Task 1:Coin Flipping Simulation and Probability Analysis

You are tasked with conducting a probability analysis using a coin flipping simulation. The goal of this coursework is to deepen your understanding of probability concepts and apply them to practical scenarios. Follow the instructions below to complete the assignment.

Instructions:

Part 1: Simulation

A. Design the Coin Flipping Simulation:
   - Create a Python program that simulates flipping a fair coin multiple times.
   - Specify the number of coin flips (N) as a parameter that you can adjust.

B. Perform the Simulation:
   - Conduct the coin flipping simulation, recording the outcomes of each flip (heads or tails).
   - Keep track of the number of heads and tails observed during the simulation.

Part 2: Probability Analysis

A. Calculate Experimental Probabilities:

- After running the simulation, calculate the experimental probabilities of the following events:
  - ❖ Probability of getting heads (P(Heads)).
  - ❖ Probability of getting tails (P(Tails)).
  - ❖ Probability of getting a specific sequence of coin flips (e.g., HTH).

B. Theoretical Probabilities:
   - ❖ Calculate the theoretical probabilities for the same events based on the principles of a fair coin (P(Heads) = P(Tails) = 0.5).

C. Comparison and Interpretation:
   - ❖ Compare the experimental probabilities with the theoretical probabilities for each event.
   - ❖ Interpret the results and explain any discrepancies or similarities.

Part 3: Report and Presentation

Prepare a report that includes the following sections:

- Introduction (briefly explain the purpose of the simulation).
- Simulation design and parameters.
- Presentation of experimental probabilities and theoretical probabilities.
- Comparison and interpretation of results.
- Conclusion (summarize key findings and insights).

Note: You may adjust the number of coin flips (N) to achieve the desired level of precision in your simulation. **-30 marks**

Task 2: Monte Carlo Estimation of π (pi)

In this coursework, you will use a Monte Carlo simulation to estimate the mathematical constant π (pi). Monte Carlo simulations are a powerful technique for approximating numerical values through random sampling. Follow the instructions below to complete the assignment.

Instructions:

Part 1: Monte Carlo Simulation

A. Design the Monte Carlo Simulation:
   - Create a Python program that simulates random points within a square.

B. Generate Random Points:
   - Generate a total of N random (x, y) coordinates within the unit square, where N is a parameter that you can specify.

C. Determine Points Inside the Quarter-Circle:
   - Implement logic to determine which generated points fall inside the quarter-circle with a radius of 1 and centered at the origin (0, 0).
   - Track the number of points that fall inside the quarter-circle.

Part 2: Estimating π (pi)

A. Calculate π Estimate:
   - Calculate an estimate of π using the following formula:
   - π (pi) ≈ (Number of Points Inside the Quarter-Circle / Total Number of Points) * 4

Part 3: Report

Prepare a report that includes the following sections:

- Introduction (briefly explain the purpose of the simulation).
- Simulation design and parameters (include the value of N).
- Description of the Monte Carlo simulation process.
- Presentation of the π (pi) estimate and the formula used.
- Conclusion (summarize the estimated value of π and its accuracy).

(40 marks)

Task 3: Poisson Process Simulation and Analysis

Instructions:

Part 1: Simulation

A. Design the Poisson Process Simulation:

- Create a Python program that simulates events occurring in a Poisson process.
- Specify the average rate (λ) of events per unit time as a parameter.

B. Generate Poisson Events:

- Implement the simulation logic to generate Poisson events over a specified time interval (e.g., one hour).
- Record the timestamps of each event occurrence.

Part 2: Data Analysis

A. Calculate Event Statistics:

- After running the simulation, calculate the following statistics based on the generated event data:
  - Total number of events observed.
  - Inter-arrival times between consecutive events.
  - The average and standard deviation of inter-arrival times.

B. Visualize Event Data:
- Create visualizations (e.g., histograms or plots) to represent the distribution of event inter-arrival times.

Part 3: Report and Analysis
Prepare a report that includes the following sections:

- Introduction (explain the concept of a Poisson process and the purpose of the simulation).
- Simulation design and parameters (include the value of λ and the time interval).
- Presentation of event statistics and visualizations.
- Discussion of findings and insights into the Poisson process behavior.

Analysis:
Analyze the results in the context of Poisson processes. Discuss how the choice of λ and the time interval affect the observed event patterns and inter-arrival times.

Part 4: Advanced Challenge

A. Rate Variation Simulation:

- Extend the simulation to explore the behavior of a Poisson process with varying event rates (λ) over time. Observe how changes in λ impact event occurrences.

(30 marks)

Submission Guidelines:

- Submit your Python code along with comments explaining the logic of each tasks
- Compile the documentation/report all three tasks into a single file.Submit the report as a PDF document.
- Be prepared to present your findings during the designated presentation session.