Programming Challenge 4

Use the random module to generate random numbers. For instance, `random.randint(1,6)` returns a random integer ranging from $[1, 6]$. To use this module, it is necessary to import `random`.

```
import random
```

In this challenge, you roll a fair six-sided die. If the result is an odd number, you roll once more then stop; otherwise, you stop right after the first roll. Compute the probability of the sum of your total rolls. Let X and Y correspond to the results of your first roll and second roll (if possible), and Z be the sum of your total rolls. Basically, you are asked to compute the PMF of Z.

```
NumberTrials = 1000
sequenceX=[]
sequenceY=[]
sequenceZ=[]
```

If the result of your first roll is an odd number, then you roll once more then stop; if the result of your first roll is an even number, you can assume the result of your second roll is 0. (This assumption does not affect the value of Z since you stop right after the first roll.)

```
for TrialIndex in range(0, NumberTrials):
    sequenceX.append(random.randint(1,6))
    if sequenceX[TrialIndex] == 1 or sequenceX[TrialIndex] == 3 or sequenceX[TrialIndex] == 5:
        sequenceY.append(random.randint(1,6))
    else:
        sequenceY.append(0)
    sequenceZ.append(sequenceX[TrialIndex] + sequenceY[TrialIndex])
```

Look at the empirical distribution of Z.

```
percent = []
for OutcomeIndex in range(2, 12):
    percent.append(sequenceZ.count(OutcomeIndex) / float(NumberTrials))
print(percent)
```

Calculate by hands the probability $Z = 4$ and compare with the corresponding empirical result. Explore how the empirical distribution changes as `N` grows: 1000, 10000, 100000, etc. Note that the range for random variable Z should be $\{2, 3, 4, \ldots, 11\}$.