Week 7

Maze (80%)

Write a program to read in a maze typed by a user via `argv[1]` You can assume the maze will always be no larger than 20×20, walls are marked with a # and the rest are spaces. The entrance is always the gap in the wall closest to the top lefthand corner. The size of the array is given on the first line of the file (width,height):

```
10 10
##########
  #      #
# # #### #
# # #  # #
# # ## # #
# #    # #
# # ## # #
# #### # #
#      #  
##########
```

Find the exit from the mze using the recursive approach described in lectures.
If there is an exit possible, then print out the maze with the exit route shown using full stops for the route and hashes for the walls :

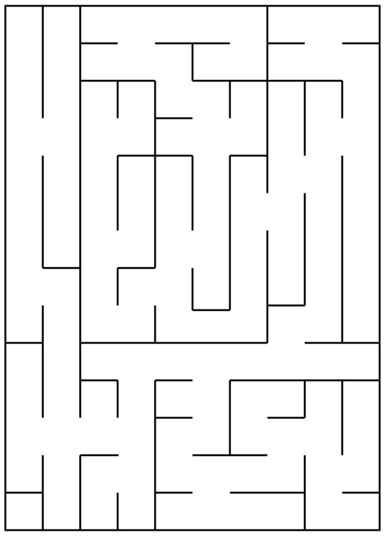
```
##########
..#......#
#.#.####.#
#.#.#  #.#
#.#.## #.#
#.#....#.#
#.# ##.#.#
#.####.#.#
#......#..
##########
```

In this case `main()` should return 0. If there is no exit, then the function should print out a suitable message declaring so and `main()` should return 1.

Maze Generation (10%)

Generating interesting mazes is a challenging problem. One method is the *Recursive Division* Method, see:

`http://en.wikipedia.org/wiki/Maze_generation_algorithm#Recursive_division_method`



Using this method (chop a chamber into **four**, leave gaps in **three** of these walls) and no other, to generate a randomly-sized maze. If you get this method to work, the user may select it by using the word `RANDOM` for `argv[1]` rather than a filename.

SDL (10%)

Show the output for your maze solving using SDL. To switch on grahical ouput, `argv[2]` will be the word `SDL`. If the user doesn't specify `SDL` then the graphics subsystem should NOT be initialised, so that we can still test your basic assignment in a text-only terminal.

Hints

- Explain whether you've done only the basic, basic+SDL, basic+generation, or full assignment in a comment at the top of your code.
- Once again I'm assessing your ability to program in C, not your expertise in maze-based algorithms. Submit what you've done, even if it doesn't fully work, and explain in a comment where you've got to.
- Neill is lecturing his Golden Rules of programming - these are the main style rules we're using to assess your programs.
- Begin slowly - try to solve asimplified version of this problem first.
- Don't use any other way of solving the maze (e.g. ink-blotting, left-hand on the wall).