Question 1:

```
from qiskit import QuantumCircuit
from qiskit.visualization import plot_histogram

!pip install qiskit
import qiskit

Requirement already satisfied: qiskit in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (1.2.4)
Requirement already satisfied: rustworkx>=0.15.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (0.15.1)
Requirement already satisfied: numpy<3,>=1.17 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (1.26.4)
Requirement already satisfied: scipy>=1.5 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (1.13.0)
Requirement already satisfied: sympy>=1.3 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (1.13.3)
Requirement already satisfied: dill>=0.3 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from qiskit) (2.9.0.post0)
Requirement already satisfied: stevedore>=3.0.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (5.3.0)
Requirement already satisfied: typing-extensions in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from qiskit) (4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit) (0.13.0)
Requirement already satisfied: six>=1.5 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.8.0->qiskit) (1.16.0)
Requirement already satisfied: pbr>=2.0.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from stevedore>=3.0.0->qiskit) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from sympy>=1.3->qiskit) (1.3.0)

!pip install qiskit-aer
```

```
Requirement already satisfied: qiskit-aer in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (0.15.1)
Requirement already satisfied: qiskit>=1.1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.2.4)
Requirement already satisfied: numpy>=1.16.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.26.4)
Requirement already satisfied: scipy>=1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.13.0)
Requirement already satisfied: psutil>=5 in c:\users\geetapriya\
appdata\roaming\python\python311\site-packages (from qiskit-aer)
(5.9.8)
Requirement already satisfied: rustworkx>=0.15.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (0.15.1)
Requirement already satisfied: sympy>=1.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (1.13.3)
Requirement already satisfied: dill>=0.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in c:\users\
geetapriya\appdata\roaming\python\python311\site-packages (from
qiskit>=1.1.0->qiskit-aer) (2.9.0.post0)
Requirement already satisfied: stevedore>=3.0.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (5.3.0)
Requirement already satisfied: typing-extensions in c:\users\
geetapriya\appdata\roaming\python\python311\site-packages (from
qiskit>=1.1.0->qiskit-aer) (4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (0.13.0)
Requirement already satisfied: six>=1.5 in c:\users\geetapriya\
appdata\roaming\python\python311\site-packages (from python-
dateutil>=2.8.0->qiskit>=1.1.0->qiskit-aer) (1.16.0)
Requirement already satisfied: pbr>=2.0.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
stevedore>=3.0.0->qiskit>=1.1.0->qiskit-aer) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from sympy>=1.3->qiskit>=1.1.0->qiskit-aer) (1.3.0)

# Use Aer's qasm_simulator
from qiskit_aer import Aer

pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.21 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install pylatexenc
```

```
Requirement already satisfied: pylatexenc in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (2.10)
Note: you may need to restart the kernel to use updated packages.
```

```python
from qiskit import QuantumCircuit,transpile
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# Create a Quantum Circuit with 3 qubits and 3 classical bits
qc = QuantumCircuit(3, 3)

# Apply a Hadamard gate on qubit 0
qc.h(0)
```

