



```
In [1]: !pip install qiskit qiskit-aer --upgrade
```

```
Collecting qiskit
  Downloading qiskit-2.2.1-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (12 kB)
Collecting qiskit-aer
  Downloading qiskit_aer-0.17.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (8.3 kB)
Collecting rustworkx<=0.15.0 (from qiskit)
  Downloading rustworkx-0.17.1-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (10 kB)
Requirement already satisfied: numpy<3,>=1.17 in /usr/local/lib/python3.12/dist-packages (from qiskit) (2.0.2)
Requirement already satisfied: scipy<=1.5 in /usr/local/lib/python3.12/dist-packages (from qiskit) (1.16.2)
Requirement already satisfied: dill<=0.3 in /usr/local/lib/python3.12/dist-packages (from qiskit) (0.3.8)
Collecting stevedore<=3.0.0 (from qiskit)
  Downloading stevedore-5.5.0-py3-none-any.whl.metadata (2.2 kB)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.12/dist-packages (from qiskit) (4.15.0)
Requirement already satisfied: psutil<=5 in /usr/local/lib/python3.12/dist-packages (from qiskit-aer) (5.9.5)
Requirement already satisfied: python-dateutil<=2.8.0 in /usr/local/lib/python3.12/dist-packages (from qiskit-aer) (2.9.0.post0)
Requirement already satisfied: six<=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil<=2.8.0->qiskit-aer) (1.17.0)
Downloading qiskit-2.2.1-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (8.0 MB)
 8.0/8.0 MB 65.0 MB/s eta 0:00:00
Downloading qiskit_aer-0.17.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.4 MB)
 12.4/12.4 MB 80.9 MB/s eta 0:00:00
Downloading rustworkx-0.17.1-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.2 MB)
 2.2/2.2 MB 33.9 MB/s eta 0:00:00
Downloading stevedore-5.5.0-py3-none-any.whl (49 kB)
 49.5/49.5 kB 1.8 MB/s eta 0:00:00
Installing collected packages: stevedore, rustworkx, qiskit, qiskit-aer
Successfully installed qiskit-2.2.1 qiskit-aer-0.17.2 rustworkx-0.17.1 stevedore-5.5.0
```

```
In [ ]: #Task 1
```

```
In [9]: sim = AerSimulator()

qc = QuantumCircuit(1, 1)

qc.h(0)
qc.x(0)
qc.s(0)
qc.t(0)
qc.rz(0.5, 0)
```

```

qc.measure_all()

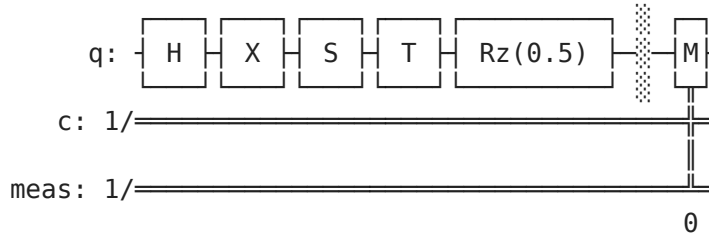
print("Quantum Circuit:")
print(qc.draw())

compiled = transpile(qc, sim)
result = sim.run(compiled, shots=1024).result()

counts = result.get_counts()
print("\nMeasurement Counts:", counts)

```

Quantum Circuit:



Measurement Counts: {'0 0': 509, '1 0': 515}

In [ ]: *#Task 2*

In [4]: `import numpy as np`

```

In [10]: # $\theta = \pi/4$ 
theta = np.pi/4
sim = AerSimulator()

qc = QuantumCircuit(1, 1)

qc.x(0)
qc.h(0)
qc.s(0)
qc.t(0)
qc.rz(theta, 0)

qc.measure_all()

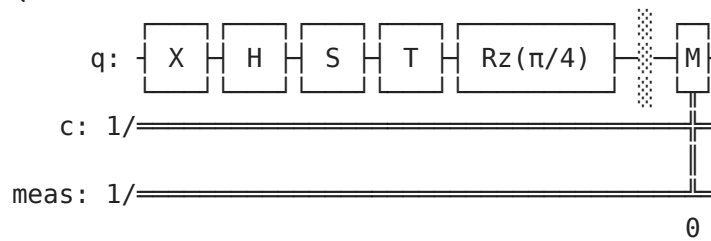
print("Quantum Circuit:")
print(qc.draw())

compiled = transpile(qc, sim)
result = sim.run(compiled, shots=1024).result()

counts = result.get_counts()
print("\nMeasurement Counts:", counts)

