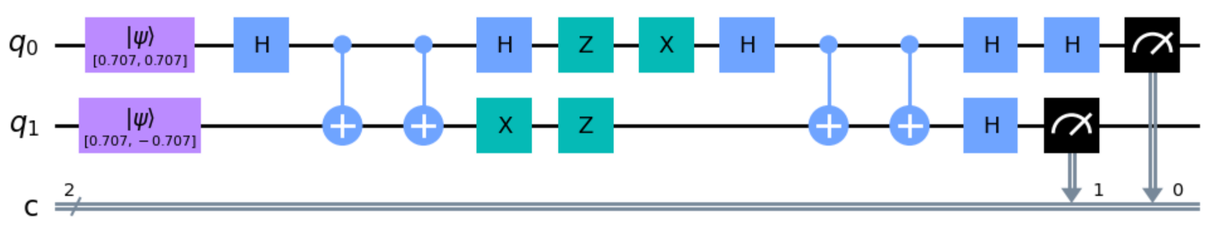
In [6]:
```
import qiskit
from qiskit import *
import numpy as np
from qiskit_ibm_provider import IBMProvider
```

The below code is for the 5th Questions, the answer is "10" state with 100% prob. {Note qiskit follows "little indian notation so "01" is equivalent to "10"}

In [14]:
```
qc = QuantumCircuit(2,2)
qc.initialize([1/np.sqrt(2), 1/np.sqrt(2)],0)
qc.initialize([1/np.sqrt(2), -1/np.sqrt(2)],1)
qc.h(0)
qc.cx(0,1)
qc.cx(0,1)
qc.h(0)
qc.z(0)
qc.x(1)
qc.x(0)
qc.z(1)
qc.h(0)
qc.cx(0,1)
qc.cx(0,1)
qc.h(0)
qc.h(1)
qc.h(0)
qc.measure([0,1],[0,1])
qc.draw('mpl')
```

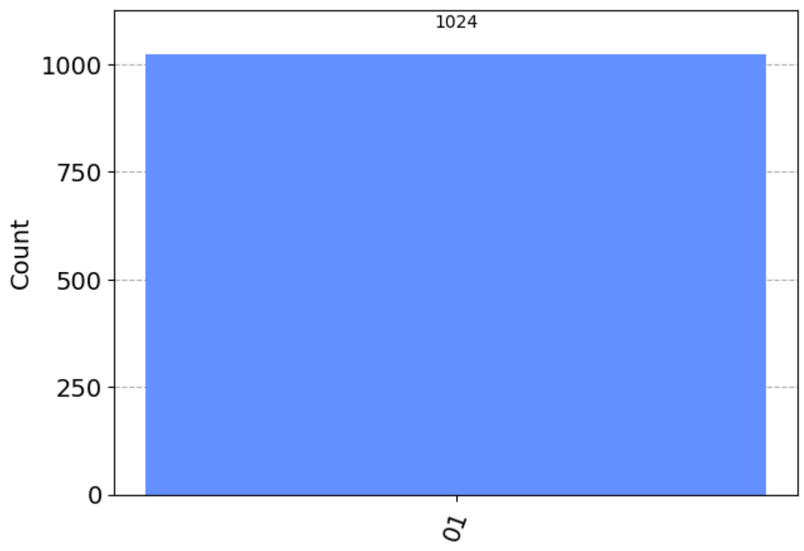
Out[14]:



In [18]:
```
backend = Aer.get_backend('qasm_simulator')
job = execute(qc,backend,shots=1024)
counts = job.result().get_counts()
```

In [20]:
```
from qiskit.visualization import plot_histogram, plot_state_qsphere
plot_histogram(counts)
```

