# EECS 233 Programming Assignment #4
### Due Thursday, Dec. 6 before midnight
### 100 points

In this programming assignment you will implement an 8-puzzle solver. The 8-puzzle is a simple puzzle in which there are 8 numbered tiles arranged in a 3x3 grid with one blank space. The tiles only slide horizontally or vertically, and any tile that is adjacent to the blank space can be moved into that space. The goal is move the tiles from their initial, disordered state to the following goal state:

|   |   |   |
|---|---|---|
|   | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Your program should do the following:
1. Initialize the puzzle to the goal state or a specified state if supplied. *Note*: you cannot choose states purely at random, because the goal state is not necessarily reachable from all puzzle states.
2. Randomize the puzzle with a specified number of random moves from the goal state.
3. Print out the current state of the puzzle to the console.
4. Perform a depth first search to solve the puzzle.
5. Perform a breadth first search to solve the puzzle.

Both of your search routines should:
- ensure that you do not repeatedly loop through the same state sequence
- check that you have not exceeded a pre-specified maximum allowed storage memory
- print out the solution sequence that goes from the start state to the goal state

**Test code**. Write test code that demonstrates the correctness of each of your methods. You will be graded on both the completeness and conciseness of your tests, i.e. a small set of well chosen test cases is better than a large set of redundant tests that miss some essential case. Your test cases should demonstrate the differences and limitations of the breadth-first and depth-first solutions. Your program should also clearly indicate where it can be modified so that it can be easily tested by the graders.

**Grading**: puzzle randomizer: 10%, depth-first solver: 30%, breadth-first solver: 30%, code design, style, and comments: 15%, test code: 15%.

**Extra credit**: (25 points): Generalize your code to implement an N-puzzle, i.e. $N = n \times n - 1$ for integer values of n, i.e. N=3, 8, 15, etc.