Simulation & Modeling Laboratory

1. Write a Python program to print a string from a given input string where all occurrences of its first char have been changed to '$', except the first char itself.
Input String: ‘energetic’
Output: ‘en$rg$tic’

2. Take a list of integers as input. Remove the duplicate integers from the list. Print the updated list.
Sample Input: [1,2,1,3,5,3,7]
Output: Number of unique items- [1,2,3,5,7]

3. Write a Python program to count the number of even and odd numbers from a list of numbers. Print the count. Then remove the even numbers from the list and print the updated list.

4. Write a python program that prints the following pattern:

```
*
* *
* * *
* * * *
* * *
* *
*
```

5. Write a Python program to display all the prime numbers within a range. Given start=25 and end=50.

Expected Output:

```
Prime numbers between 25 and 50 are:
29
31
37
41
43
47
```

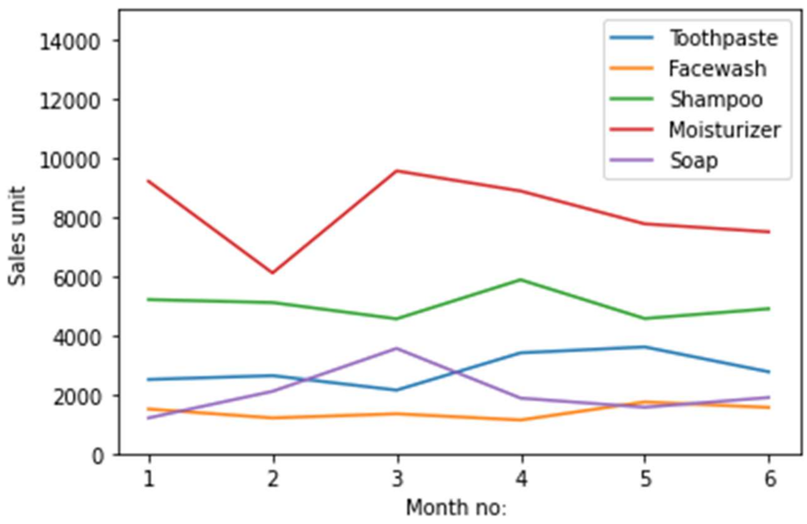
6. Given a two list. Create a third list by picking an odd-index element from the first list and even index elements from second.
Example: listOne = [3, 6, 9, 12, 15, 18, 21]
listTwo = [4, 8, 12, 16, 20, 24, 28]
Expected Output:

```
Element at odd-index positions from list one:[6, 12, 18]
Element at even-index positions from list two: [4, 12, 20, 28]
Printing Final third list :[6, 12, 18, 4, 12, 20, 28]
```

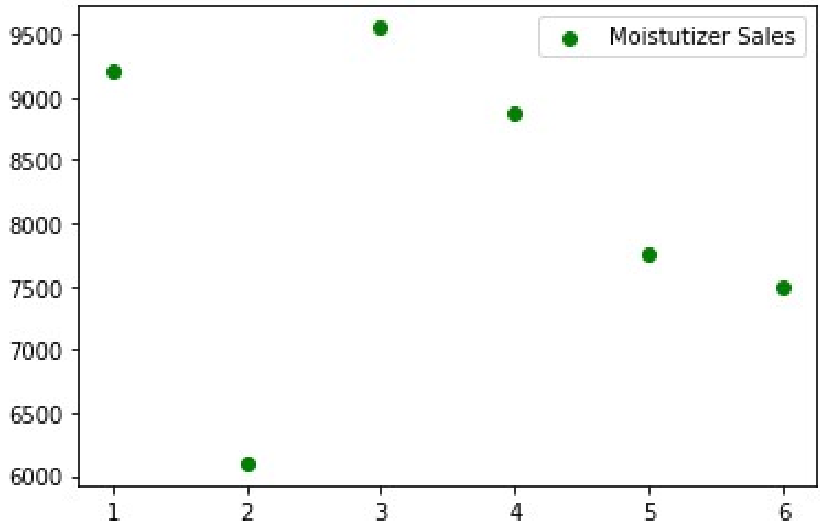
7. See the following Table. The table contains 6 Month’s sales data for different products of the company.