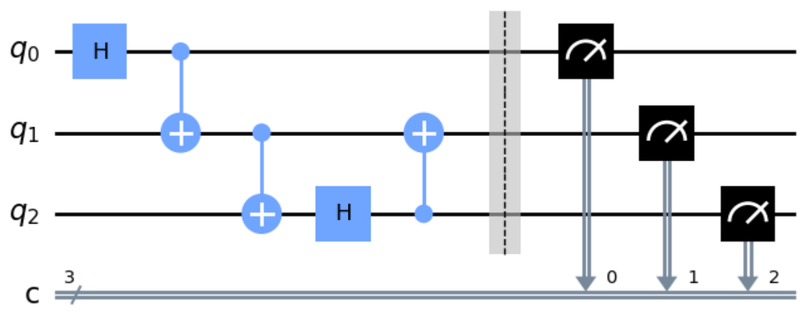
Out[20]:



The below code is for the 2th_b_part Questions, the answer is shown in the below graph with 25% prob each.

In [24]:
```
qc_1 = QuantumCircuit(3,3)
qc_1.h(0)
qc_1.cx(0,1)
qc_1.cx(1,2)
qc_1.h(2)
qc_1.cx(2,1)
qc_1.barrier()
qc_1.measure([0,1,2],[0,1,2])
qc_1.draw('mpl')
```

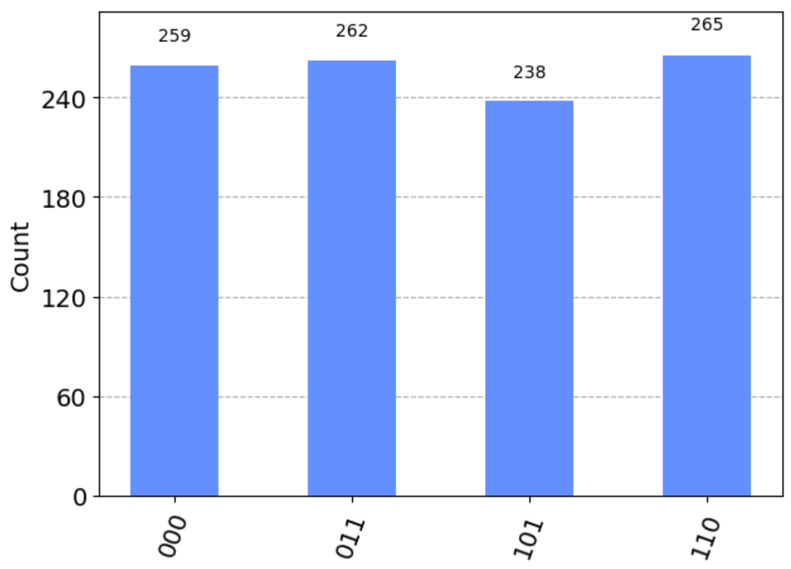
Out[24]:



In [26]:
```
provider = IBMProvider()
backend = provider.get_backend('ibmq_qasm_simulator')
job = execute(qc_1,backend,shots=1024)
counts = job.result().get_counts()
```

In [27]:
```
plot_histogram(counts)
```

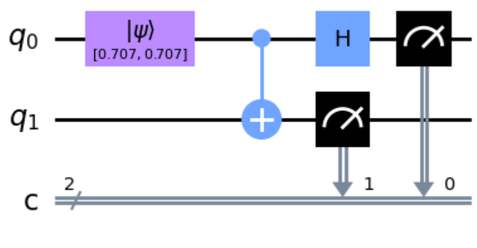
Out[27]:



The below code is for the 2th_a_part Questions, the answer is shown in the below graph with 25% prob each.

In [39]:
```
qc_2 = QuantumCircuit(2,2)
qc_2.initialize([1/np.sqrt(2), 1/np.sqrt(2)],0) #It is just a "Hadamard gate{H}"
qc_2.cx(0,1)
qc_2.h(0)
qc_2.measure([0,1],[0,1])
qc_2.draw('mpl')
```

Out[39]:



In [40]:
```
backend = Aer.get_backend('qasm_simulator')
job = execute(qc_2,backend,shots=1024)
counts = job.result().get_counts()
```

In [41]:
```
plot_histogram(counts)
```

Out[41]: