

In [15]:

```
import numpy as np
from numpy import pi
from qiskit import *
from qiskit.visualization import plot_histogram
%matplotlib inline
```

In [16]:

#Applying QPE

```
n=int(input())
cr=ClassicalRegister(n)
qr=QuantumRegister(n)
qc=QuantumCircuit(qr,cr)
for qubit in range(n-1):
    qc.h(qubit)

qc.x(n-1)

repetitions = 1
for counting_qubit in range(n-1):
    for i in range(repetitions):
        qc.cu1(2*pi/3, counting_qubit, n-1); # This is C-U
        repetitions *= 2

for qubit in range((n-1)//2):
    qc.swap(qubit, n-qubit-2)

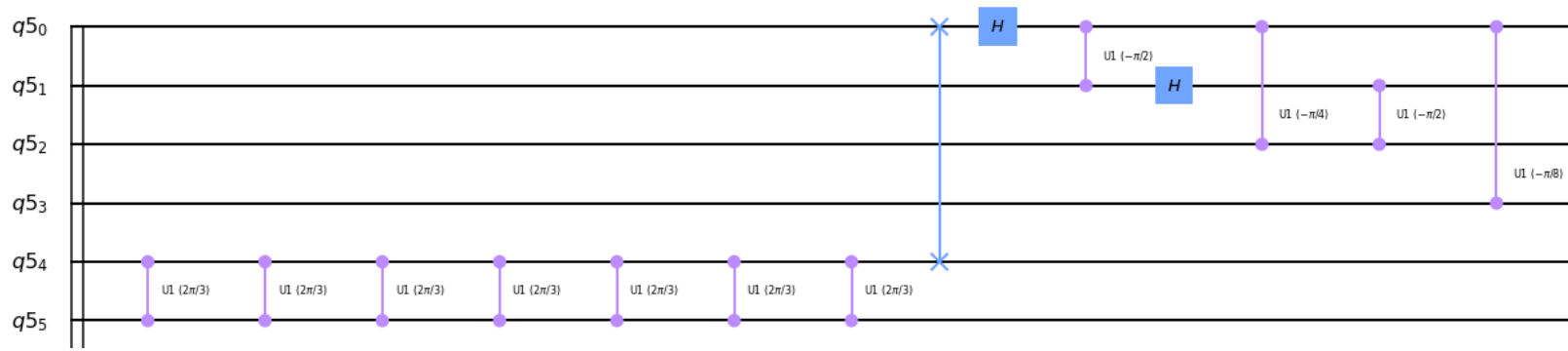
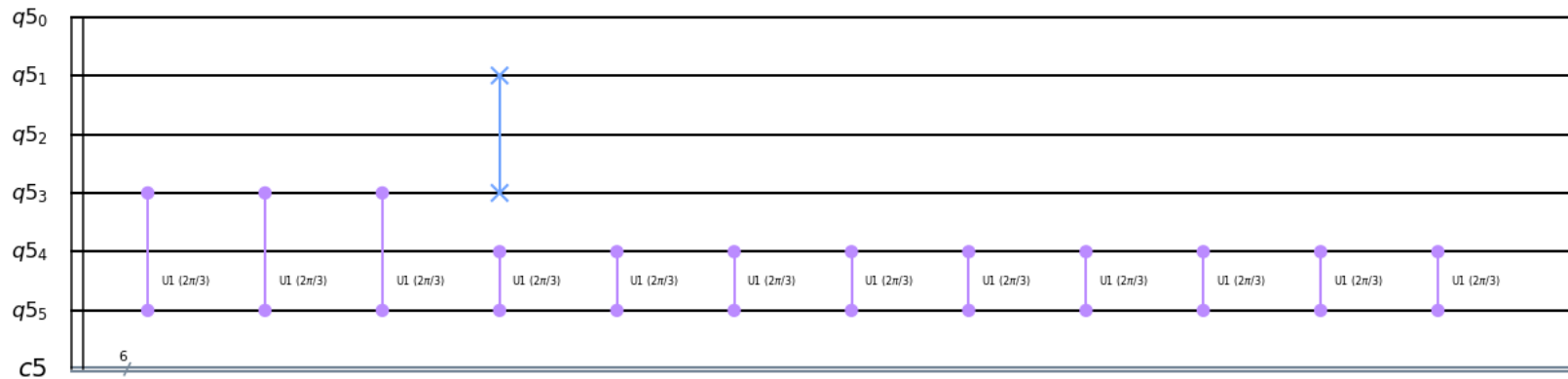
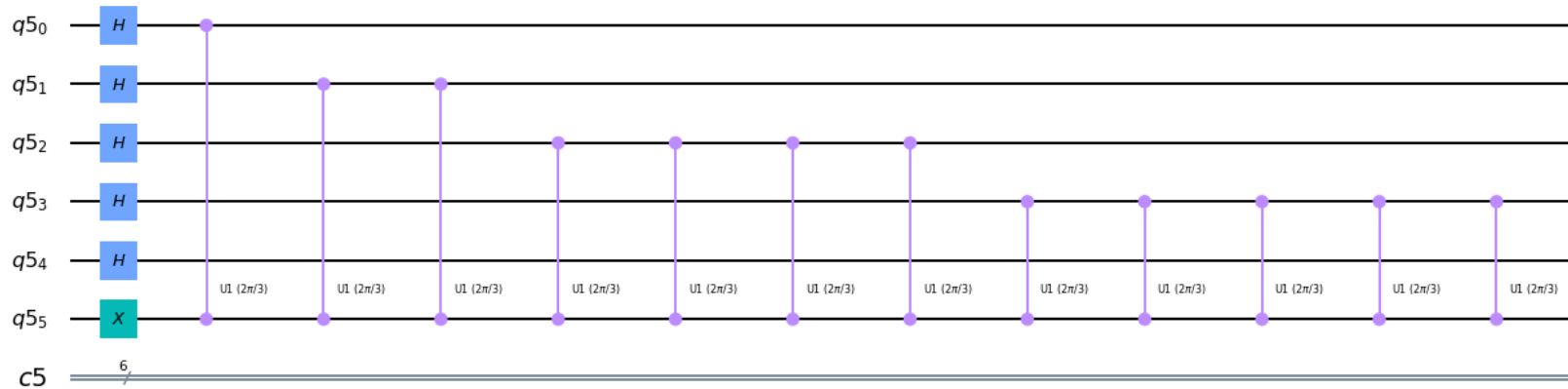
for j in range(n-1):
    for m in range(j):
        qc.cu1(-pi/float(2**(j-m)), m, j)
    qc.h(j)

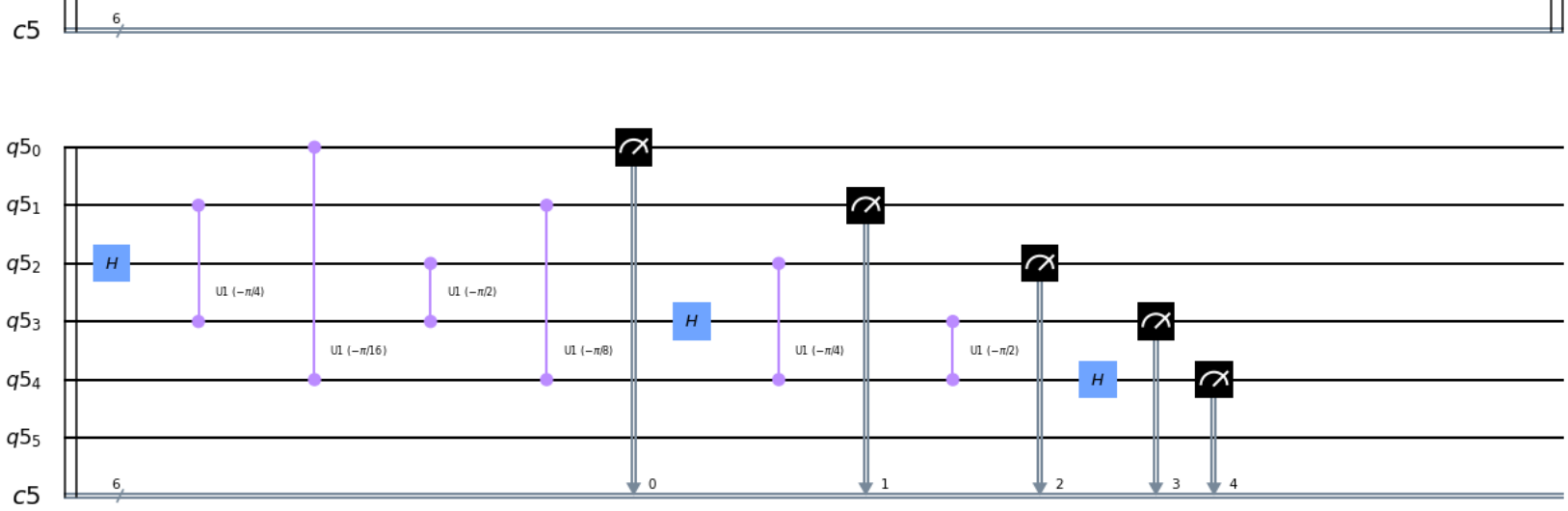
for j in range(n-1):
    qc.measure(j,j)
```

```
qc.draw(output='mpl')
```

6

Out[16]:





In [17]:

```
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, backend=simulator, shots=1024).result()
counts=result.get_counts()
print(counts)
plot_histogram(counts)
```

```
{'010110': 2, '011000': 1, '000101': 1, '000011': 4, '000110': 4, '00  
0111': 7, '010011': 4, '010000': 6, '010100': 2, '010010': 1, '00100  
1': 32, '001000': 13, '001101': 13, '010101': 1, '001111': 4, '00101  
1': 693, '010001': 2, '011100': 1, '001010': 175, '000100': 3, '00000  
0': 1, '001110': 7, '001100': 47}
```

Out[17]:

