CS 461
Spring 2019
Program 1
Due Sunday night, Feb 10.

Your first programming assignment uses unguided search to solve a game problem.

THE PROBLEM:
Consider the 8-puzzle: 8 tiles each numbered 1-8 on a 3x3 grid, with one space empty. Any adjacent tile can be moved into the empty square (in effect swapping the locations of that tile and the empty square). The goal is to begin with some arbitrary arrangement and end with the tiles in the following arrangement:

```
123
456
78E
```

where ‘E’ denotes the empty square.

One possible complication is that permutations of the game board fall into 2 disjoint sets of odd or even parity, only one of which can reach the goal. Thus, half of all possible tile arrangements cannot lead to a solution. Your program must correctly detect whether a solution is possible or not.

Your input file is a simple text file. The first element, on a line by itself, is the number of puzzles contained in the file. After that will be the specified number of puzzles in the above format (3 lines of 3 characters each, each character will be a digit 1-8 or an upper case ‘E’). Each puzzle is separated from the next by a blank line. Your program will be tested on a different data file with the same format.

Given a puzzle, your program must find a solution if it exists. There are several different strategies that can be used, of varying complexity and data-storage needs. For each puzzle, your program should print a list of the moves that should be made to reach the solution. If there is no solution for a puzzle, your program should print a short message to that effect.

There is no requirement that your solution be shortest-length or optimal, but must be correct.

Programming notes:

- Your program may be in Python, Java, C, C++, or C#.
- If you’re applying an algorithm that will find the shortest path, here’s a useful statistic: The longest shortest path for a solvable 8-puzzle has length 31.
- You will probably want to avoid any loops (returning to a state you have already visited) in your search.
- There are pros and cons to depth-first v. breadth-first search; use whichever you’d prefer.