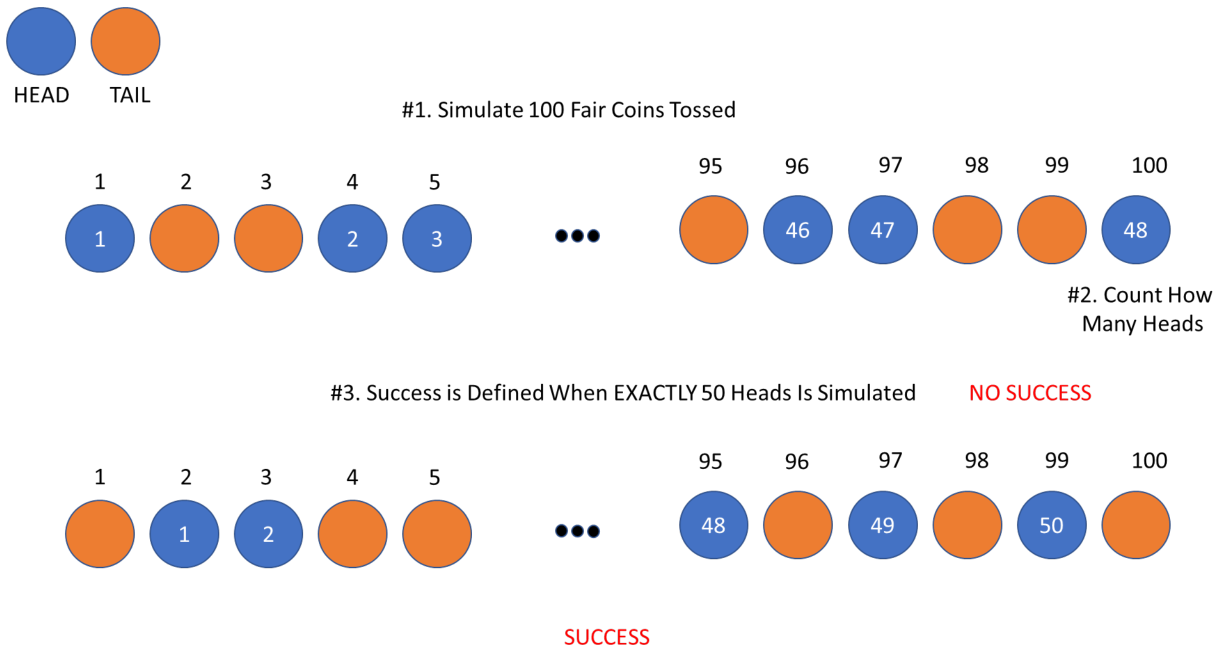
ENG-101
Intro Computing Engineers

DUE: 26 September 2018

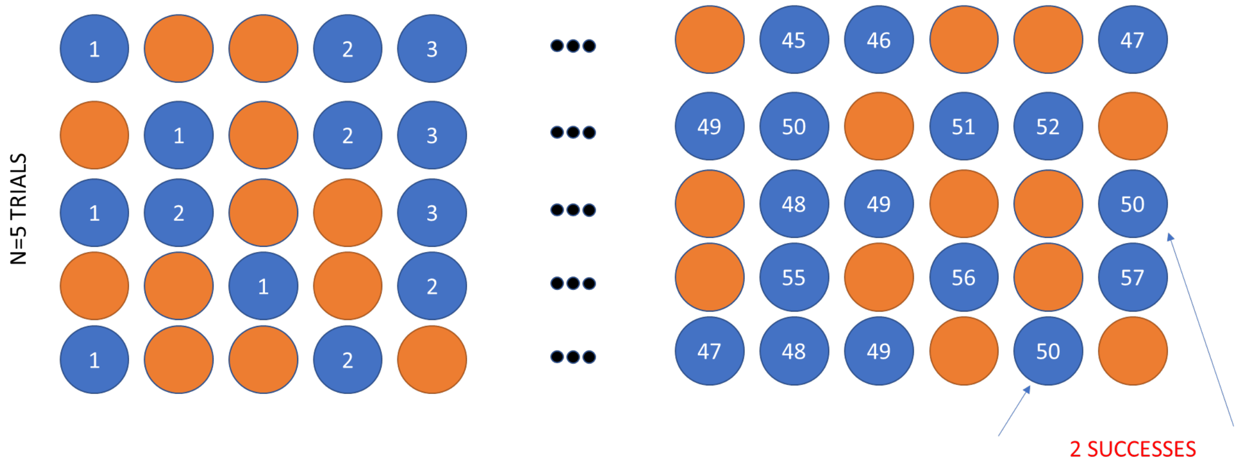
Question 1 (20 points)

Design a well-documented, Python function coinToss(N) to simulate the experiment of tossing 100 unbiased, fair coins in N trials. Record the number of times that the number of coins is <u>exactly fifty heads</u> for each trial.



