P2: Path Thru a Maze

You are to develop a program that can traverse a maze in the shortest possible manner.

Input

Your program should be able to handle input in the form of a maze represented by an *m* × *n* character of 1s and 0s, with an "E" or "e" placed in the entry location. Note, there will only be one entrance.

You may move vertically or horizontally (but not diagonally) in any direction that contains a 0; you may not move to a location that contains a 1. If you move to a border location that contains a 0, you have found an exit.You may "mark" locations in the maze with some other character value after you have visited them.

You may either use I/O redirection to read in a maze or accept a filename as the first argument on the command line. That is, I want to be able to test your code by typing either ./p2 maze1.dat > out.txt or ./p2 < maze1.dat > out.txt. However, you should not require the user to specify the number of rows or columns in the maze.

I will compile your code using clang++ -std=c++11 -o p2 *.cpp and test it on files maze1.dat, maze2.dat, and maze3.dat which will be given. It is your responsibility to ensure that your code will compile and run successfully under these conditions.

You may also test your program using following mazes:

```
11E11101
10000001
10111101
10000001
11111111
```

```
1111E11111
1001010001
1001000101
1101011101
1101001101
1001000001
1011010111
1001110001
1000000101
1111110111
```

Sample Source Files: Maze 1, Maze 2, and Maze 3 **will be provided**

Output

Your output should consist of a shortest path through the maze. You should output the solution as

- a list of ordered pairs. In addition, you may
- print the original maze with the shortest path through it *clearly* marked in some manner (for additional creativity points).

If a solution exists, *the last line printed* should be the number of steps in the shortest path(s) through the maze. If no solution exists, then a message should be printed which indicates that fact.