```
# -*- coding: utf-8 -*-
"""
Created on Sat Apr  7 19:41:40 2018

EE381 Project 3

@author: Brian
"""
import math
import random
import matplotlib.pyplot as plt

def combination(n , x):
    return (math.factorial(n) / (math.factorial(x) * math.factorial(n - x)))

def binomialDist(trial, expSuc, prob):
    return ((combination(trial, expSuc)) * (prob**expSuc) * ((1-prob)**(trial-expSuc)))




#part 1
TRIAL = 5
SUCCESS = 3
PROBABILITY_OF_SUCCESS = 0.7


print("==============================Problem 1====================================")
print("In 5 trials, with the rate of success as 0.7, the probability of exactly",
      "3 successful trials is",
      '{0:.4}'.format(binomialDist(TRIAL, SUCCESS, PROBABILITY_OF_SUCCESS) * 100),
      "%. This is according to the binomial distribution equation")

'''
Prints out 30.87% for the probability to 3 successful trials
'''

print("==============================Problem 2====================================")
print("The expected outcome is:", TRIAL * PROBABILITY_OF_SUCCESS)

'''
Prints out an expected outcome of 3.5
'''

goodProbability=float(input("Enter the probability of success: "))
numberTrials=int(input("Enter the number of trials to simulate: "))
wantedSuccess = int(input("Enter the number of successes: "))
plotInst = [0,0,0,0,0,0]
```

```
while(not(wantedSuccess in range(0,5))):
    wantedSuccess=int(input("Success needs to be from 0 to 5: "))
success = 0
fail = 0
overallSuccess = 0
for i in range(numberTrials):
    trialSuccess = 0
    trialFail = 0
    for j in range(5):
        r = random.uniform(0,1)
        if r < goodProbability:
            trialSuccess = trialSuccess + 1
            overallSuccess = overallSuccess + 1
            print("\tSuccess")
        else:
            trialFail = trialFail + 1
            print("\tFail")
    plotInst[trialSuccess] = plotInst[trialSuccess] + 1
    if(trialSuccess == wantedSuccess):
        success = success + 1
        print("Trial", i + 1, ": Success")
    else:
        fail = fail + 1
        print("Trial", i + 1, ": Fail")

print("The number of Successes : ", success)
print("The number of Failures  : ", fail)


#part 2
print("In the trial, the expected outcome is:", overallSuccess / numberTrials)

#part 3
print("==============================Problem 3====================================")
print("The resulting graph: ")
plt.bar(range(len(plotInst)), plotInst, 1/1.5, color="green")
fig = plt.gcf()
```