Have the function coinToss(N) return the number of times exactly fifty heads occurred divided by the number of trials. In the illustration below, coinToss(5) returns 2/5 to the calling program because in two of the five trials exactly fifty heads was observed twice.

coinToss(5)



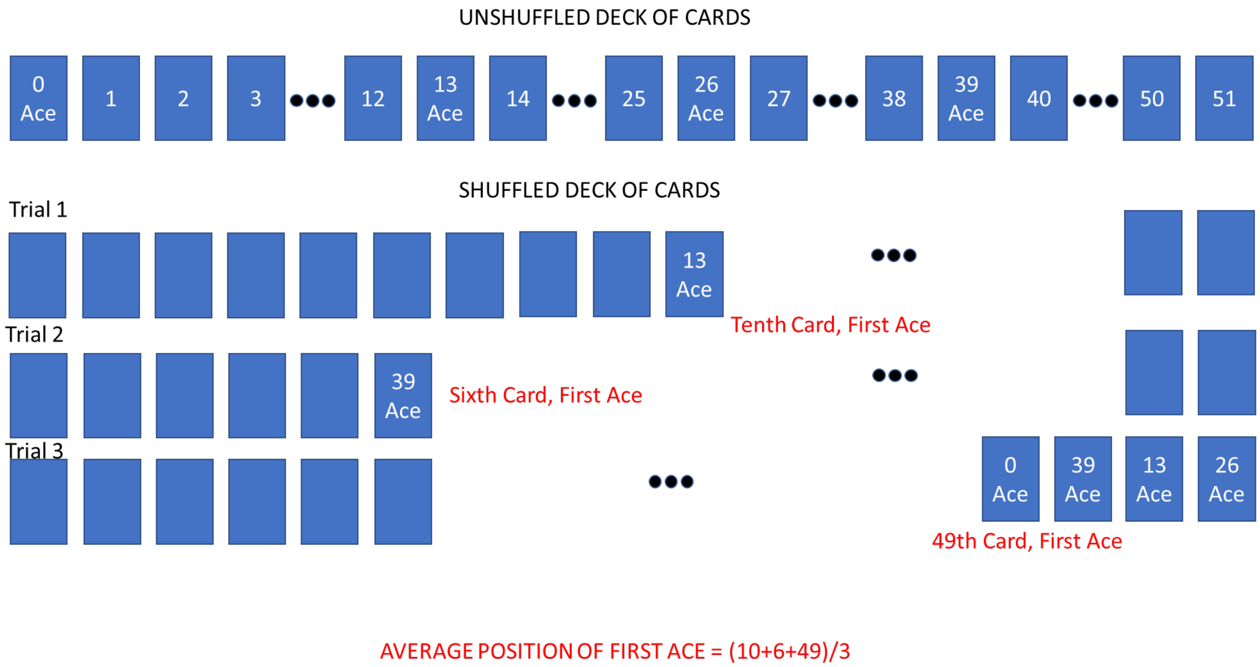
return(2/5) 2 SUCCESSES/5 TRIALS

Probability theory suggests that the probability for achieving exactly fifty heads in one hundred-coin flips is $\frac{100!}{50!50!}\left(\frac{1}{2}\right)^{100}$.

Submit in a Python program randomCoinToss.py that contains coinToss()and a main() function to determine how many trials are needed to ensure that the coin toss result is within 0.00001 of the exact result provided in the equation. You may import the math module to calculate the 100! And 50! to express the exact result.

Question 2 (20 points)

Design a well-documented, Python program myCards.py to simulate shuffling 52 playing cards. In the simulation, the shuffled cards are turned up from the top until the first ace appears. Have your simulation estimate how many cards are required, on average, to reveal the first ace by conducting the simulation 1,000,000 times.



In the simulation above, using three trials, the average position of the first ace is (10+6+49)/3. On average, probability theory suggests that the first ace, of any suit, will be seen at 48/5 position within the deck.

Hint: One way of simulating a shuffled deck is by using the function sample within the random module. Here, shuffleDeck = random.sample(range(52), 52), is a list of shuffled cards.