Implementation of Task 1 in Cirq

```
!pip install cirq
```

```
Collecting cirq
  Downloading cirq-1.4.1-py3-none-any.whl.metadata (7.4 kB)
Collecting cirq-aqt==1.4.1 (from cirq)
  Downloading cirq_aqt-1.4.1-py3-none-any.whl.metadata (1.6 kB)
Collecting cirq-core==1.4.1 (from cirq)
  Downloading cirq_core-1.4.1-py3-none-any.whl.metadata (1.8 kB)
Collecting cirq-google==1.4.1 (from cirq)
  Downloading cirq_google-1.4.1-py3-none-any.whl.metadata (2.0 kB)
Collecting cirq-ionq==1.4.1 (from cirq)
  Downloading cirq_ionq-1.4.1-py3-none-any.whl.metadata (1.6 kB)
Collecting cirq-pasqal==1.4.1 (from cirq)
  Downloading cirq_pasqal-1.4.1-py3-none-any.whl.metadata (1.6 kB)
Collecting cirq-rigetti==1.4.1 (from cirq)
  Downloading cirq_rigetti-1.4.1-py3-none-any.whl.metadata (1.7 kB)
Collecting cirq-web==1.4.1 (from cirq)
  Downloading cirq_web-1.4.1-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: requests~=2.18 in /usr/local/lib/python3.11/dist-packages (from cirq-aqt==1.4.1->cirq) (2.32.3)
Requirement already satisfied: attrs>=21.3.0 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (25.2.0)
Collecting duet>=0.2.8 (from cirq-core==1.4.1->cirq)
  Downloading duet-0.2.9-py3-none-any.whl.metadata (2.3 kB)
Requirement already satisfied: matplotlib~=3.0 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (3.10.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (3.4.2)
Requirement already satisfied: numpy~=1.22 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (1.26.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (2.2.2)
Requirement already satisfied: sortedcontainers~=2.0 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (2.4.0)
Requirement already satisfied: scipy~=1.0 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (1.14.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (1.13.1)
Requirement already satisfied: typing-extensions>=4.2 in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (4.12.
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from cirq-core==1.4.1->cirq) (4.67.1)
Requirement already satisfied: google-api-core>=1.14.0 in /usr/local/lib/python3.11/dist-packages (from google-api-core[grpc]>=1.14.0
Requirement already satisfied: proto-plus>=1.20.0 in /usr/local/lib/python3.11/dist-packages (from cirq-google==1.4.1->cirq) (1.26.1)
Requirement already satisfied: protobuf<5.0.0,>=3.15.0 in /usr/local/lib/python3.11/dist-packages (from cirq-google==1.4.1->cirq) (4.
Collecting pyquil<5.0.0,>=4.11.0 (from cirq-rigetti==1.4.1->cirq)
  Downloading pyquil-4.16.1-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: googleapis-common-protos<2.0.0,>=1.56.2 in /usr/local/lib/python3.11/dist-packages (from google-api-co
Requirement already satisfied: google-auth<3.0.0,>=2.14.1 in /usr/local/lib/python3.11/dist-packages (from google-api-core>=1.14.0->g
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in /usr/local/lib/python3.11/dist-packages (from google-api-core[grpc]>=1.14.0-
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.11/dist-packages (from google-api-core[grpc]
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->c
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->cirq)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->ci
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->cirq) (1
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.1->c
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib~=3.0->cirq-core==1.4.
Requirement already satisfied: deprecated<2.0.0,>=1.2.14 in /usr/local/lib/python3.11/dist-packages (from pyquil<5.0.0,>=4.11.0->cirq
Collecting packaging>=20.0 (from matplotlib~=3.0->cirq-core==1.4.1->cirq)
  Downloading packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
Collecting qcs-sdk-python>=0.20.1 (from pyquil<5.0.0,>=4.11.0->cirq-rigetti==1.4.1->cirq)
  Downloading qcs_sdk_python-0.21.17-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (7.0 kB)
Collecting quil>=0.15.3 (from pyquil<5.0.0,>=4.11.0->cirq-rigetti==1.4.1->cirq)
  Downloading quil-0.16.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.8 kB)
Collecting rpcq<4.0.0,>=3.11.0 (from pyquil<5.0.0,>=4.11.0->cirq-rigetti==1.4.1->cirq)
  Downloading rpcq-3.11.0.tar.gz (45 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.6/45.6 kB 2.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
```

