In [39]:
```
from qiskit import QuantumRegister, ClassicalRegister
from qiskit import QuantumCircuit, execute,IBMQ
from qiskit.tools.monitor import job_monitor
from qiskit.circuit.library import QFT
from qiskit.visualization import circuit_drawer
from qiskit.circuit.library.standard_gates import CU1Gate
import numpy as np
```

In [40]:
```
pi = np.pi
```

In [41]:
```
q = QuantumRegister(5,'q')
c = ClassicalRegister(5,'c')
```

In [42]:
```
for i in range(0, 3):
  circuit = QuantumCircuit(q,c)
```

In [43]:
```
circuit.x(q[4])
circuit.x(q[2])
circuit.x(q[0])
```

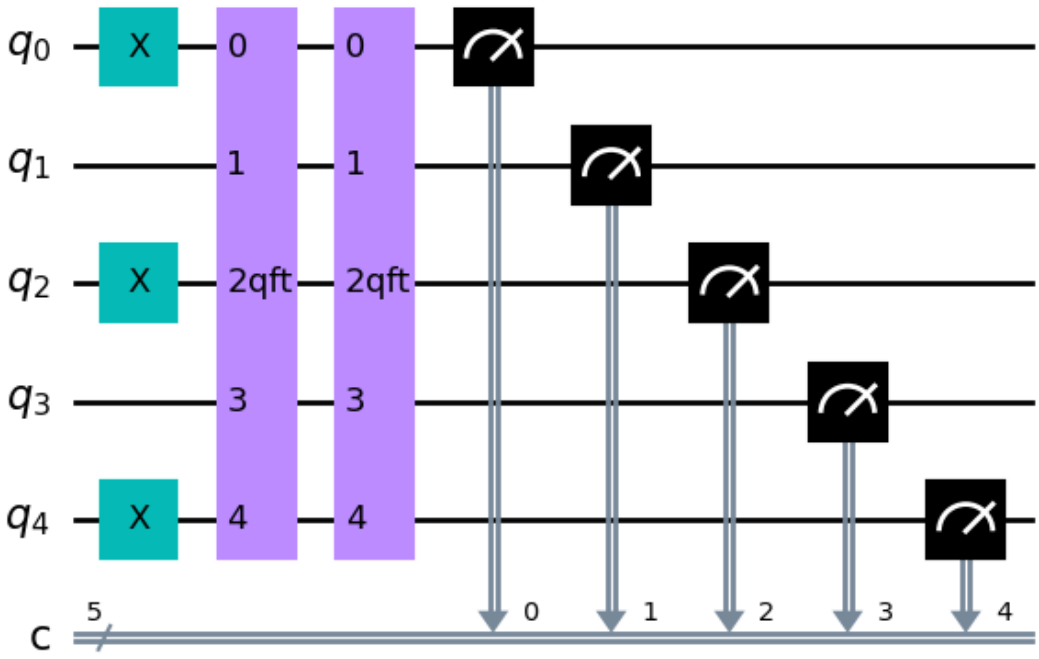
Out[43]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7f711d387b20>
```

In [44]:
```
num_qubits = 5
circuit.append(QFT(num_qubits=num_qubits, approximation_degree=0, do_swaps=True,
circuit.append(QFT(num_qubits=num_qubits, approximation_degree=0, do_swaps=True,
```

Out[44]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7f711bead9f0>
```

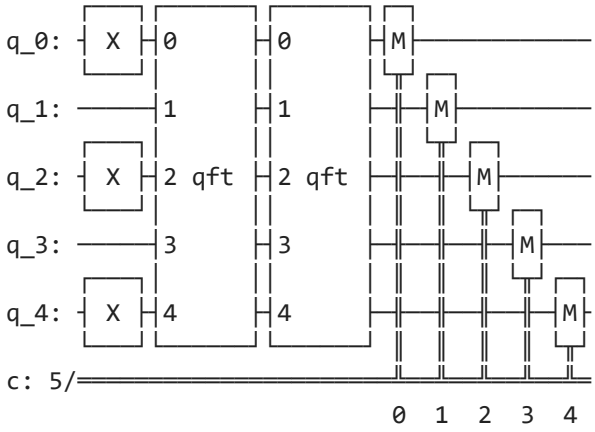
In [45]:
```
circuit.measure(q,c)
circuit.draw(output='mpl',filename='qft2.png')
```