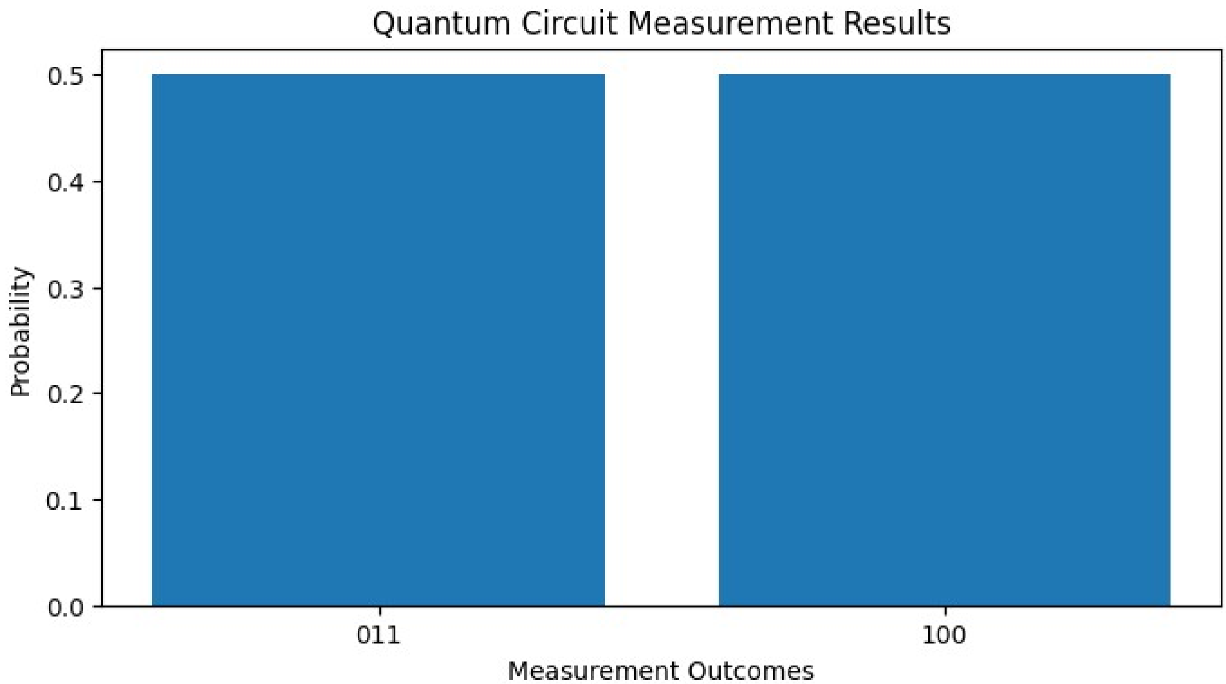
```
# Apply a Toffoli (CCX) gate with controls on qubits 0 and 1, and
target on qubit 2
qc.ccx(0, 1, 2)

# Draw the circuit (text-based representation)
print(qc.draw('text'))

# Add measurement instructions to measure the qubits
qc.measure([0, 1, 2], [0, 1, 2])  # Measure qubits 0, 1, and 2 into
classical bits 0, 1, and 2

# Transpile the circuit for the simulator (optional, for optimization)
qc_transpiled = transpile(qc)

# Use the Aer simulator to get measurement results
simulator = Aer.get_backend('qasm_simulator')

# Execute the quantum circuit on the simulator
job = simulator.run(qc_transpiled, shots=1024)
result = job.result()
```

```
     ┌───┐     
q_0: ┤ H ├──■──
     └───┘  │  
q_1: ───────■──
          ┌─┴─┐
q_2: ─────┤ X ├
          └───┘
c: 3/══════════
```

```
from qiskit.primitives import Sampler

# Measure each qubit into the corresponding classical bit
qc2.measure([0, 1, 2], [0, 1, 2])

# Use Sampler to get measurement results
sampler = Sampler()
results = sampler.run(qc2).result()
statistics = results.quasi_dists[0].binary_probabilities()

# Plot the histogram of results
plt.figure(figsize=(8, 4))
plt.title("Quantum Circuit Measurement Results")
plt.bar(statistics.keys(), statistics.values())
plt.xlabel("Measurement Outcomes")
plt.ylabel("Probability")
plt.show()
```

```
C:\Users\GeetaPriya\AppData\Local\Temp\ipykernel_9416\4281708371.py:5:
DeprecationWarning: The class ``qiskit.primitives.sampler.Sampler`` is
```

```
deprecated as of qiskit 1.2. It will be removed no earlier than 3
months after the release date. All implementations of the
`BaseSamplerV1` interface have been deprecated in favor of their V2
counterparts. The V2 alternative for the `Sampler` class is
`StatevectorSampler`.
  sampler = Sampler()
```