1) Implement a simple quantum operation with Cirq or Pennylane

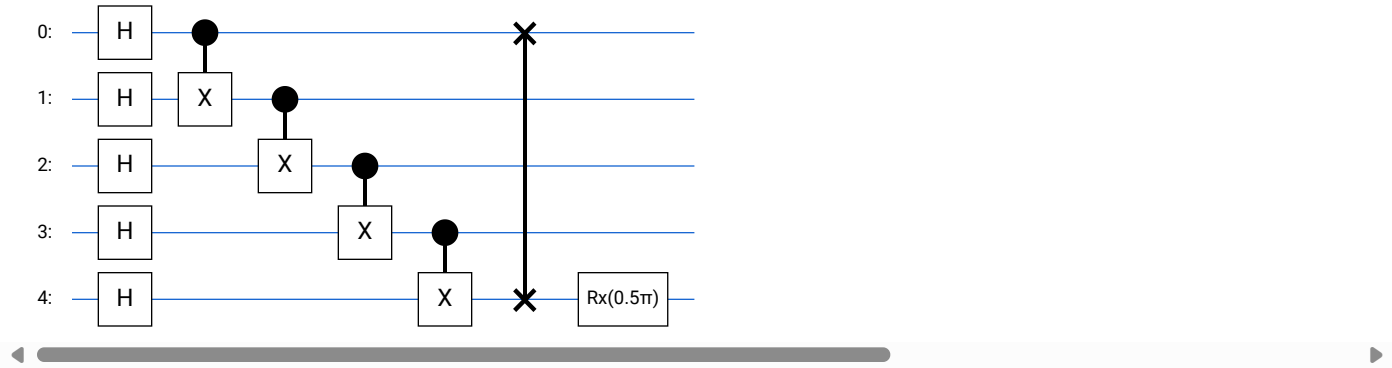
- With 5 qubits
- Apply Hadamard operation on every qubit
- Apply CNOT operation on (0, 1), (1,2), (2,3), (3,4)
- SWAP (0, 4)
- Rotate X with pi/2 on any qubit
- Plot the circuit

```
import cirq
import matplotlib.pyplot as plt
import numpy as np
from cirq.contrib.svg import SVGCircuit
```

```
import random

#Creating 5 qubits
qubits= [cirq.LineQubit(i) for i in range(5)]

#Creating a circuit, and applying Hadamard gate to each qubit
circuit= cirq.Circuit()
circuit.append(cirq.H(qubit) for qubit in qubits)

#Applying CNOT operation on all pairs of adjacent qubits
circuit.append(cirq.CNOT(qubits[i], qubits[i+1]) for i in range(4))

#Applying SWAP operation on 0th and 4th qubit
circuit.append(cirq.SWAP(qubits[0],qubits[4]))

# Apply RX(pi/2) on a random valid qubit
random_int=random.randint(0,4)
circuit.append(cirq.rx(np.pi/2)(qubits[random_int]))

print(f"Printing the circuit:\n {circuit}\n\n\n")
print("Plotting the circuit:")
display(SVGCircuit(circuit))
```

```
Printing the circuit:
0: ──H──@──────────────×──────────────
        │              │
1: ──H──X──@───────────┼──────────────
           │           │
2: ──H─────X──@────────┼──────────────
              │        │
3: ──H────────X──@─────┼──────────────
                 │     │
4: ──H───────────X──×──Rx(0.5π)──
```

Plotting the circuit:



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

- Apply a Hadmard gate to the first qubit
- rotate the second qubit by pi/3 around X
- Apply Hadamard gate to the third and fourth qubit
- Perform a swap test between the states of the first and second qubit |q1 q2> and the third and fourth qubit |q3 q4>

```
# Creating a qubit array with random size from 4 to 10
random_size = random.randint(4, 10)
qubitss = [cirq.LineQubit(i) for i in range(random_size)]

# Creating a circuit
circ = cirq.Circuit()

# Applying Hadamard gate to the 1st qubit (with 0th index)
circ.append(cirq.H(qubitss[0]))

# Rotating the second qubit (with 1st index) by pi/3
circ.append(cirq.rx(np.pi/3)(qubitss[1]))  # Corrected index

# Applying Hadamard gate to the third and fourth qubit (i.e. with index 2 and 3)
circ.append(cirq.H(qubitss[2]))
circ.append(cirq.H(qubitss[3]))
```

```
# Swapping 1st and 2nd qubit i.e. (0,1) and 3rd and 4th qubit (2,3)
circ.append(cirq.SWAP(qubitss[0], qubitss[1]))
circ.append(cirq.SWAP(qubitss[2], qubitss[3]))

# Printing the circuit
print(f"Printing the circuit:\n {circ}\n\n")

# Plotting the circuit
print("Plotting the circuit:")
display(SVGCircuit(circ))
```

```
Printing the circuit:
 0: ──H──────────x──
                 │
1: ──Rx(0.333π)──x──

2: ──H──────────x──
                 │
3: ──H──────────x──


Plotting the circuit:
```