HW6: CS 110X C 2013

Note: This homework (and all remaining homework assignments) is a **partner homework** and must be completed by each partner pair. When you complete this assignment, you must not share your answers with any other student. Only one person from a partner pair needs to submit the assignment.

Please make sure that **as a team** you work together on these problems, but you also each individually understand the code for each of the associated programs.

Q1	Simulation Exercise
	In a Random Walk over two dimensions, the initial coordinate is (0, 0). Then a random direction is chosen (an angle whose value is 0 ≤ angle ≤ 2*π.) and then the position is updated as follows:

```
angle = random.random() * 2 * math.pi
x = x + math.cos(angle)
y = y + math.sin(angle)
```

Write a Python function `randomWalk(n)` that returns a *tuple* containing two lists, the first value contains `n+1` X-Coordinates and the second contains `n+1` Y-Coordinates. You can assume n ≥ 0.

Write a Python function `plotRandomTrials(numTrials, numSteps)` that computes a number of independent random walks – using the `randomWalk(numSteps)` –and plots all paths using `pylab.plot()` on the same plot.

Sample Output
```
>>> randomWalk(3)
([0, 0.560246160309, -0.07291991366628, 0.4476766974271],
 [0, -0.8283261675561, -0.054310196941, -0.9081129660391])
>>> randomWalk(0)
([0], [0])
```

The above shows how `randomWalk(3)` returns a tuple containing two values, each a list of four values. Note that the first "point" of (0, 0) appears as the first values in each of these two lists. When calling `plotRandomTrials(200, 200)`, for example, the plot on the right shows the accumulation of 200 random walks, each with 200 steps.

Note that I cannot post specific outputs for these "random walks". Use the above values and graph on the right as examples of what your code would produce.

Q2	Simulation Exercise
	In Craps, a player rolls a pair of normal six-sided dice. If the initial roll is a 2, 3, or 12, the player loses. If the roll is a 7 or 11, the player wins. Any other initial roll causes the player to "roll for point." That is, the player keeps rolling the two dice until either rolling a 7 or re-rolling the value of the initial roll. If the player re-rolls the initial roll before rolling a 7, it's a win. Rolling a 7 first is a loss. Write a function `crapsRollWins()` which returns `True` or `False` based upon randomly rolling the dice using the above rules. *Note: You will not be able to test this function by itself, because it will return different values because of its random nature.* Write a function `simulateCraps(numPlays)` that invokes `crapsRollWins()` a number of times and returns the number of wins that were witnessed from those plays. Based upon running a number of trials, what is the probability that the player will win? Be sure to include this value in your submitted homework (see template). Write a `randomTrials(numTrials, numPlays)` that returns a probability (a number between 0 and 1) that averages a number of distinct trials with the given number of plays. Compare the computed result of this game with http://mathworld.wolfram.com/Craps.html. In the long run, will the house always win? **Separate answer: There seems to be something asymmetric about the rules of this game. Temporarily change the rules so that the player wins by getting a 7 or 11, but loses immediately only when rolling a 2 or 12. Now rerun randomTrials(100, 1000) and tell us the player's chance of winning. Does the odds of the game still favor the house?** ==NOTE: WHEN YOU CHANGE THE RULES TO ANSWER THE ABOVE QUESTION, BE SURE TO CHANGE THE CODE BACK TO THE ORIGINAL RULES WHEN SUBMITTING YOUR FINAL ASSIGNMENT.==

Q3	Review of loops
	Write a `repeat(list, multiplier)` function that takes a list of values and a multiplier value. It returns a new list that contains each of the elements in `list` repeated a `multiplier` number of times. You can assume that `multiplier` is a number ≥ 0 and `list` is non-empty.

```
Sample Output
>>> repeat([1, 2, 4], 3)
[1, 1, 1, 2, 2, 2, 4, 4, 4]
>>> repeat([1,2,3],2)
[1, 1, 2, 2, 3, 3]
```

Q4	Review of function parameters
	Define a function `squareEach(list)` that takes a `list` of integer or float values. This function modifies `list` in place. This function does not need to return anything.

Sample output appears on the right. You can assume that `list` is non-empty.

```
Sample Output
>>> list = [2, 4, 7]
>>> squareEach(list)
>>> list
[4, 16, 49]
```

How To Get Started On This Assignment

A template HW6.py file is provided to you.

You are responsible for properly documenting all functions as you have seen me do in class. The rubric will assign points for documentation, so pay attention!

Submit your HW6.py file using the web-based turnin system. As we have mentioned in class, only one of the team members needs to submit the assignment. But just make sure that something gets submitted!

Make sure that you don't write any additional code to invoke these functions, since that gets in the way of the TAs grading the assignments. Good Luck!