CS Games 2016

Relay Programming

Participants	**1**
Workstations	**1**
Value	**5%**
Duration	**3 x 1 hour**

Vim Maze Solver

Problem Description

In this part of the challenge, you must implement a Maze solver in Vimscript.

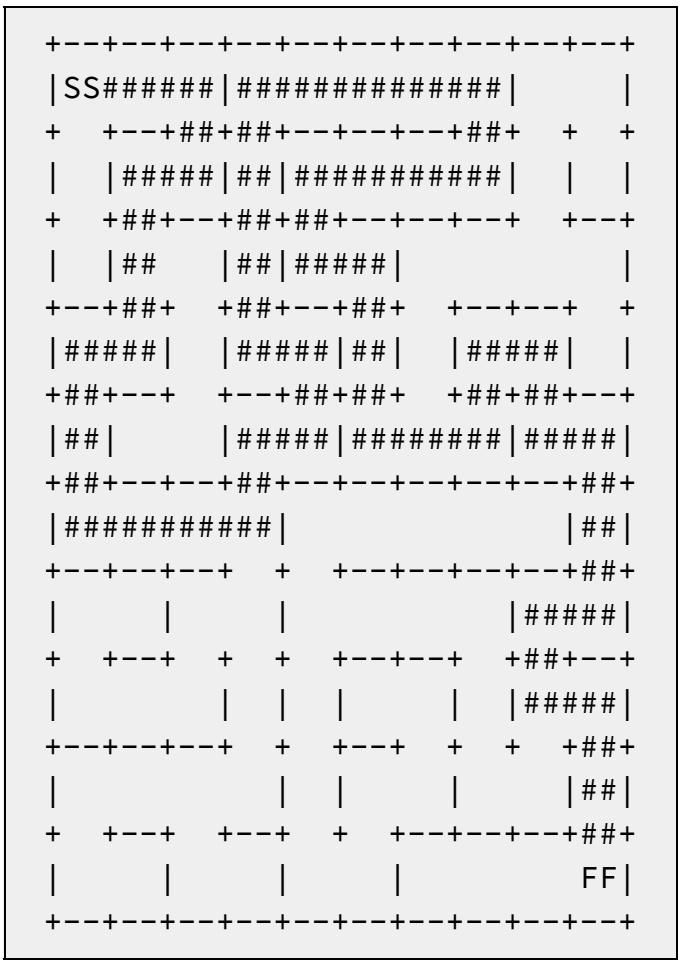
Your program must read the current buffer from vim and interpret it as a maze. It must then modify the buffer to print a possible path from start to finish. To launch your solver, you must use the `:solvemaze` command.

Example *of maze file*

```
+--+--+--+--+--+--+--+--+--+--+
|SS      |              |     |
+  +--+  +  +--+--+--+  +  +  +
|  |     |  |           |  |  |
+  +  +--+  +  +--+--+--+  +--+
|  |     |  |     |           |
+--+  +  +  +--+  +  +--+--+  +
|     |  |     |  |  |     |  |
+  +--+  +--+  +  +  +  +  +--+
|  |     |     |        |     |
+  +--+--+  +--+--+--+--+--+  +
|           |              |  |
+--+--+--+  +  +--+--+--+--+  +
|     |     |           |     |
+  +--+  +  +  +--+--+  +  +--+
|        |  |  |     |  |     |
+--+--+--+  +  +--+  +  +  +  +
|           |  |     |     |  |
+  +--+  +--+  +  +--+--+--+  +
|     |     |     |        FF|
+--+--+--+--+--+--+--+--+--+--+
```

`SS` represents the beginning of the maze, and `FF` the end of the maze.

Example *of a solution for the previous example.*



Correction

You must turn in your `.vim` files, we will evaluate your work based on the following criteria:

- ❏ Functionality - 50%
- ❏ Efficiency - 20%
- ❏ Optimality of the proposed solution - 15%
- ❏ Code Quality - 15%