Practical 6

Name : $Deepak\ Kumar\ Bharti$

Roll No. : 25056765013

Group : A

Part A : Law of Large Numbers (LLN)

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

2. Consider sample sizes

n = 10, 50, 100, 500, 1000

In [2]:
```
n_values = [10, 50, 100, 500, 1000]
print("Sample sizes:", n_values)
```

```
Sample sizes: [10, 50, 100, 500, 1000]
```

---

1. Generate random samples from a **Uniform(0,1)** distribution.

In [3]:
```
# Seting seed for reproducibility
np.random.seed(42)

# Generating samples from Uniform(0,1)
uniform_samples = {}
for n in n_values:
    uniform_samples[n] = np.random.uniform(0, 1, n)
```

---

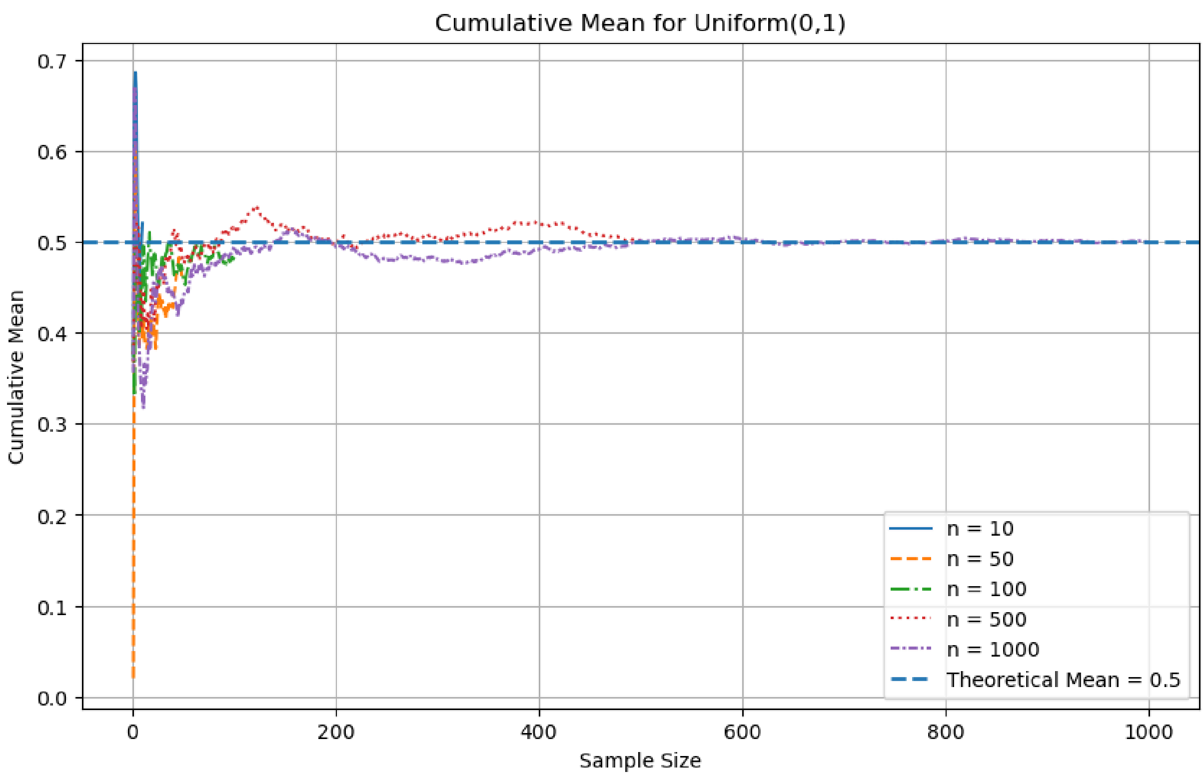
3. For each sample size, compute the sample mean.

In [4]:
```
uniform_means = {}
print("\nSample means for Uniform(0,1):")
for n, sample in uniform_samples.items():
    uniform_means[n] = sample.mean()
    print(f"For n = {n}, Sample Mean = {uniform_means[n]:.4f}")
```

```
Sample means for Uniform(0,1):
For n = 10, Sample Mean = 0.5201
For n = 50, Sample Mean = 0.4570
For n = 100, Sample Mean = 0.4829
For n = 500, Sample Mean = 0.5019
For n = 1000, Sample Mean = 0.4999
```

4. Plot the cumulative mean against sample size.

In [5]:
```
plt.figure(figsize=(10, 6))
line_styles = ['-', '--', '-.', ':', (0,(3,1,1,1))]
for (n, sample), style in zip(uniform_samples.items(), line_styles):
    cum_mean = np.cumsum(sample) / np.arange(1, n + 1)
    plt.plot(range(1, n + 1), cum_mean, linestyle=style, label=f"n = {n}")
plt.axhline(0.5, linestyle='--', linewidth=2, label="Theoretical Mean = 0.5")
plt.xlabel("Sample Size")
plt.ylabel("Cumulative Mean")
plt.title("Cumulative Mean for Uniform(0,1)")
plt.legend()
plt.grid(True)
plt.show()
```



5. Compare the empirical mean with the theoretical mean E(X) = 0.5 and verify the **Law of Large Numbers**.

In [6]:
```
print("As sample size increases, the sample mean approaches the theoretical mean 0.5, verifying the Law of Large Numbers.")
```

```
As sample size increases, the sample mean approaches the theoretical mean 0.5,
verifying the Law of Large Numbers.
```