Out[45]:

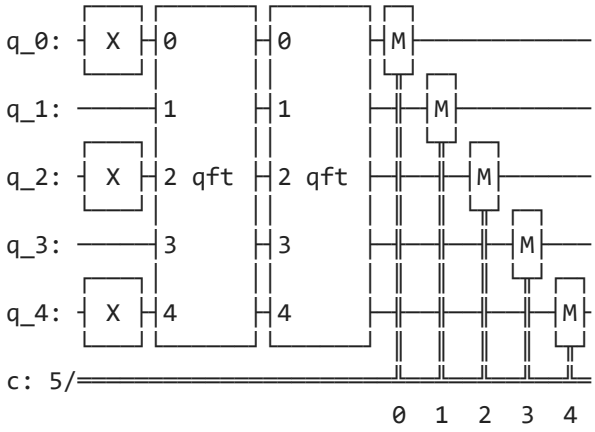


In [46]:
```
print("QFT Circuit:")
print(circuit_drawer(circuit, output='text'))
```

```
QFT Circuit:
```



In [47]:
```
print(circuit)
```



In [48]:
```
from qiskit_aer import AerSimulator
from qiskit import transpile
```

In [49]:
```
backend = AerSimulator()
```

In [50]:
```
qc_compiled = transpile(circuit, backend)
```

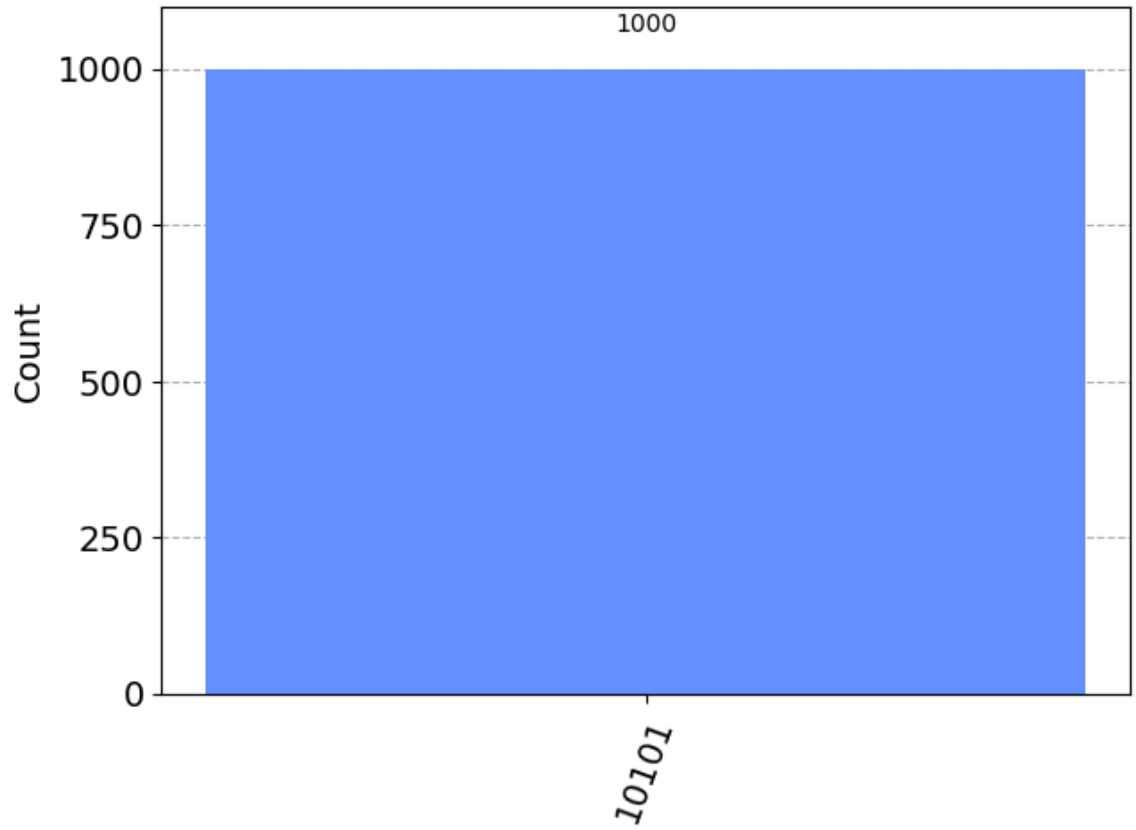
In [51]:
```
job_sim = backend.run(qc_compiled, shots=1000)
result_sim = job_sim.result()
```