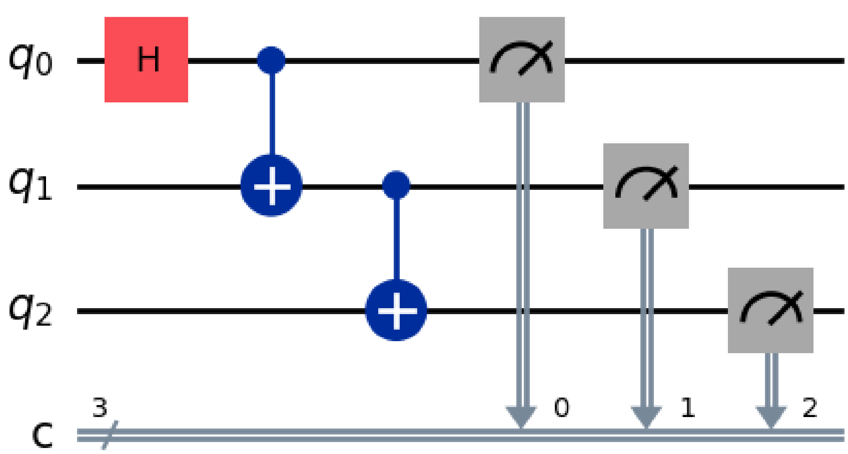

```
# Execute the quantum circuit on the simulator
job = simulator.run(qc_transpiled, shots=1024)
result = job.result()

# Get the counts of measurement results
counts = result.get_counts(qc)
print("Measurement Results:", counts)

Measurement Results: {'000': 532, '001': 492}
```

With different inputs:

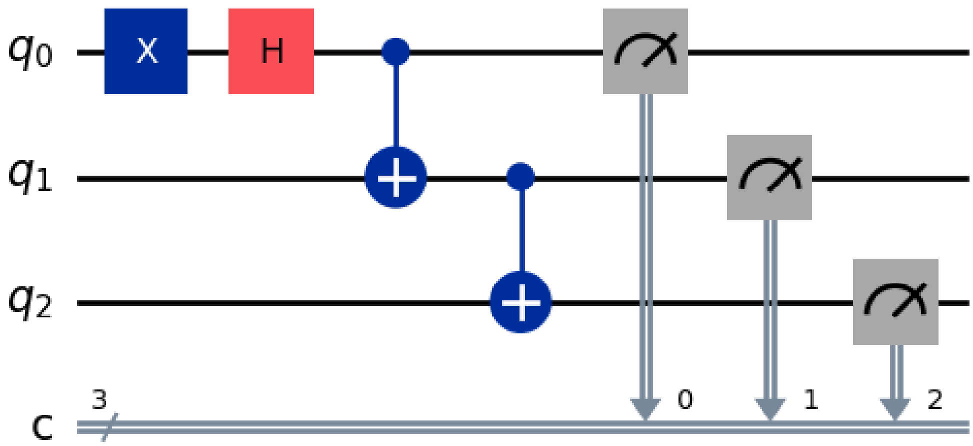
```
from qiskit import QuantumCircuit

def HSHT_circuit(initial_state='0'):
    # Create a quantum circuit with 3 qubits and 3 classical bits
    qc = QuantumCircuit(3, 3)

    # Set the initial state of qubit 0 if specified as '1'
    if initial_state == '1':
```

```
        qc.x(0)  # Apply X gate to flip qubit 0 to |1>

    # Apply Hadamard gate on qubit 0
    qc.h(0)

    # Apply CNOT gates as per the original circuit
    qc.cx(0, 1)
    qc.cx(1, 2)

    # Measure all qubits
    qc.measure([0, 1, 2], [0, 1, 2])

    return qc

# Set up the AerSimulator
simulator = AerSimulator()

# Create circuits for the two initial states |0> and |1>
qc0 = HSHT_circuit('0')
qc1 = HSHT_circuit('1')

# Display the circuits
print("Circuit for input |0⟩:")
display(qc0.draw(output='mpl'))
print("Circuit for input |1⟩:")
display(qc1.draw(output='mpl'))
```

```
Circuit for input |0⟩:
```




```
Circuit for input |1⟩:
```



