EGR-101 Intro Computing Engineers

Due: 27 September 2020 at 6:00PM – start of class.

Question 1 (10 Points)

Is An Armstrong Number Function

Write a well-documented Python program, *hmwk4Q1.py*, that defines a function *isArmstrong.py(N)*. The function accepts an integer N and returns $True$ if N is a Armstrong number, otherwise, returns $False$. Next, define the main function that asks the user for a range of numbers, say from 1 to M, and counts how many Armstrong numbers there are within the range. An Armstrong number is a number, which forms the total of the same number, when each of its digits is raised to the power of the number of digits in the number. For example, 0, 1 and 153 are Armstrong numbers. The latter number is an Armstrong number as $153 = 1^3 + 5^3 + 3^3$. Also, 1634 is an Armstrong number as $1634 = 1^4 + 6^4 + 3^4 + 4^4$.

Grading:

Correct *isArmstrong.py(N)* function (+4 points). Correct *main()* function (+4 points). Include __name__ is set to '__main__' to ensure the program initiates when it is run (+2 points).

Question 2 (10 Points)

Abraham Series to Compute π

Write a well-documented Python program, *hmwk4Q2.py*, that calculates π. Construct a function *abr_pi.py(N)* accepts an integer N and returns an estimate of π computing the Abraham Series using N terms. The series is:

$$\pi = \sum_{n=0}^{\infty} \frac{2\,(-1)^n 3^{\frac{1}{2}-n}}{2n+1}$$

Have the $main()$ call upon your function for various values of N to ensure it is correctly working.

Grading:

Correct *abr_pi.py(N)* function (+4 points). Correct *main()* function (+4 points). Include __name__ is set to '__main__' to ensure the program initiates when it is run (+2 points).

Question 3 (10 Points)

How Many Terms of The Abraham Series?

Write a well-documented Python program, *hmwk4Q3.py*, that determines the number of terms in the infinite series of Abraham to obtain a specified level of precision with the actual value of π, as computed by Python's math module. Your $main$ program prompts a user for an integer M. Your application identifies the number of terms N to achieve an estimate of π within 10^{-M}. Your program should call upon your function defined in *hmwk4Q2.py*. As a comment include the answer to the question when M = 9.

Grading:

Correct *main()* function (+4 points). Include __name__ is set to '__main__' to ensure the program initiates when it is run (+2 points). Correct answer to M = 9 question (+4 points).

Question 4 (10 Points)

My Birthday Function

Write a well-documented Python program, *hmwk4Q4.py*, that simulates a single trial of a person walking into a room on N people and determining if your birthday matches any of the other N people in the room. Design a function *myBirthday(N)* that returns $True$ if a match is found, or $False$ if no match is found. Import the python random number module into your program.

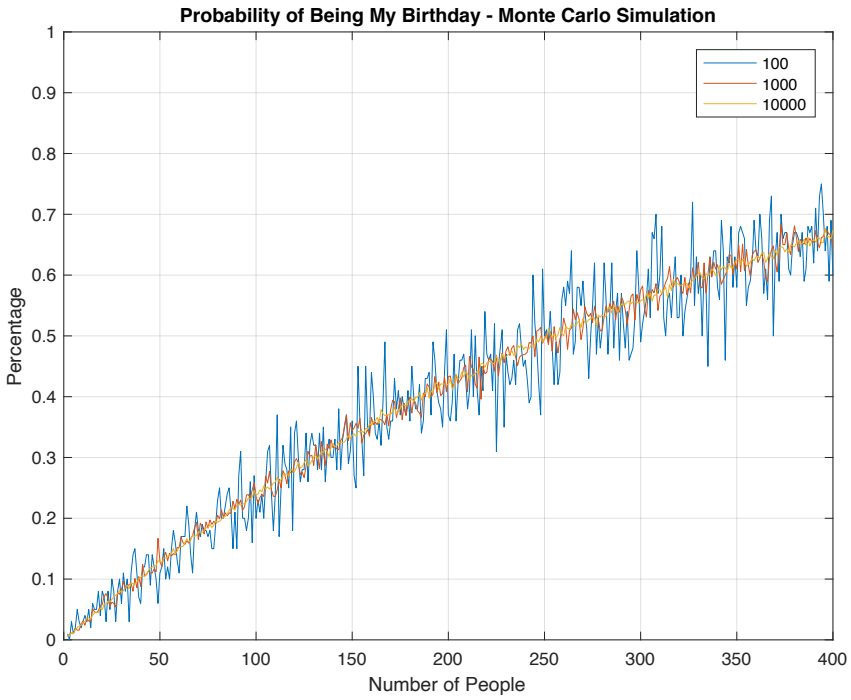
Grading:

Correct *main()* function (+4 points). Include __name__ is set to '__main__' to ensure the program initiates when it is run (+2 points). Correct *myBirthday(N)* function (+4 points).

Question 5 (10 Points)

My Birthday Program

Write a well-documented Python program, *hmwk4Q5.py*, that simulates multiple trials of a person walking into a room on N people and determining if your birthday matches any of the other N people in the room. Have your program run the experiment 100, 1000, and 10000 times respectively. For each trial, form three lists of the results as the number of people are varied from 2 to 400 people. The first list is for 100 trials, the second list is for 1000 trials, and the third list is for the 10,000 trials



Grading:

Correct *main()* function with outer-for loop (+4 points). Correct inner-for-loop (+4 points). Correct list (+2 points).