Lab program - 2

To implement 8X8 puzzle

Algorithm

1	3	2
4	6	7
5	8	0

1	2	3
4	5	6
7	8	0

Step 1: Start

Step 2: Initialize puzzle.
Create an instance of puzzle.
The initial board is created randomly using two dimensional array

Step 3: Using loop we have to solve the puzzle.

Step 4: Display the current state of the board using print board function

Call the move method or function to move the number which is specified by the user ~~Stop~~ swap the blank tile with

Step 5: Check for the valid move.
Ensure the move direction is valid based on the current position.

Step 6: Check for puzzle solved.
is solved method is compare the current board state with the goal

Step 7: Once the loop exist the puzzle is solved or else the puzzle cannot be solved.

Step 8: End.

```
Output              Input

1   2   3           1   2   3
4   5   6           4   5   0
7   8   0           6   7   8
```

The moves are ['down', 'left', 'left', 'up', 'right', 'down', 'right', 'up', 'left', 'left', 'down', 'right', 'right']

```
import heapq

class puzzleNode:
    def _init (self, state, parent = None,
                    move = None, cost=0):
    self.state = state
    self.parent = parent
    self.move = move
    self.cost = cost
    self.heuristic = self.calculate_heuristic()

def _lt (self, other):
return (self.cost + self.heuristic) <
(other.cost + other.heuristic)
```