Question 2 :

```
!pip install qiskit-aer

^C

Requirement already satisfied: qiskit-aer in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (0.15.1)
Requirement already satisfied: qiskit>=1.1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.2.4)
Requirement already satisfied: numpy>=1.16.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.26.4)
Requirement already satisfied: scipy>=1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.13.0)
Requirement already satisfied: psutil>=5 in c:\users\geetapriya\
appdata\roaming\python\python311\site-packages (from qiskit-aer)
(5.9.8)
Requirement already satisfied: rustworkx>=0.15.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (0.15.1)
Requirement already satisfied: sympy>=1.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (1.13.3)
Requirement already satisfied: dill>=0.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in c:\users\
geetapriya\appdata\roaming\python\python311\site-packages (from
qiskit>=1.1.0->qiskit-aer) (2.9.0.post0)
```

```
Requirement already satisfied: stevedore>=3.0.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit>=1.1.0->qiskit-aer) (5.3.0)
Requirement already satisfied: typing-extensions in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from qiskit>=1.1.0->qiskit-aer) (4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from qiskit>=1.1.0->qiskit-aer) (0.13.0)
Requirement already satisfied: six>=1.5 in c:\users\geetapriya\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.8.0->qiskit>=1.1.0->qiskit-aer) (1.16.0)
Requirement already satisfied: pbr>=2.0.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from stevedore>=3.0.0->qiskit>=1.1.0->qiskit-aer) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\geetapriya\appdata\local\programs\python\python311\lib\site-packages (from sympy>=1.3->qiskit>=1.1.0->qiskit-aer) (1.3.0)
```

```python
# Use Aer's qasm_simulator
from qiskit_aer import Aer

from qiskit import QuantumCircuit, transpile
from qiskit import transpile
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

from qiskit import QuantumCircuit
from qiskit.circuit.library import XGate
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# Create a Quantum Circuit with 3 qubits
qc2 = QuantumCircuit(3, 3)

# Apply a Hadamard gate on qubit q0
qc2.h(0)

# Apply an X (NOT) gate on qubit q2
qc2.x(2)

# Apply a controlled-SWAP (Fredkin gate) with q0 as control and q1, q2 as target qubits
qc2.cswap(0, 1, 2)

# Draw the circuit
qc2.draw('text')
```

```
     ┌───┐
q_0: ┤ H ├──■─
```

```
     └───┘  │
q_1: ───────X─
     ┌───┐  │
q_2: ┤ X ├──X─
     └───┘
c: 3/═════════
```

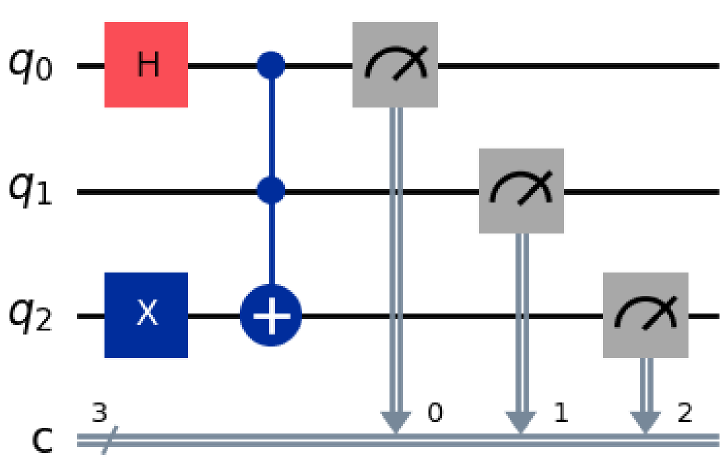
With different inputs:

```python
# Create a quantum circuit with 3 qubits and 3 classical bits
qc = QuantumCircuit(3, 3)

# Apply Hadamard gate on qubit 0
qc.h(0)

# Apply X gate on qubit 2
qc.x(2)

# Apply Toffoli (CCX) gate with q0 and q1 as control and q2 as target
qc.ccx(0, 1, 2)

# Measure all qubits
qc.measure([0, 1, 2], [0, 1, 2])
```

```
<qiskit.circuit.instructionset.InstructionSet at 0x238fc653550>
```

```python
# Display the circuit
print("Quantum Circuit:")
display(qc.draw(output='mpl'))
```

```
Quantum Circuit:
```



```python
# Execute the quantum circuit on the simulator
job = simulator.run(qc_transpiled, shots=1024)
result = job.result()

# Get the counts of measurement results
counts = result.get_counts(qc)
print("Measurement Results:", counts)
```

```
Measurement Results: {'10': 1024}
```

