Quiz: Localization Program

For the purpose of this homework assume that the robot can move only left, right, up, or down. It cannot move diagonally. Also, for this assignment, the robot will never overshoot its destination square; it will either make the movement or it will remain stationary.

Warning:

If you define any helper functions make sure they do not rely on globally defined variables and take all their state as parameters.

Reminder:

Reference 1D sense and move functions developed during the lesson:

```
def sense(p, Z):
    q=[]
    for i in range(len(p)):
        hit = (Z == world[i])
        q.append(p[i] * (hit * pHit + (1-hit) * pMiss))
    s = sum(q)
    for i in range(len(q)):
        q[i] = q[i] / s
    return q

def move(p, U):
    q = []
    for i in range(len(p)):
        s = pExact * p[(i-U) % len(p)]
        s = s + pOvershoot * p[(i-U-1) % len(p)]
        s = s + pUndershoot * p[(i-U+1) % len(p)]
        q.append(s)
    return q
```

```
# test 1
colors = [['G', 'G', 'G'],
          ['G', 'R', 'G'],
          ['G', 'G', 'G']]
measurements = ['R']
motions = [[0,0]]
sensor_right = 1.0
p_move = 1.0
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.0, 0.0, 0.0],
     [0.0, 1.0, 0.0],
     [0.0, 0.0, 0.0]])

# test 2
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R']
motions = [[0,0]]
sensor_right = 1.0
p_move = 1.0
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.0, 0.0, 0.0],
     [0.0, 0.5, 0.5],
     [0.0, 0.0, 0.0]])

# test 3
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R']
motions = [[0,0]]
sensor_right = 0.8
p_move = 1.0
```

```
    [[0.06666666666, 0.06666666666, 0.06666666666],
     [0.06666666666, 0.26666666666, 0.26666666666],
     [0.06666666666, 0.06666666666, 0.06666666666]])

# test 4
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R', 'R']
motions = [[0,0], [0,1]]
sensor_right = 0.8
p_move = 1.0
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.03333333333, 0.03333333333, 0.03333333333],
     [0.13333333333, 0.13333333333, 0.53333333333],
     [0.03333333333, 0.03333333333, 0.03333333333]])

# test 5
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R', 'R']
motions = [[0,0], [0,1]]
sensor_right = 1.0
p_move = 1.0
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.0, 0.0, 0.0],
     [0.0, 0.0, 1.0],
     [0.0, 0.0, 0.0]])

# test 6
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R', 'R']
```

```
p_move = 0.5
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.0289855072, 0.0289855072, 0.0289855072],
     [0.0724637681, 0.2898550724, 0.4637681159],
     [0.0289855072, 0.0289855072, 0.0289855072]])

# test 7
colors = [['G', 'G', 'G'],
          ['G', 'R', 'R'],
          ['G', 'G', 'G']]
measurements = ['R', 'R']
motions = [[0,0], [0,1]]
sensor_right = 1.0
p_move = 0.5
p = localize(colors,measurements,motions,sensor_right,p_move)
correct_answer = (
    [[0.0, 0.0, 0.0],
     [0.0, 0.33333333, 0.66666666],
     [0.0, 0.0, 0.0]])
```

The video shows sensor_wrong and p_stay defined globally and referenced from sense() and move() helper functions. This approach will not actually work when submitting the assignment. You must pass in all required state into your sense() and move() functions from inside your localize() routine: def sense(p, colors, measurements, sensor_wrong): ... def move(p, motion, p_stay): ... def localize(...): ... p = move(p, motion, p_stay) p = sense(p, colors, measurement, sensor_wrong) [You should NOT modify the function signature of localize()]