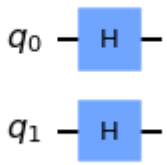
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, Aer, IBMQ, assemble
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from ibm_quantum_widgets import *
from qiskit.providers.ibmq import least_busy

# Loading your IBM Quantum account(s)
provider = IBMQ.load_account()
```

In [2]:
```
n = 2
grover_circuit = QuantumCircuit(n)
```

In [3]:
```
def initialize_s(qc, qubits):
    """Apply a H-gate to 'qubits' in qc"""
    for q in qubits:
        qc.h(q)
    return qc
```

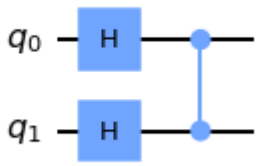
In [4]:
```
grover_circuit = initialize_s(grover_circuit, [0,1])
grover_circuit.draw()
```

Out[4]:



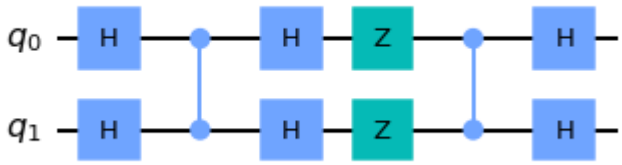
In [5]:
```
grover_circuit.cz(0,1) # Oracle
grover_circuit.draw()
```

Out[5]:



In [6]:
```
# Diffusion operator (U_s)
grover_circuit.h([0,1])
grover_circuit.z([0,1])
grover_circuit.cz(0,1)
grover_circuit.h([0,1])
grover_circuit.draw()
```

Out[6]:

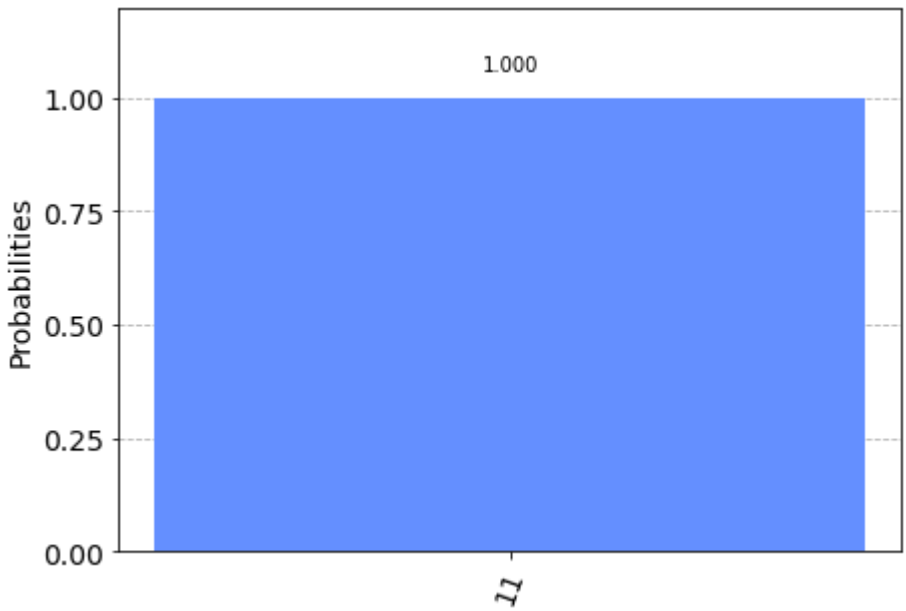


In [7]:
```
sim = Aer.get_backend('aer_simulator')
# we need to make a copy of the circuit with the 'save_statevector'
# instruction to run on the Aer simulator
grover_circuit_sim = grover_circuit.copy()
grover_circuit_sim.save_statevector()
qobj = assemble(grover_circuit_sim)
result = sim.run(qobj).result()
statevec = result.get_statevector()
from qiskit_textbook.tools import vector2latex
vector2latex(statevec, pretext="|\\psi\\rangle =")
```

$$|\psi\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

In [8]:
```
grover_circuit.measure_all()

aer_sim = Aer.get_backend('aer_simulator')
qobj = assemble(grover_circuit)
result = aer_sim.run(qobj).result()
counts = result.get_counts()
plot_histogram(counts)
```

Out[8]:



In [9]:
```
# Load IBM Q account and get the least busy backend device
provider = IBMQ.load_account()
provider = IBMQ.get_provider("ibm-q")
device = least_busy(provider.backends(filters=lambda x: x.configuration
().n_qubits >= 3 and
                                      not x.configuration().simulator and
x.status().operational==True))
print("Running on current least busy device: ", device)
```

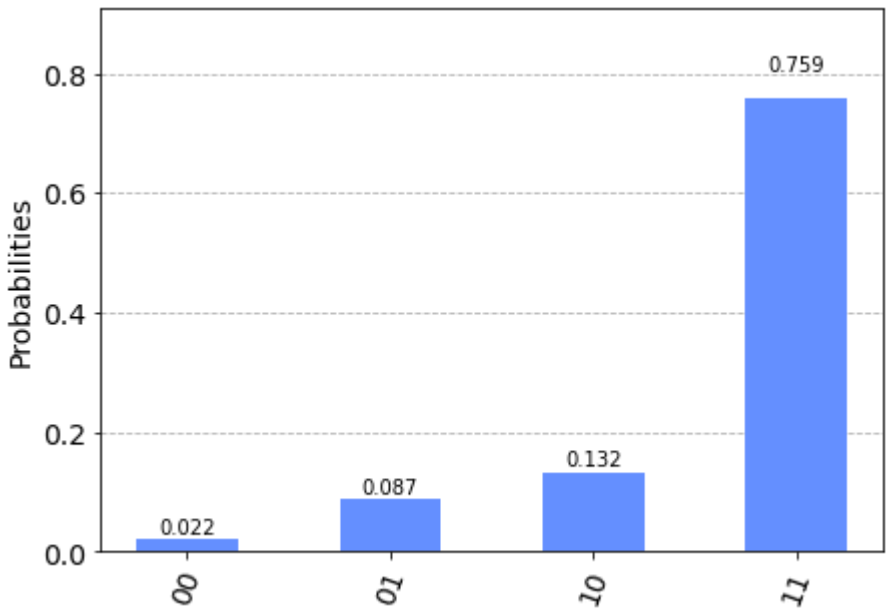
```
ibmqfactory.load_account:WARNING:2021-07-14 07:28:32,133: Credentials
are already in use. The existing account in the session will be repla
ced.

Running on current least busy device:  ibmqx2
```

In [10]:
```
# Run our circuit on the least busy backend. Monitor the execution of t
he job in the queue
from qiskit.tools.monitor import job_monitor
transpiled_grover_circuit = transpile(grover_circuit, device, optimizat
ion_level=3)
job = device.run(transpiled_grover_circuit)
job_monitor(job, interval=2)
```

```
Job Status: job has successfully run
```

In [11]:
```
# Get the results from the computation
results = job.result()
answer = results.get_counts(grover_circuit)
plot_histogram(answer)
```

Out[11]:



In [ ]:
```
This code is a part of Qiskit
© Copyright IBM 2017, 2020.

This code is licensed under the Apache License, Version 2.0. You may ob
tain a copy of this license in the LICENSE.txt file in the root directo
ry of this source tree or at http://www.apache.org/licenses/LICENSE-2.
0.
Any modifications or derivative works of this code must retain this cop
yright notice, and modified files need to carry a notice indicating tha
t they have been altered from the originals.

#Program executed by Bhadale IT in IBM Quantum Lab (https://www.bhadale
it.com).
#For more details on the Qiskit code and tutorials visit https://qiski
t.org/ website
```