Month no	Toothpaste	Facewash	Shampoo	Moisturizer	Soap
1	2500	1500	5200	9200	1200
2	2630	1200	5100	6100	2100
3	2140	1340	4550	9550	3550
4	3400	1130	5870	8870	1870
5	3600	1740	4560	7760	1560
6	2760	1555	4890	7490	1890

i) Read all product sales data and show it using a multiline plot. Display the number of sales per month for each product using multiline plots. (i.e., separate Plotline for each product). The graph will look like this: (check this with the value given at the table; you will understand)



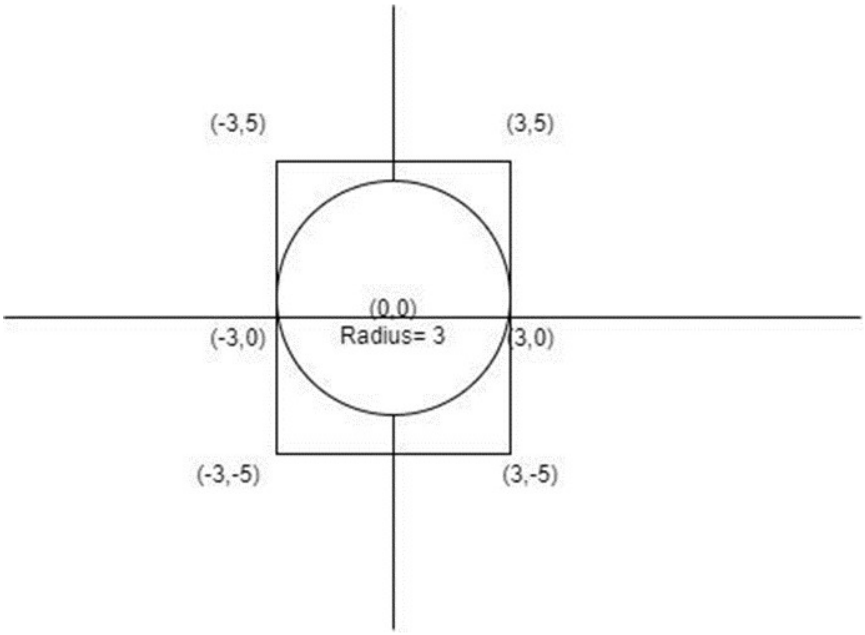
ii) Read Moisturizer sales data for each month and show it using a scatter plot. The plot will look like this:



iii) Calculate the total number of sales each month and show the value using a bar diagram.

8. Find the value of PI using Buffon's Needle Experiment. Understand the Buffon's Needle Experiment and implement the process using Python. Use 500, 1000, 5000 points to see the result. Take d and l as input. Draw the Scatter Plot graph to show the points of hit and miss.

9. See the figure below:



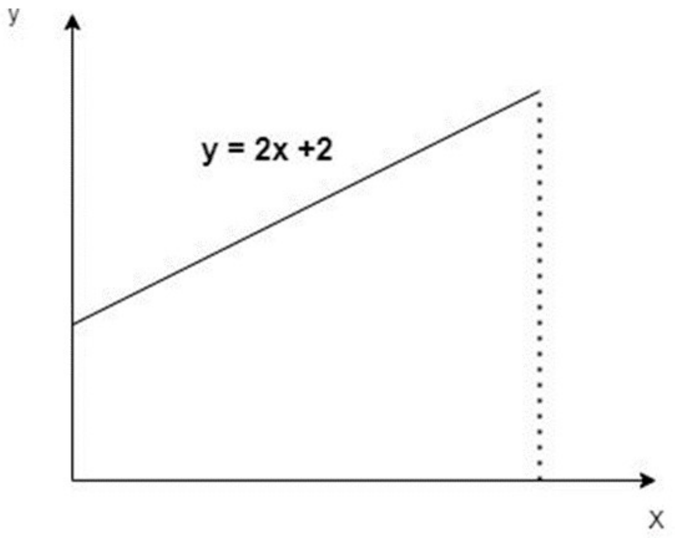
Using Monte Carlo simulation, find the value of PI and area of the circle using the given circle and square. You have to simulate the value for n=100, 1000, 5000 and 10000 trials. Show the scatter plot, value of PI, value of the area for each value of n. (Just as shown in the class).

At the end of the simulation, draw two-bar diagrams.

First Bar Plot: x-axis: number of trials, y-axis: PI -value (Shown in the class)

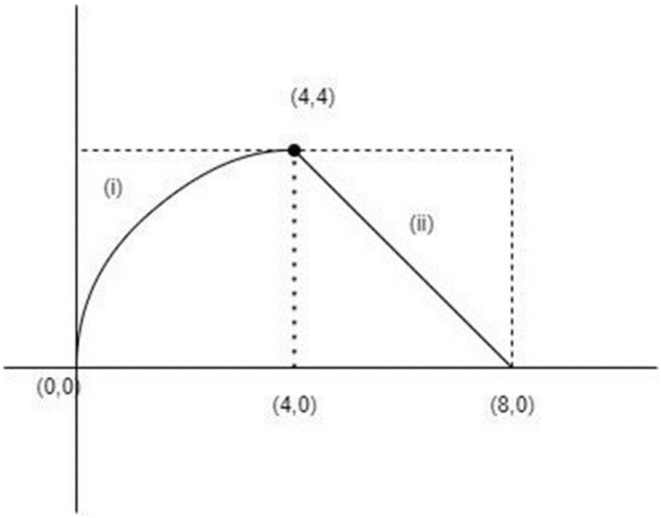
Second Bar Plot: x-axis: number of trials, y-axis: Area of the circle

10. Find the area under the given line below using the Monte Carlo simulation.



Simulate this area for n=100, 1000,5000,10000 trials. For each value of n, print the area of the triangle and draw scatter plots for each case.

11. Find the area under the curve below using the Monte Carlo simulation. Use the drawn rectangle.



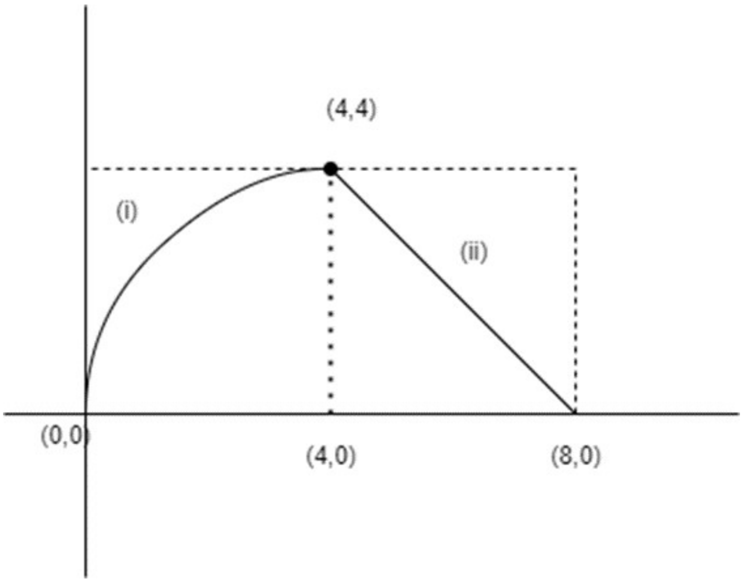
Equation of curve - (i) is: $y^2=4x$

Equation of (ii) is: $y = 8 - x$

Simulate this area for n=100, 1000,5000,10000 trials. For each value of n, print the area and draw scatter plots for each case.

12. Find the Integral of $x^2e^x \ln(x)$ limit x=0 to x=2 using Monte Carlo integral. Use 100, 1000, 5000, 10000 data points and print the integral value and error estimate. Also draw error vs n graph. Here n= number of points.

13. Find the area under the curve below using the Monte Carlo Integral.



Equation of curve - (i) is: $y^2=4x$
Equation of (ii) is: $y = 8 - x$
Simulate this area for n=100, 1000,5000,10000 points. For each value of n, print the area and estimated error of integral.

14. Suppose there are 4 cars, A, B, C, and D. D is chased by B, C is chased by A, B is chased by C, A is chased by D. Initial positions of A, B, C, and D are (10,0), (0,10), (10,10) and (0,0). Velocities of

A, B, C, and D are 3, 5, 7, and 2 ms -1 .

Now Simulate this Chase Problem for t=20 unit time. Print the x and y coordinate value of each vehicle at every time step. If the distance between any 2 vehicles is less than 5 m then a car will shoot its target [not destroy]. Print all the shootings and finally print the number of times each car got shot during the simulation.