Question 3:

Part 1:

```
!pip install qiskit-aer
```

```
Requirement already satisfied: qiskit-aer in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (0.15.1)
Requirement already satisfied: qiskit>=1.1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.2.4)
Requirement already satisfied: numpy>=1.16.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.26.4)
Requirement already satisfied: scipy>=1.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit-aer) (1.13.0)
Requirement already satisfied: psutil>=5 in c:\users\geetapriya\
appdata\roaming\python\python311\site-packages (from qiskit-aer)
(5.9.8)
Requirement already satisfied: rustworkx>=0.15.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (0.15.1)
Requirement already satisfied: sympy>=1.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (1.13.3)
Requirement already satisfied: dill>=0.3 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
qiskit>=1.1.0->qiskit-aer) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in c:\users\
geetapriya\appdata\roaming\python\python311\site-packages (from
qiskit>=1.1.0->qiskit-aer) (2.9.0.post0)
Requirement already satisfied: stevedore>=3.0.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (5.3.0)
Requirement already satisfied: typing-extensions in c:\users\
geetapriya\appdata\roaming\python\python311\site-packages (from
qiskit>=1.1.0->qiskit-aer) (4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in c:\users\
```

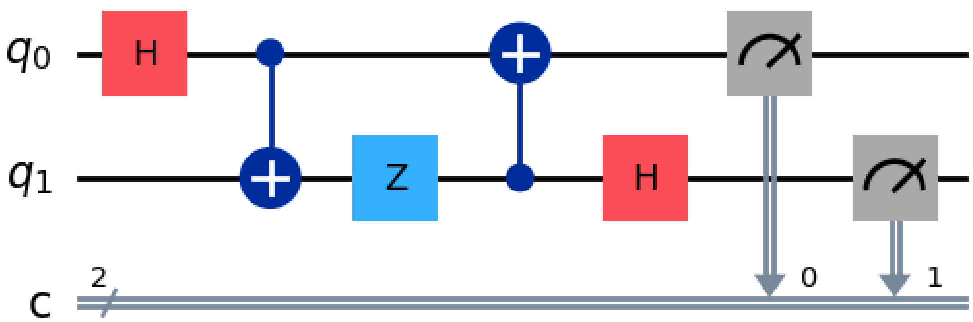
```
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from qiskit>=1.1.0->qiskit-aer) (0.13.0)
Requirement already satisfied: six>=1.5 in c:\users\geetapriya\
appdata\roaming\python\python311\site-packages (from python-
dateutil>=2.8.0->qiskit>=1.1.0->qiskit-aer) (1.16.0)
Requirement already satisfied: pbr>=2.0.0 in c:\users\geetapriya\
appdata\local\programs\python\python311\lib\site-packages (from
stevedore>=3.0.0->qiskit>=1.1.0->qiskit-aer) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\
geetapriya\appdata\local\programs\python\python311\lib\site-packages
(from sympy>=1.3->qiskit>=1.1.0->qiskit-aer) (1.3.0)
```

```python
from qiskit_aer  import Aer

from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_bloch_multivector
import matplotlib.pyplot as plt

from qiskit import QuantumCircuit, transpile
from qiskit import transpile
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# Create a quantum circuit with 2 qubits and 2 classical bits
qc = QuantumCircuit(2, 2)

# Apply Hadamard gate on qubit 0
qc.h(0)

# Apply CNOT gate with q0 as control and q1 as target
qc.cx(0, 1)

# Apply Z gate on qubit 1
qc.z(1)

# Apply another CNOT gate with q1 as control and q0 as target
qc.cx(1, 0)

# Apply Hadamard gate on qubit 1
qc.h(1)

# Measure both qubits
qc.measure([0, 1], [0, 1])
```

```
<qiskit.circuit.instructionset.InstructionSet at 0x1bff3c8e890>
```

```python
# Display the circuit
print("Quantum Circuit:")
display(qc.draw(output='mpl'))
```

```
Quantum Circuit:
```