In [52]:
```
counts = result_sim.get_counts(qc_compiled)
print(counts)
```

```
{'10101': 1000}
```

In [53]:
```
from qiskit.visualization import plot_histogram
plot_histogram(counts)
```

Out[53]:

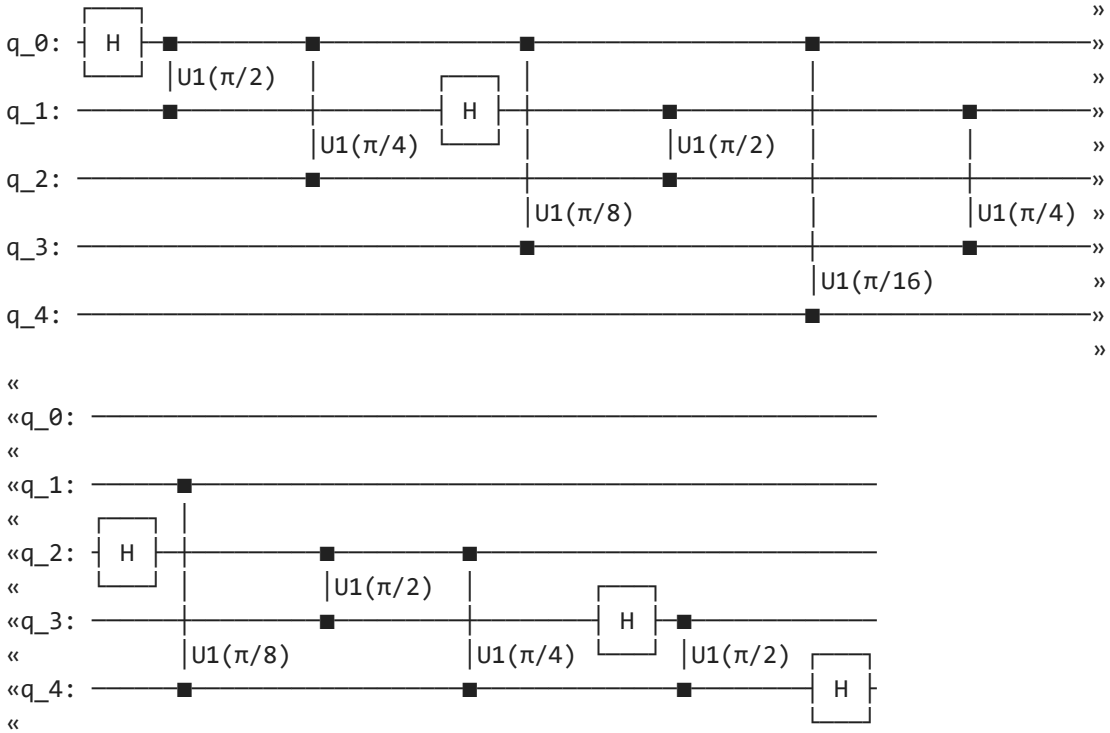


In [56]:

```
def qft(circuit, n):
    # Apply Hadamard gates
    for j in range(n):
        circuit.h(j)
        for k in range(j+1, n):
            angle = pi / float(2**(k-j))
            circuit.append(CU1Gate(angle), [j, k])

num_qubits = 5
qft_circuit = QuantumCircuit(num_qubits)

qft(qft_circuit, num_qubits)
qft_circuit.draw(output='mpl',filename='qft2_circuit.png')
print(qft_circuit)
```


q_0:
q_1:
q_2:
q_3:
q_4:
H
U1(π/2)
U1(π/4)
U1(π/8)
U1(π/16)
U1(π/2)
U1(π/4)
H
H
H
U1(π/2)
U1(π/8)
U1(π/4)
U1(π/2)

In [ ]: