Task I: Quantum Computing Part

1) Implement a simple quantum operation with Cirq or Pennylane

a) With 5 qubits

b) Apply Hadamard operation on every qubit

c) Apply CNOT operation on (0, 1), (1,2), (2,3), (3,4)

d) SWAP (0, 4)

e) Rotate X with pi/2 on any qubit

f) Plot the circuit

```
!pip install pennylane
```

```
import pennylane as qml
import numpy as np
import matplotlib.pyplot as plt

dev = qml.device("default.qubit", wires=5)

@qml.qnode(dev)
def circuit():
    # b)Apply Hadamard operation on every qubit
    for i in range(5):
        qml.Hadamard(wires=i)

    # c)Apply CNOT operation on (0, 1), (1,2), (2,3), (3,4)
    qml.CNOT(wires=[0, 1])
    qml.CNOT(wires=[1, 2])
    qml.CNOT(wires=[2, 3])
    qml.CNOT(wires=[3, 4])

    # d)SWAP (0, 4)
    qml.SWAP(wires=[0, 4])

    # e)Apply an RX(pi/2) gate on qubit 2
    qml.RX(np.pi/2, wires=2)

    return qml.state()

# f)Plot the circuit
circuit()

# Draw the circuit
drawer = qml.draw(circuit)
print(drawer())
```

```
0: ──H─╭●─────────╭SWAP──────────┤  State
1: ──H─╰X─╭●──────╰─────────────┤  State
2: ──H────╰X─╭●────╰─────RX(1.57)─┤  State
3: ──H───────╰X─╭●─╰─────────────┤  State
4: ──H──────────╰X─╰SWAP──────────┤  State
```

---

2) Implement a second circuit with a framework of your choice:

a)Apply a Hadmard gate to the first qubit

b)rotate the second qubit by pi/3 around X

c)Apply Hadamard gate to the third and fourth qubit

d)Perform a swap test between the states of the first and second qubit |q1 q2> and the third and fourth qubit |q3 q4>

```
n_qubits = 4
dev = qml.device("default.qubit", wires=n_qubits)
```

```
@qml.qnode(dev)
def circuit():
    qml.Hadamard(wires=0) # a) Apply a hadamard gate to the first qubit
    qml.RX(np.pi / 3, wires=1) # b) Rotate the second qubit by pi/3 around X axis

    qml.Hadamard(wires=2) # c) Apply a Hadamard gate to the third and fourth qubit
    qml.Hadamard(wires=3)

    qml.SWAP(wires=[0, 1]) # d) Perform a swap test between the states of the first and second qubit |q1 q2>
    qml.SWAP(wires=[2, 3]) # and the third and fourth qubit |q3 q4>

    return qml.probs(wires=[0, 1, 2, 3])

print(qml.draw(circuit)())
```

```
0: ──H─────────╭SWAP─┤ ╭Probs
1: ──RX(1.05)──╰SWAP─┤ ├Probs
2: ──H─────────╭SWAP─┤ ├Probs
3: ──H─────────╰SWAP─┤ ╰Probs
```