Sliding Puzzle Example

A sliding puzzle, sliding block puzzle, or sliding tile puzzle is a combination puzzle that challenges a player to slide (frequently flat) pieces along certain routes (usually on a board) to establish a certain end-configuration. The pieces to be moved may consist of simple shapes, or they may be imprinted with colours, patterns, sections of a larger picture (like a jigsaw puzzle), numbers, or letters.

Rules of this game are very simple - we are sliding (←, →, ↑ , ↓) tiles to reach the final state in which all numbers are in order with '1' in the top left corner of the board.

Lets use a tuple of length 8 for state representation. Using this representation scheme the following state:

0 1 2

3 4 5

6 7 8

is represented as the tuple:

(0,1,2,3,4,5,6,7,8)

Where 0 represents the blank position.

```
goal = (1,2,3,4,5,6,7,8,0)
start = (8,2,0,4,7,6,3,5,1)
```

```
def printstate(state):
    """
    prints states in a 3x3 tabular form
    """

    print(state[0:3])
    print(state[3:6])
    print(state[6:9])

def printpuzzle(state):
    """
    prints the puzzle state in a better looking form
    """

    print('_______\n|{}|{}|{}|\n|{}|{}|{}|\n|{}|{}|{}|\n_______'.format(state[
0],
state[1],
state[2],
state[3],
state[4],
state[5],
state[6],
state[7],
state[8],
))
```

```
printpuzzle(start)
```

Transition model

```
def result(statein,action):
    """
    Returns the state produced as a result of performing 'action'
    on the given state 'statein'
    """
    stateout = list(statein) # # make a local copy of statein

    if action == 'Up':
        idx = statein.index(0)
        stateout[idx] = statein[idx-3]
        stateout[idx-3] = 0

    elif action == 'Down':
        idx = statein.index(0)
        stateout[idx] = statein[idx+3]
        stateout[idx+3] = 0

    elif action == 'Left':
        idx = statein.index(0)
        stateout[idx] = statein[idx-1]
        stateout[idx-1] = 0

    elif action == 'Right':
        idx = statein.index(0)
        stateout[idx] = statein[idx+1]
        stateout[idx+1] = 0
    return tuple(stateout)
```

Exercise # 1: You need to complete the game so that a player can play slide puzzle game.

Additional Task: write a program in which user gives the starting sequence and the program will solve and display each and every step to solve the puzzle.

Exercise # 2: Write a program to create a class called "MusicLibrary" with a collection of songs and methods to add and remove songs, and to play a random song.

Exercise # 3: Write a program to create a class called "Restaurant" with attributes for menu items, prices, and ratings, and methods to add and remove items, and to calculate average rating.

Exercise # 4: Write a program to create a class called "Bank" with a collection of accounts and methods to add and remove accounts, and to deposit and withdraw money. Also define a class called "Account" to maintain account details of a particular customer.