####
##.....#####.###
#####.##.###.#####
#####.#####.#####
#####.#####.#####
#####.#####.#####
##.....##
#####.##
###.....##
###E#####
```

(continued on next page)

5. Maze Solver (cont.)

```
#####
#.....#####
####S.#####
####.#####.####
####.#####.####
####.....#####
#####.#####...####
#####.#####...####
#.....#####
#####.#####
#####.....#####
##.....#####
#####.#####
#####...#.#...#
#####...#####
#####.#####
##.....#
#####
###.....#
###E#####
```

Example Output to Screen

Maze #1: YES

Maze #2: NO