Also, draw the graph showing the path of each car. Use plt.plot() function to do this.

15. Suppose G1, G2 and G3 are three separate Congruential Generators. The specification and recursive relation of both generators are given below:

G1	$Z_{1,i} =(13Z_{1,i-1} + Z_{1,i-2} +3) \bmod 16$, $Z_{1,0} =12$, $Z_{1,1} =7$ ($Z_{1,i}$ means Z_i of first CG)
G2	$Z_{2,i} = (12Z_{2,i-1}^{2} + 13Z_{2,i-2}) \bmod 17$, $Z_{2,0} =3$, $Z_{2,1} =5$ ($Z_{2,i}$ means Z_i of second CG)
G3	$Z_{3,i} = (Z_{3,i-1}^{3} + Z_{3,i-2}) \bmod 15$, $Z_{3,0} =2$, $Z_{3,1} =7$ ($Z_{3,i}$ means Z_i of Third CG)

Now G1, G2 and G3 are combined together to generate a random number.
Write a code to implement Wichmann/ Hill Method. Generate 100, 1000 and
5000 random numbers. Show a histogram,
X -axis: index of a random number, i
Y-axis: the random number Z_i

16. You have to implement a queueing model. The model has 2 servers and 1 queue. Let's say we have server-1 and server-2.

The arrival of the customer: If both servers are idle, the customer will go to server-1. If any of the servers are idle and the other one is busy then the customer will go to Idle sever. If both servers are busy then the customer will stand in the queue.
The departure of a customer: Departure can occur from any of the servers. Then the next customer from the queue will be chosen according to queue policy (FIFO, LIFO, or SJF).
Tasks:
1. Print clock_value, server_status of both servers, arrival list of queue, service times in queue, next_arrival time, next_departure time of both servers, total delay, area_under_q(t), area_under_b(t) for both servers at each clock step.

2. We have to implement SJF, LIFO. Report average_delay, expected number of customers in the queue, expected utilization of the server-1, and sever-2 for each queue policy.

3. Important: **You cannot make separate files for each queue policy. You can input a variable in your code to choose the queue policy.** As for example 1 for FIFO, 2 for SJF, and 3 for LIFO.

4. You can use the given Inter-arrivals and service times in your code.
Inter-arrival Times: A 1 = 0.4, A 2 = 1.2, A 3 = 0.5, A 4 =1.7, A 5 = 0.2, A 6 = 1.6, A 7 = 0.2, A 8 = 1.4, A 9 = 1.9, A10 = 0.7
Service times are: S 1 = 2.0, S 2 = 0.7, S3 = 0.2, S4 = 1.1, S5 = 3.7, S6 = 0.6

5. Then take inter-arrival and service times randomly from exponential distribution of mean = 1.2 for inter-arrival times and mean = 1.3 for service times. You can use **np.random.exponential(mean, size)** function from python to do this job. (use np.random.seed(0) for consistent values)

6. Prepare a **report** like below:
Take Interarrival times = exponential with mean 1.2
Take Service times = exponential with mean 1.3
Run the Simulation for 3 cases: num_delays until 10, 30, and 60 customers.
And fill up the following table for each case.

Performance Measure	FIFO	LIFO	SJF
Average Delay			
Expected Number of Customers in the queue			
Expected Utilization of the server-1			
Expected Utilization of the server-2			

17. Inventory System Simulation.

You have to simulate a (M,N) inventory system. The distribution of daily demands and lead time are given below:

Demand	Probability
0	0.10
1	0.25
2	0.35
3	0.21
4	0.09

Lead Time	Probability
1	0.6
2	0.3
3	0.1

Tasks:

1. Input value of m and n. m = maximum inventory
   n = review period

2. Assume that orders are placed at the close of business and are received for inventory at the beginning of business as determined by the lead time.

3. The simulation has been started with the inventory level at 3 units and an order of 8 units scheduled to arrive in 2 days' time. (Initially)

4. Values of daily demand and lead time must be sampled randomly from the above distribution Table. See **np.random.choice(a,p)** function to do this task.(You should use np.random.seed() for consistent random values.)

5. Simulate the system for 10 cycles. Estimate average ending units in inventory and how many days shortage occurs,

6. Draw inventory_level vs day graph.
   X -axis : day number
   Y- axis : Ending_inventory of each day