```

Quantum Circuit:



Measurement Counts: {'1 0': 509, '0 0': 515}

```
In [11]: #θ = π/2
theta = np.pi/2
sim = AerSimulator()

qc = QuantumCircuit(1, 1)

qc.x(0)
qc.h(0)
qc.s(0)
qc.t(0)
qc.rz(theta, 0)

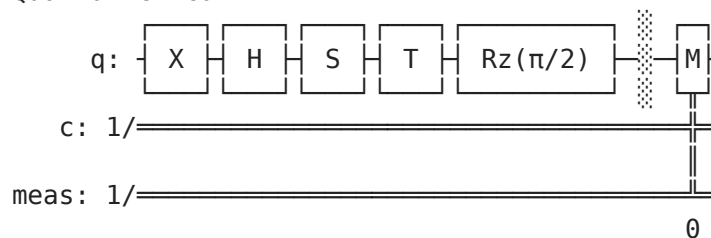
qc.measure_all()

print("Quantum Circuit:")
print(qc.draw())

compiled = transpile(qc, sim)
result = sim.run(compiled, shots=1024).result()

counts = result.get_counts()
print("\nMeasurement Counts:", counts)
```

Quantum Circuit:



Measurement Counts: {'1 0': 549, '0 0': 475}

```
In [12]: #θ = π
theta=np.pi
sim = AerSimulator()

qc = QuantumCircuit(1, 1)

qc.x(0)
qc.h(0)
qc.s(0)
```

```

qc.t(0)
qc.rz(theta, 0)

qc.measure_all()

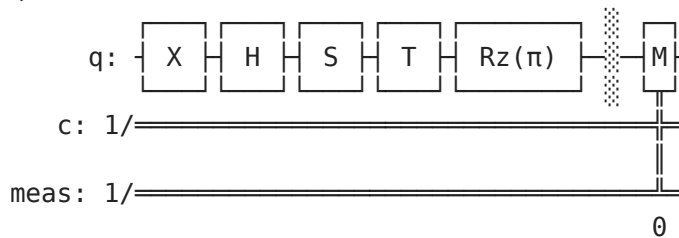
print("Quantum Circuit:")
print(qc.draw())

compiled = transpile(qc, sim)
result = sim.run(compiled, shots=1024).result()

counts = result.get_counts()
print("\nMeasurement Counts:", counts)

```

Quantum Circuit:



Measurement Counts: {'1 0': 493, '0 0': 531}

In [ ]: *#Task 3*

```

In [13]: sim = AerSimulator()

qc = QuantumCircuit(1, 1)

qc.x(0)
qc.s(0)
qc.t(0)
qc.rz(0.5, 0)

qc.measure_all()

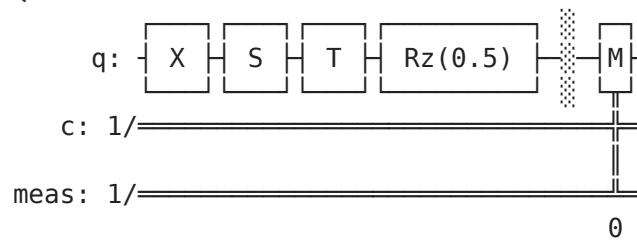
print("Quantum Circuit:")
print(qc.draw())

compiled = transpile(qc, sim)
result = sim.run(compiled, shots=1024).result()

counts = result.get_counts()
print("\nMeasurement Counts:", counts)

```

Quantum Circuit:



Measurement Counts: {'1 0': 1024}