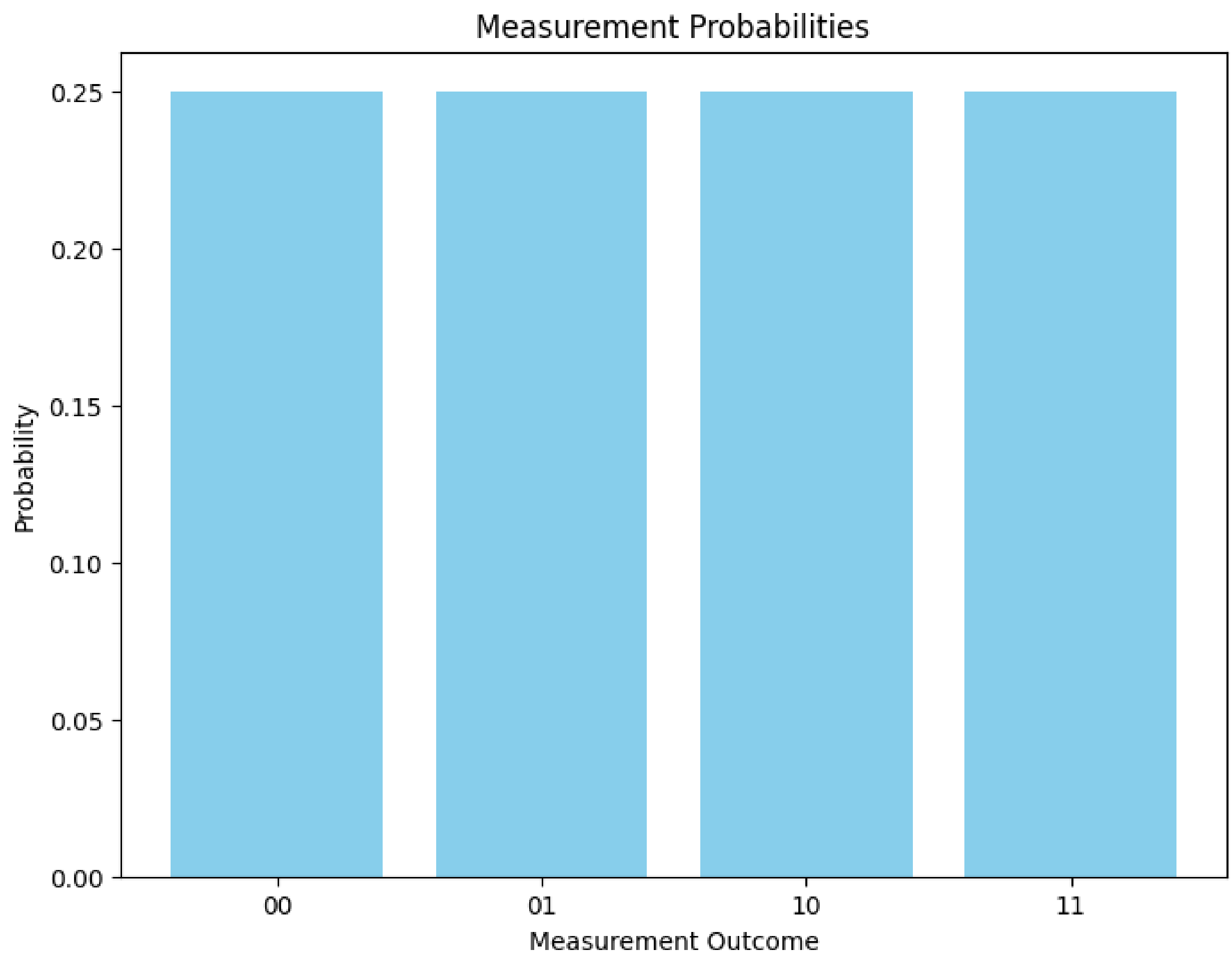

```python
# Create a Quantum Circuit with 2 qubits
qc = QuantumCircuit(2)

# Step 1: Apply Hadamard gate on the first qubit
qc.h(0)

# Step 2: Apply CNOT gate with the first qubit as control and the
second qubit as target
qc.cx(0, 1)

# Step 3: Apply Z gate on the second qubit
qc.z(1)

# Step 4: Apply another CNOT gate with the first qubit as control and
the second as target
qc.cx(0, 1)

# Step 5: Apply Hadamard gate on the second qubit
qc.h(1)

# Draw the circuit
qc.draw('text')
```

```
     ┌───┐
q_0: ┤ H ├──■─────────■───────
     └───┘┌─┴─┐┌───┐┌─┴─┐┌───┐
q_1: ─────┤ X ├┤ Z ├┤ X ├┤ H ├
          └───┘└───┘└───┘└───┘
```

```python
# Initialize the statevector in the |01⟩ state directly by applying X
to qubit 1
state = Statevector.from_label('01')
state = state.evolve(qc)  # Evolve the state through the circuit

# Display the final state vector
print("Final state vector:", state)

# Plot the Bloch vector for each qubit
bloch_plot = plot_bloch_multivector(state)
plt.show(bloch_plot)  # Display the Bloch sphere plot

# Measure probabilities directly from the statevector
probabilities = state.probabilities_dict()
print("Measurement probabilities:", probabilities)

# Convert probabilities dictionary for plotting
bitstrings = list(probabilities.keys())
values = list(probabilities.values())
```

```
Final state vector: Statevector([0.5+0.j, 0.5+0.j, 0.5+0.j, 0.5+0.j],
            dims=(2, 2))
Measurement probabilities: {'00': 0.2499999999999999, '01':
0.2499999999999999, '10': 0.2499999999999999, '11':