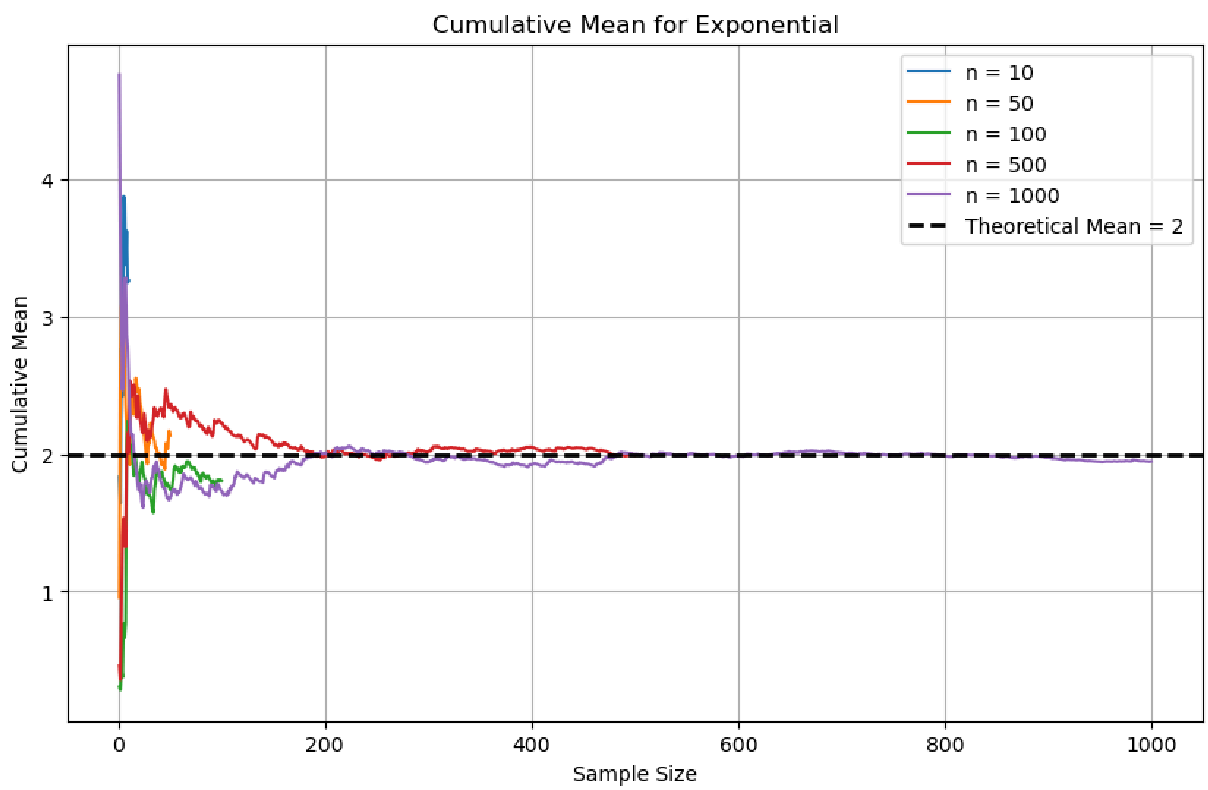
6. Repeat Steps 2-5 for the exponential distribution with mean mu = 2.

```
In [7]: exponential_samples = {}
        for n in n_values:
            exponential_samples[n] = np.random.exponential(scale = 2, size = n)
```

```
In [8]: exponential_means = {}
        print("Sample means for Exponential with mu = 2:")
        for n, sample1 in exponential_samples.items():
            exponential_means[n] = sample1.mean()
            print(f"For n = {n}, Sample Mean = {exponential_means[n]:.4f}")
```

```
Sample means for Exponential with mu = 2:
For n = 10, Sample Mean = 3.2658
For n = 50, Sample Mean = 2.1381
For n = 100, Sample Mean = 1.8071
For n = 500, Sample Mean = 1.9951
For n = 1000, Sample Mean = 1.9487
```

```
In [9]: plt.figure(figsize=(10, 6))
        line_styles1 = ['-', '--', '-.', ':', (0,(3,1,1,1))]
        for (n,sample1), style in zip(exponential_samples.items(), line_styles1):
            cum_mean_exp = np.cumsum(sample1) / np.arange(1, n + 1)
            plt.plot(range(1, n + 1), cum_mean_exp, label=f"n = {n}")
        plt.axhline(2, color="black", linestyle="--", linewidth=2, label="Theoretical Mean = 2")
        plt.xlabel("Sample Size")
        plt.ylabel("Cumulative Mean")
        plt.title("Cumulative Mean for Exponential")
        plt.legend()
        plt.grid(True)
        plt.show()
```



7. Comment on the effect of skewness on the convergence of the sample mean.

1. As the sample size increases, more observations are included, providing a better representation of the population distribution.
2. The sample mean varies considerably for small sample sizes but becomes more stable as the sample size increases.
3. The cumulative mean initially fluctuates but gradually stabilizes as the number of observations increases.
4. The empirical mean approaches the theoretical mean $E(X)=0.5$ as the sample size increases, thereby verifying the Law of Large Numbers.
5. For the exponential distribution, the sample mean converges to the theoretical mean $\mu=2$, but the convergence is slower compared to the uniform distribution.
6. Since the exponential distribution is positively skewed, the sample mean exhibits larger fluctuations initially and converges more slowly than in the symmetric uniform distribution.

Part B : Central Limit Theorem (CLT)

---

9. Simulate **1000 independent samples**, each of size n = 30, from **Gamma distribution** with shape 2 and scale 3.

In [10]:
```
# Parameters
num_samples = 1000
sample_size = 30

# Generating Gamma samples
gamma_samples = np.random.gamma(shape=2, scale=3, size=(num_samples, sample_size))
```

