TASK 1: Born Rule for Measurement Probabilities

Aim: To compute measurement probabilities of quantum states using the Born rule.

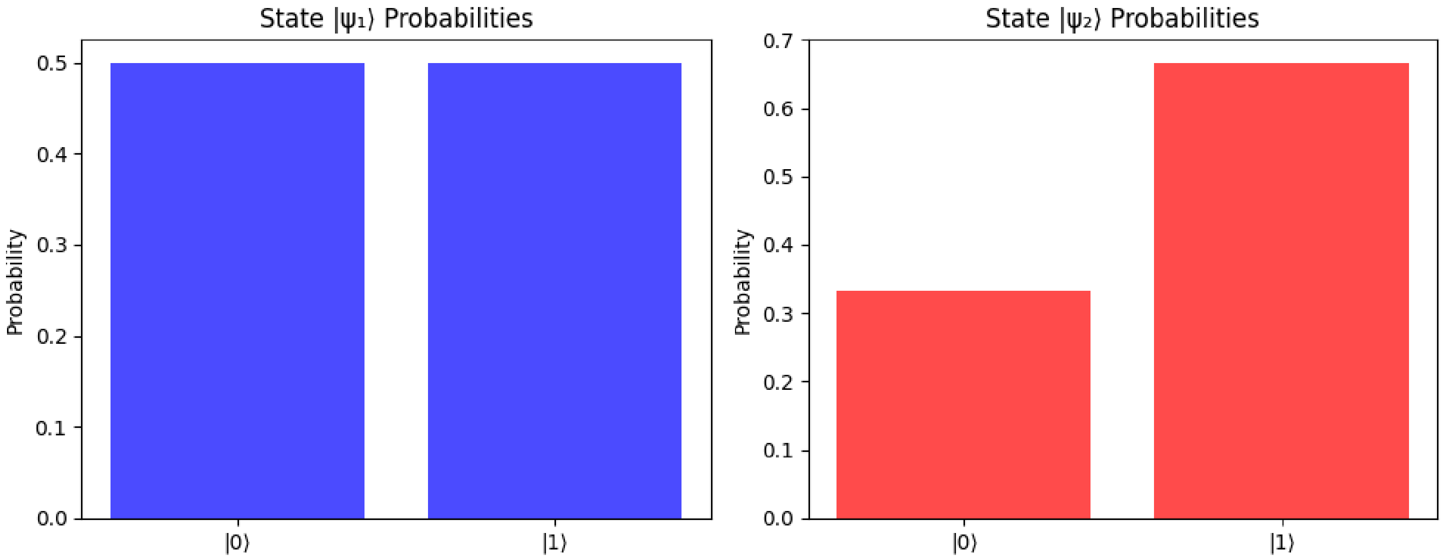
Algorithm:

1. Define quantum superposition states.
2. Apply Born rule to compute measurement probabilities.
3. Normalize probabilities.
4. Visualize results using bar charts.

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

print("\n" + "="*50)
print("TASK 1: BORN RULE - MEASUREMENT PROBABILITIES")
print("="*50)

def born_rule_probabilities(psi):
    """Calculate measurement probabilities using Born rule: P = |<basis|psi>|^2"""
    probabilities = np.abs(psi)**2
    return probabilities / np.sum(probabilities) # Normalize

# Create superposition states
psi_1 = np.array([1/np.sqrt(2), 1/np.sqrt(2)]) # |+⟩ state
psi_2 = np.array([1/np.sqrt(3), np.sqrt(2/3)]) # Custom superposition

print("Superposition state 1: |ψ₁⟩ =", psi_1)
print("Measurement probabilities:", born_rule_probabilities(psi_1))

print("Superposition state 2: |ψ₂⟩ =", psi_2)
print("Measurement probabilities:", born_rule_probabilities(psi_2))

# Visualization
states = ['|0⟩', '|1⟩']
probs_1 = born_rule_probabilities(psi_1)
probs_2 = born_rule_probabilities(psi_2)

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.bar(states, probs_1, color='blue', alpha=0.7)
plt.title('State |ψ₁⟩ Probabilities')
plt.ylabel('Probability')

plt.subplot(1, 2, 2)
plt.bar(states, probs_2, color='red', alpha=0.7)
```

```
plt.title('State |ψ₂⟩ Probabilities')
plt.ylabel('Probability')

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

```
==================================================
TASK 1: BORN RULE - MEASUREMENT PROBABILITIES
==================================================
Superposition state 1: |ψ₁⟩ = [0.70710678 0.70710678]
Measurement probabilities: [0.5 0.5]
Superposition state 2: |ψ₂⟩ = [0.57735027 0.81649658]
Measurement probabilities: [0.33333333 0.66666667]
```



Result:

These results validate the Born Rule's Fundamental role in predicting measurement statistics in quantum mechanics.