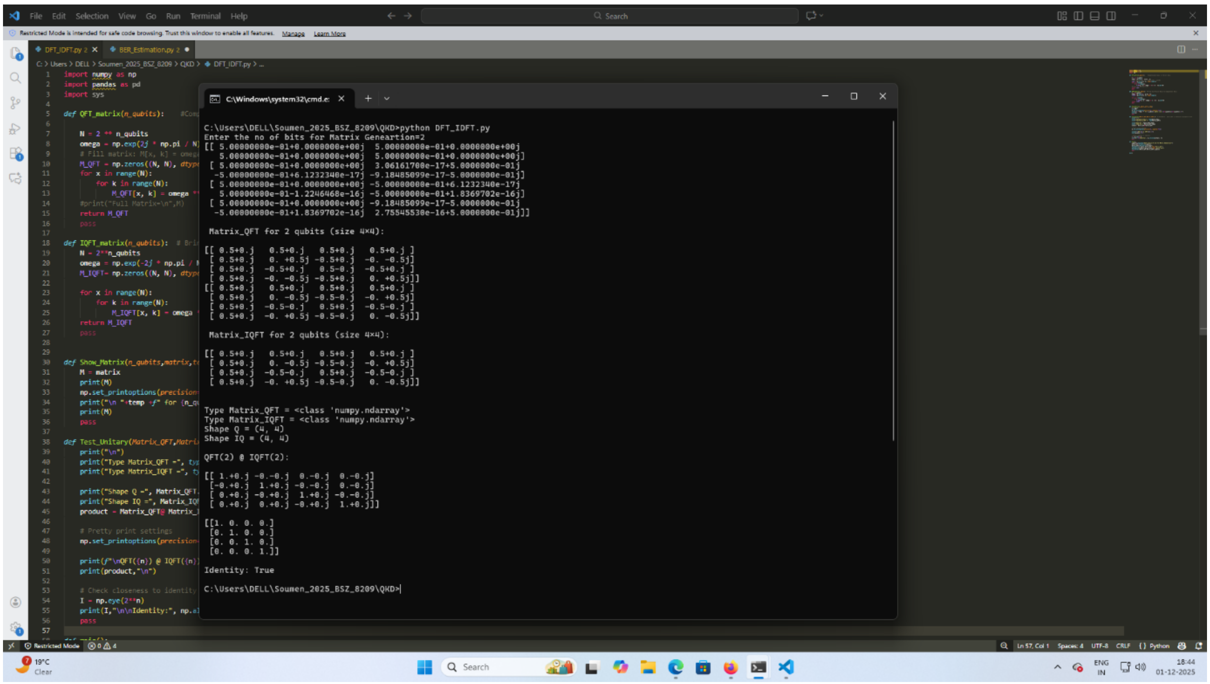
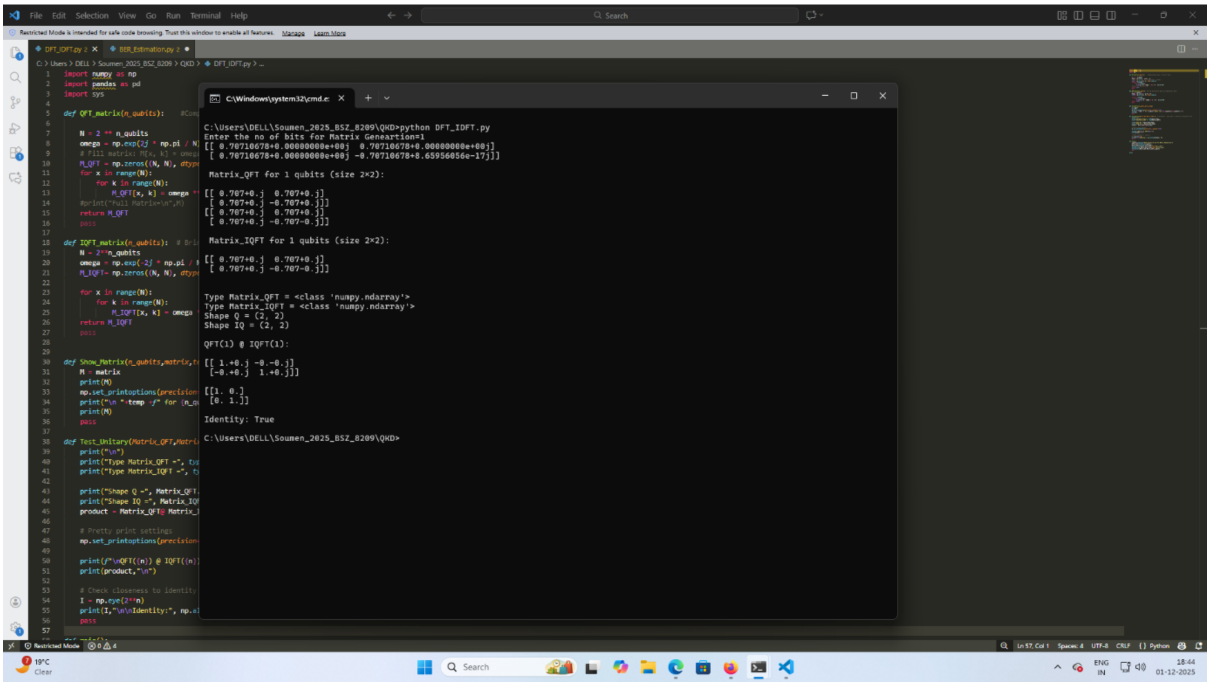
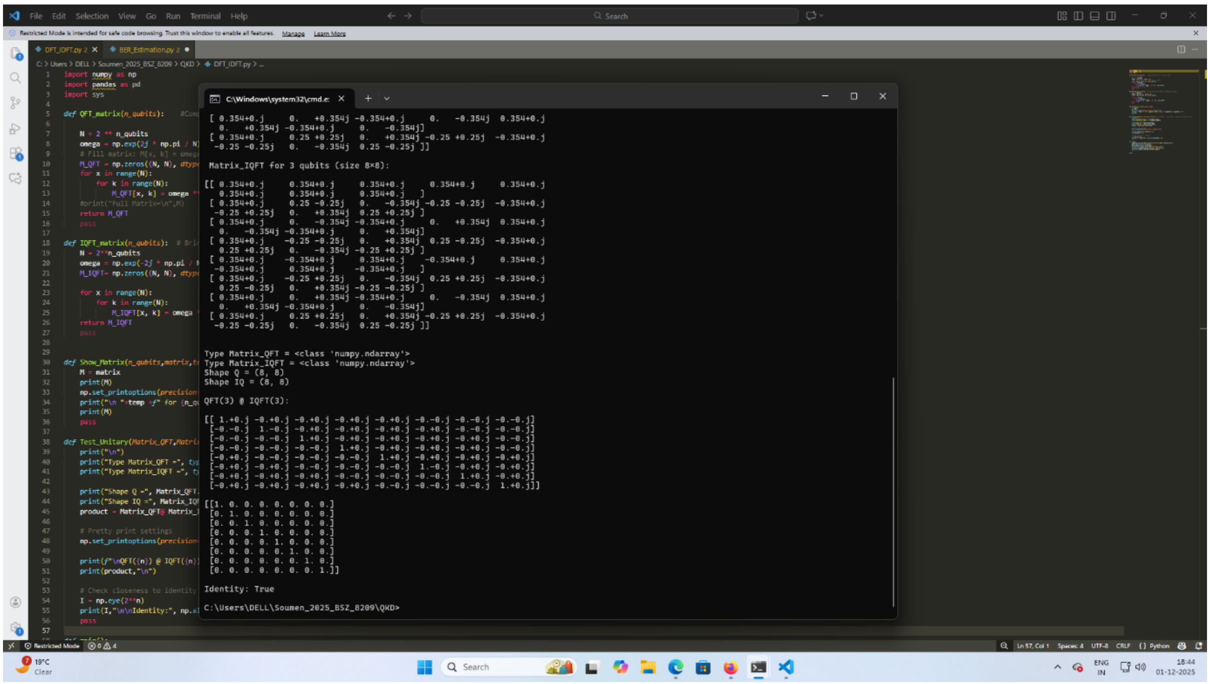
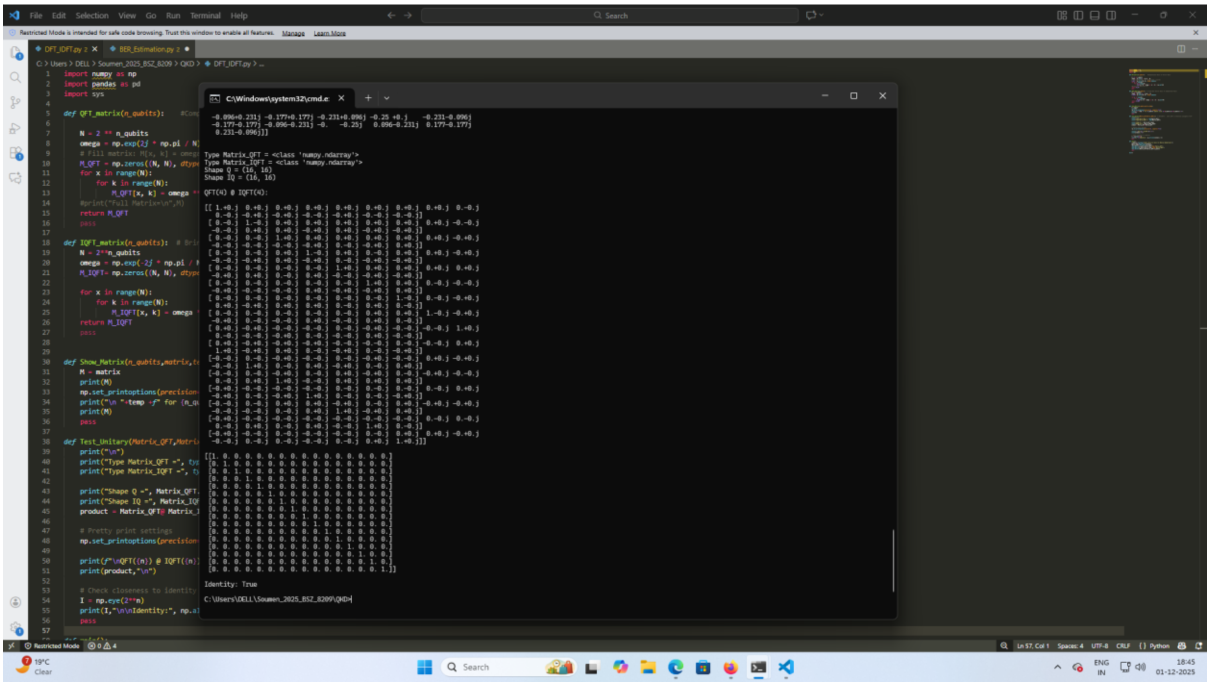
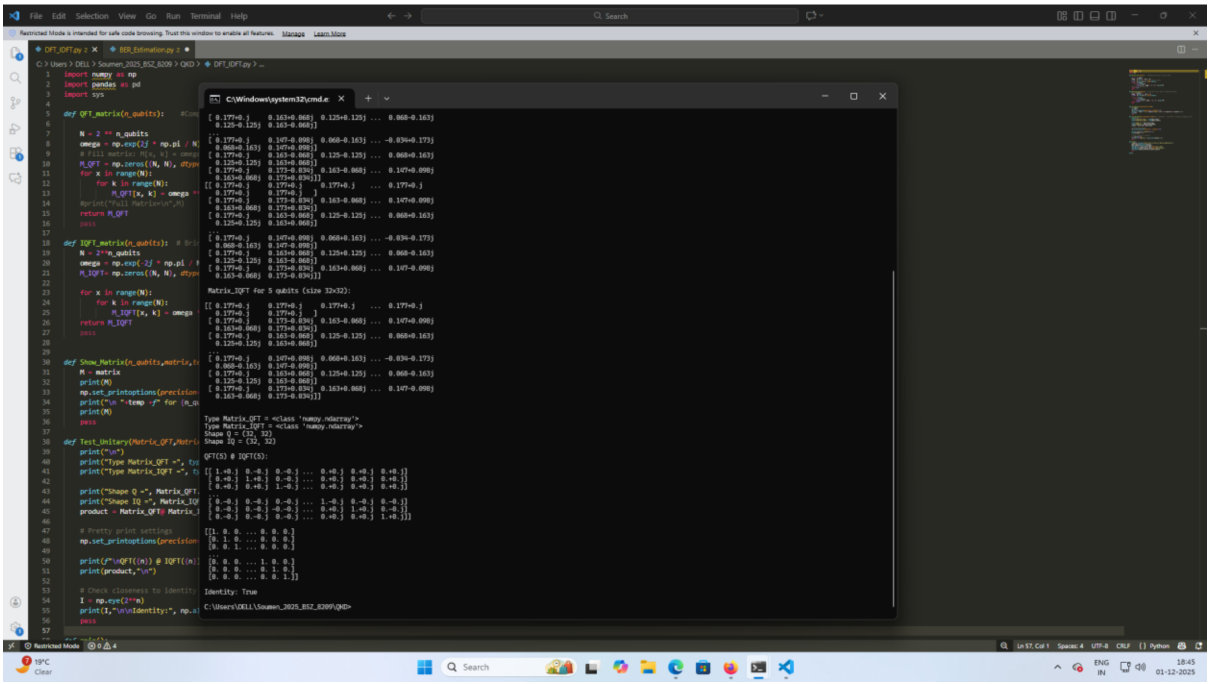
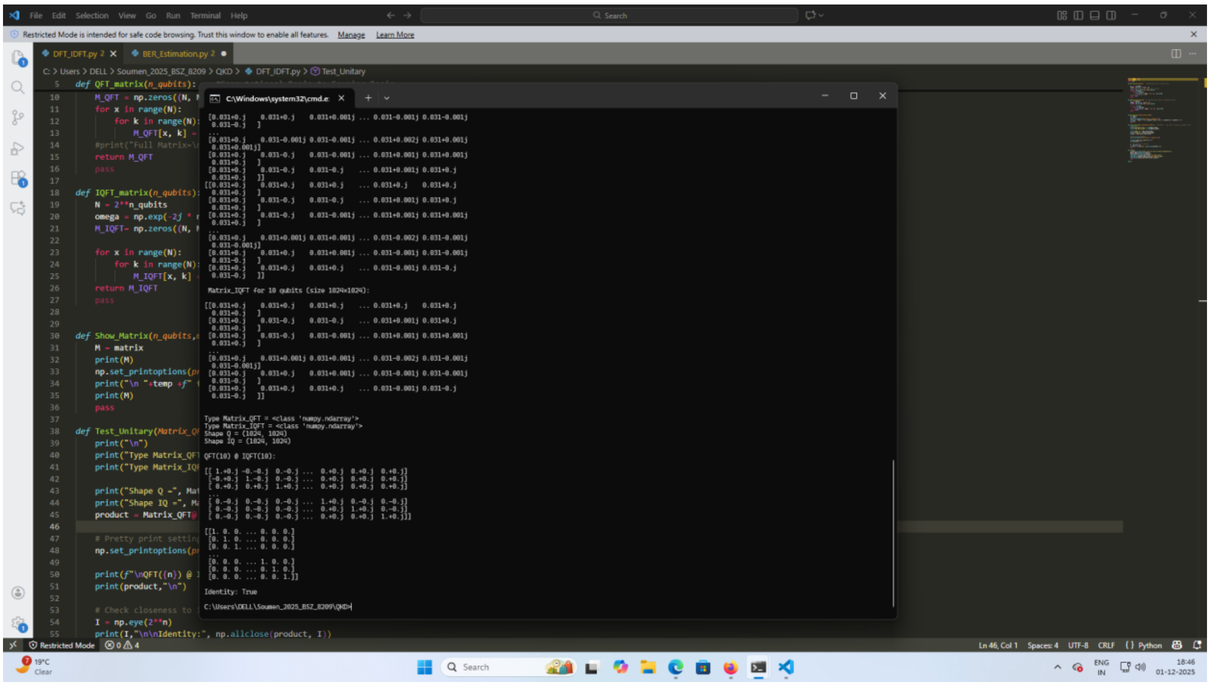

C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python DFT_IDFT.py