---

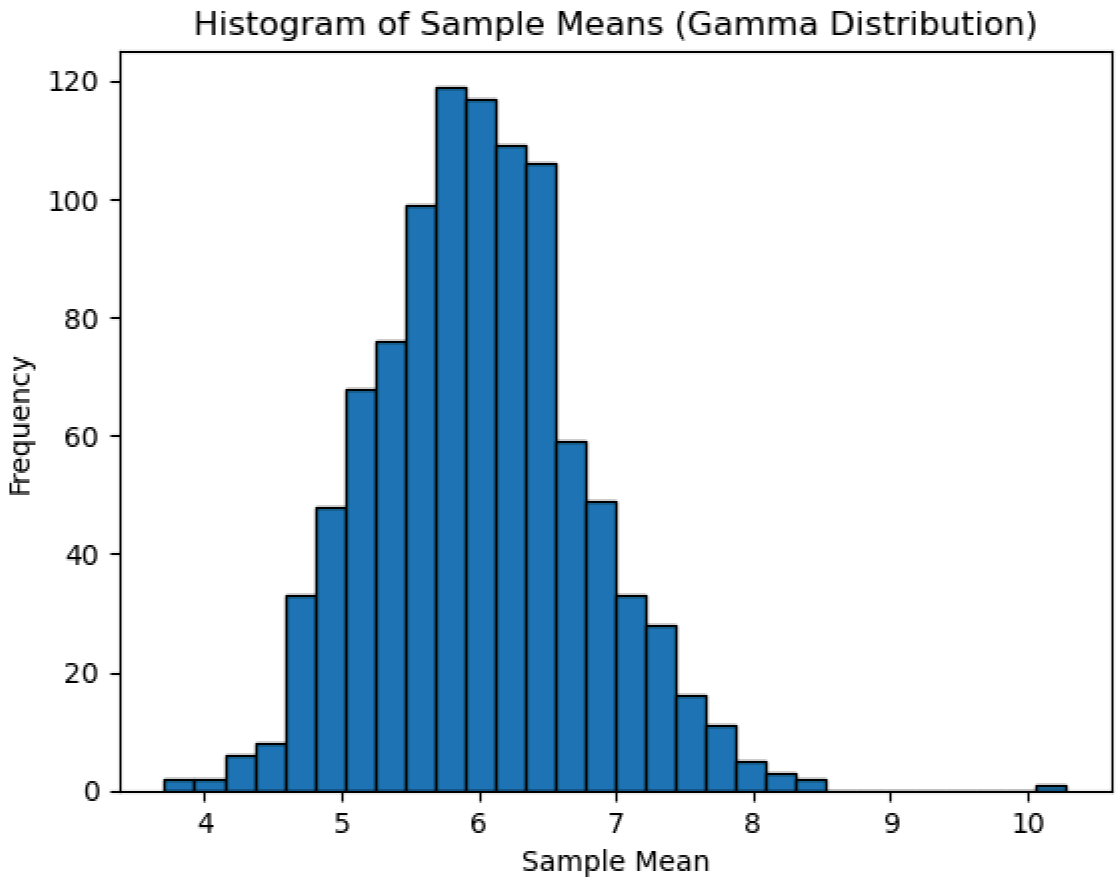
10. Compute the sample mean for each sample.

In [11]:
```
sample_means_gamma = np.mean(gamma_samples, axis=1)
```

---

11. Plot the histogram of the sample means and comment on its shape.

In [12]:
```
plt.figure()
plt.hist(sample_means_gamma, bins=30, edgecolor="black")
plt.xlabel("Sample Mean")
plt.ylabel("Frequency")
plt.title("Histogram of Sample Means (Gamma Distribution)")
plt.show()
```



- The histogram of the sample means is approximately **bell-shaped and symmetric**.
- Although the original **Gamma distribution is positively skewed**, the distribution of sample means tends toward a **normal distribution**.
- The center of the histogram is around **6**, which matches the theoretical mean:

$$\mu = \text{shape} \times \text{scale} = 2 \times 3 = 6$$

- The spread is smaller compared to the original Gamma distribution, showing **reduced variability** of sample means.

This confirms the **Central Limit Theorem**, which states that for a sufficiently large sample size ((n = 30)), the sampling distribution of the mean is approximately normal, regardless of the parent distribution's shape.