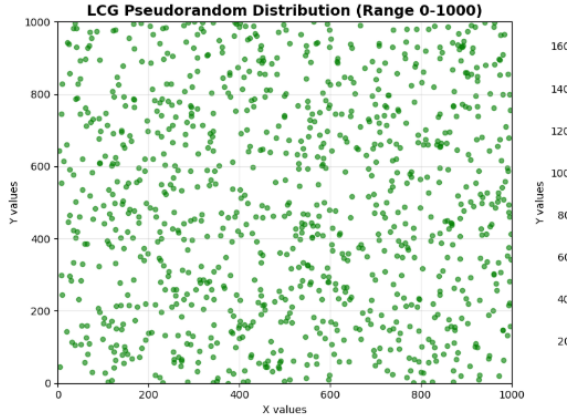
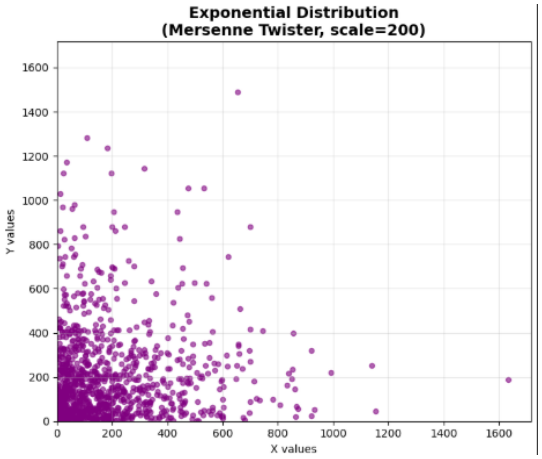
```
import numpy as np
import matplotlib.pyplot as plt

# Linear Congruential Generator (LCG) implementation with scaling
def lcg(n, seed=42, a=1664525, c=1013904223, m=2**32, scale=1000):
    numbers = []
    x = seed
    for _ in range(n):
        x = (a * x + c) % m
        numbers.append((x / m) * scale)  # Scale output to [0, scale)
    return np.array(numbers)

# Set parameters
n_points = 1000
scale_lcg = 1000

# Generate pseudorandom numbers using LCG scaled to [0, 1000)
lcg_x = lcg(n_points, seed=42, scale=scale_lcg)
lcg_y = lcg(n_points, seed=1234, scale=scale_lcg)

# Generate pseudorandom numbers using exponential distribution with bigger scale
np.random.seed(42)
exp_scale = 200
exp_x = np.random.exponential(scale=exp_scale, size=n_points)
exp_y = np.random.exponential(scale=exp_scale, size=n_points)

# Plot scatter plots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

ax1.scatter(lcg_x, lcg_y, alpha=0.6, s=20, color='green')
ax1.set_title('LCG Pseudorandom Distribution (Range 0-1000)', fontsize=14, fontweight='bold')
```

```
ax1.set_xlabel('X values')

ax1.set_ylabel('Y values')

ax1.grid(True, alpha=0.3)

ax1.set_xlim(0, scale_lcg)

ax1.set_ylim(0, scale_lcg)


ax2.scatter(exp_x, exp_y, alpha=0.6, s=20, color='purple')

ax2.set_title(f'Exponential Distribution\n(Mersenne Twister, scale={exp_scale})', fontsize=14,
fontweight='bold')

ax2.set_xlabel('X values')

ax2.set_ylabel('Y values')

ax2.grid(True, alpha=0.3)


# Adjust axis limits dynamically for exponential plots

max_exp = max(np.max(exp_x), np.max(exp_y))

ax2.set_xlim(0, max_exp * 1.05)

ax2.set_ylim(0, max_exp * 1.05)


plt.tight_layout()

plt.show()


# Print statistics

print("=== Distribution Comparison ===")

print(f"LCG - Mean X: {np.mean(lcg_x):.2f}, Std X: {np.std(lcg_x):.2f}")

print(f"LCG - Mean Y: {np.mean(lcg_y):.2f}, Std Y: {np.std(lcg_y):.2f}")

print(f"Exponential - Mean X: {np.mean(exp_x):.2f}, Std X: {np.std(exp_x):.2f}")

print(f"Exponential - Mean Y: {np.mean(exp_y):.2f}, Std Y: {np.std(exp_y):.2f}")


# Histograms

fig, ((ax3, ax4), (ax5, ax6)) = plt.subplots(2, 2, figsize=(12, 9))
```

```
ax3.hist(lcg_x, bins=30, alpha=0.7, color='green', edgecolor='black')
ax3.set_title('LCG X Distribution (0-1000)')
ax3.set_xlabel('Value')
ax3.set_ylabel('Frequency')

ax4.hist(lcg_y, bins=30, alpha=0.7, color='green', edgecolor='black')
ax4.set_title('LCG Y Distribution (0-1000)')
ax4.set_xlabel('Value')
ax4.set_ylabel('Frequency')

ax5.hist(exp_x, bins=30, alpha=0.7, color='purple', edgecolor='black')
ax5.set_title(f'Exponential X Distribution (scale={exp_scale})')
ax5.set_xlabel('Value')
ax5.set_ylabel('Frequency')

ax6.hist(exp_y, bins=30, alpha=0.7, color='purple', edgecolor='black')
ax6.set_title(f'Exponential Y Distribution (scale={exp_scale})')
ax6.set_xlabel('Value')
ax6.set_ylabel('Frequency')

plt.tight_layout()
plt.show()
```






```
=== Distribution Comparison ===
LCG - Mean X: 511.31, Std X: 285.03
LCG - Mean Y: 496.71, Std Y: 293.48
Exponential - Mean X: 194.50, Std X: 194.40
Exponential - Mean Y: 207.02, Std Y: 209.40
```