0.2499999999999999}
```

```python
# Plot a custom bar chart for measurement probabilities
plt.figure(figsize=(8, 6))
plt.bar(bitstrings, values, color='skyblue')
plt.xlabel("Measurement Outcome")
plt.ylabel("Probability")
plt.title("Measurement Probabilities")
plt.show()
```



b. Proof that the order of unitary compositions is crucial in quantum operations considering the combination of Hadamard. Phase gate and T- gate

In quantum mechanics, the order of operations (unitary transformations) is crucial due to the non-commutative nature of quantum gates. To illustrate this, we can consider a few quantum gates: the Hadamard (H) gate, the Phase (S) gate, and the T gate. We'll analyze how the outcome of applying these gates varies based on the order in which they are applied to a qubit.

Definitions of Gates Hadamard Gate (H): The Hadamard gate creates superposition. For a single qubit, it transforms: [H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{and} \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)]

Phase Gate (S): The Phase gate applies a phase of (\frac{\pi}{2}) to the state: [S|0\rangle = |0\rangle \quad \text{and} \quad S|1\rangle = i|1\rangle]

T Gate: The T gate adds a phase of (\frac{\pi}{4}): [T|0\rangle = |0\rangle \quad \text{and} \quad T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)]

If we analyze a different order or add a measurement step in between, the non-commutativity will become apparent:

Instead, apply (T) and (H) in different sequences with measurement, a clear distinction in states