EGR-101 Intro Computing Engineers

Due: 4 October 2021 at 6:00PM – start of class.

Question 1 (10 Points)

Lottery games award players for money if they are able to match two, three four or five integers from integers selected at random in a range 1 to M (inclusive of M). Simulate a lottery game by designing a well-documented Python function, *lotteryPick(M)* that accepts an integer greater than five and returns a list of randomly selected integers. You will need to use Python's random module. Also, consider using the sample function embedded in the random module. Your call to sample should be of the form *sample(range(1, M+1),5)*. In addition, your function should NOT return a value when the user selects an M less than 5. Under this condition have your function return an empty list.

Test your function inside the main program *lottery.py*. Your main program should ask the user for an integer M representing the range M for the lottery. The program should test to ensure the integer is greater than or equal to 5. If not, the program should continue to query the user. Next, your program should call *lotteryPick(39)* one time to represent your "Quick_Pick". A "Quick_Pick" represents a random selection made by you. Next, have your main program call *lotteryPick(39)* again to represent the actual drawing. In addition, have the main program test whether or not your "Quick_Pick" has won by testing if zero, one, two, three, four or five matches have been made – and displaying the result. Finally, ask the user if they would like to play the program again with a "y/n prompt?".

Grading: 2 Points for comments, 2 Points for python function *lotteryPick(M)*, 4 Points for the main program *lottery.py* that tests your function appropriately and 2 Points for a screen capture that depicts the display your main program creates after a drawing.

```
Enter M, the Lottery Range Greater Equal To 5 39
[10, 31, 32, 38, 6]
[15, 31, 21, 37, 18]
1
Winnner: One Match Found
Do you want to play the lottery again (y/n) ? y
[28, 6, 29, 12, 27]
[14, 3, 22, 7, 8]
0
Loser: No Matches in Game
Do you want to play the lottery again (y/n) ? y
[34, 21, 2, 22, 5]
[4, 18, 23, 13, 28]
0
Loser: No Matches in Game
Do you want to play the lottery again (y/n) ? n
logout
```

Screen Capture

Question 2 (10 Points)

Lottery games award players for money if they are able to match two, three four or five integers from integers selected at random in a range 1 to M (inclusive of M). Design a well-documented, Python main program that uses the function *lotteryPick(M)* to estimate the probability of winning a lottery game in the range 1 to M (inclusive of M) by repeating the lottery game 10^N trials. Ask the user for the value of N as a prompt in your program *lottery2.py*. To estimate the probability of winning with only one match, for example, have your code count the number of times one match is found in 10^N trials and divide it by the number of trials.

Grading: 2 Points for comments, 6 Points for the main program *lottery2.py* and 2 Points for a screen capture that depicts the respective probabilities after one million trials.

```
Enter M, the Lottery Range Greater Equal To 5 39
Enter the number of Times to Play the Lottery 10 ** N 8
{0: 48333816, 1: 40269988, 2: 10394198, 3: 972492, 4: 29337, 5: 169}
logout
```

Unscaled Screen Capture – Yours Should Be Scaled!

Number	Count	Count/Ntrial	Ntrial/Count		NYLOTTERY
0	48333816	0.48333816	2.06894486		
1	40269988	0.40269988	2.48323888		
2	10394198	0.10394198	9.62075188		9.62
3	972492	0.00972492	102.828609		102.63
4	29337	0.00029337	3408.66483		3,386.81
5	169	0.00000169	591715.976		575,757.00
Ntrials =	100000000				
Winning	11396196	0.11396196	8.77485786		8.77

100 Million Trials (Ten Times the Number Required For The Assignment)

https://nylottery.ny.gov/take-5

Question 3 (10 Points)

Design a well-documented, Python main program that uses the function *lotteryPick(M)* to calculate the number of lottery tickets required to be sold before the winning ticket, containing all five numbers, is issued. Construct a function *numFailures(M)* that accepts the range of the lottery between 1 to M, inclusive of M. Test these functions in the main program included inside of *lottery3.py*.

Grading: 2 Points for comments, 4 Points for a *numFailures(M)* function, 2 Points for lottery3.py and 2 points for a screen capture that depicts running lottery3.

```
Enter M, the Lottery Range, 1 to M of the Drawing: 39
Tickets Sold Before Winning Ticket Was Issued: 377273
logout
```

Screen Capture – Your Result on Single Trial Will Likely Differ from Mine

Question 4 (10 Points)

Design a well-documented, Python main program that uses the function *lotteryPick(M) numFailures(M)* and to estimate the number of lottery tickets required to be sold before the winning ticket, containing all five numbers, is issued. Construct a function *numFailures(M)* that accepts the range of the lottery between 1 to M, inclusive of M. Your main program in lottery4.py should query the user for the number of trials N to conduct, recording the number of tickets sold before a winning number was found for each trial. Your program should conduct 10^N trials and display the average number of tickets sold over the 10^N trials conducted.

Grading: 2 Points for comments, 4 Points for lottery4.py and 4 points for a screen capture that depicts running lottery4 with M = 10, N=4 (10,000) trials.

```
Enter M, the Lottery Range, 1 to M of the Drawing: 10
Enter the number of Trials to Run the Lottery 10 ** N 4
Number of Trials Conducted 10000
Avg Tickets Sold Before Winning Ticket Was Issued: 249.3595
```

Screen Capture – Your Result on Single Trial Will Likely Differ from Mine

Question 5 (10 Points)

Display results associated with running your lottery4.py program with M=39, N =3. It will take a long time (~ 1/2 hour).

```
Enter M, the Lottery Range, 1 to M of the Drawing: 39
Enter the number of Trials to Run the Lottery 10 ** N 4
Number of Trials Conducted 10000
Avg Tickets Sold Before Winning Ticket Was Issued: 578156.8441
Elapsed Time: 34319.79 sec
```

Screen Capture – M = 39, N =4 (10,000 Trials) Took Eight Hours to Run

Grading: 10 points for the correct answer in a screen capture of your program's result.