1. St. Petersburg Paradox:

def stPetersParadox():

numHeads = 0

coinFlip = 0 # O = Heads, 1 = Tails

while coinFlip == 0:

numHeads += 1

coinFlip = random.randint(0,1)

return pow(2, numHeads)

* 1. 100
     1. Average: $8.10
     2. Max: $128
  2. 10000
     1. Average: $20.125
     2. Max: $32768
  3. 1000000
     1. Average: $22.75
     2. Max: $2097152

Assuming that I could play as much as I want, if I were playing upward of 1 million times, I can safely pay up to $20, and safely assume that at worst case I will break even.

1. Monty Hall

def montyHall():  
 doors = [0, 0, 1] # 1 is car, 0's are goats  
 random.shuffle(doors)  
 goatIndexes = []  
 for x in range(len(doors)):  
 if doors[x] == 0:  
 goatIndexes.append(x)  
  
 doorChoice = random.randint(0,2)  
 randomGoat = goatIndexes[random.randint(0,1)]  
  
 if(doorChoice == randomGoat):  
 randomGoat = (randomGoat + 1) % 2  
  
 if(doors[doorChoice] == 1): # 1 if I didn't need to switch  
 return 1  
 return 0 # 0 if switch is needed

* 1. 1000 Games with not switching strategy: 333 cars won out of 1000 games (%33)
  2. 1000 games with switching: 670 cars won out of 1000 games (%67)

TODO: FIGURE OUT THE TRUE PROBABILITY\*\*\*\*\*\*\*\*

1. RISK