Enter the no of bits for Matrix Geneartion=2
[[5.00000000e-01+0.0000000e+00j 5.00000000e-01+0.0000000e+00j
5.00000000e-01+0.0000000e+00j 5.00000000e-01+0.0000000e+00j]
[5.00000000e-01+0.0000000e+00j 3.06161700e-17+5.0000000e-01j
-5.00000000e-01+6.1232340e-17j -9.18485099e-17-5.0000000e-01j]
[5.00000000e-01+0.0000000e+00j -5.00000000e-01+6.1232340e-17j
5.00000000e-01-1.2246468e-16j -5.00000000e-01+1.8369702e-16j]
[5.00000000e-01+0.0000000e+00j -9.18485099e-17-5.0000000e-01j
-5.00000000e-01+1.8369702e-16j 2.75545530e-16+5.0000000e-01j]]
Matrix_QFT for 2 qubits (size 4×4):
[[0.5+0.j 0.5+0.j 0.5+0.j 0.5+0.j]
[0.5+0.j 0. +0.5j -0.5+0.j -0. -0.5j]
[0.5+0.j -0.5+0.j 0.5-0.j -0.5+0.j]
[0.5+0.j -0. -0.5j -0.5+0.j 0. +0.5j]]
[[0.5+0.j 0.5+0.j 0.5+0.j 0.5+0.j]
[0.5+0.j 0. -0.5j -0.5-0.j -0. +0.5j]
[0.5+0.j -0.5-0.j 0.5+0.j -0.5-0.j]
[0.5+0.j -0. +0.5j -0.5-0.j 0. -0.5j]]
Matrix_IQFT for 2 qubits (size 4×4):
[[0.5+0.j 0.5+0.j 0.5+0.j 0.5+0.j]
[0.5+0.j 0. -0.5j -0.5-0.j -0. +0.5j]
[0.5+0.j -0.5-0.j 0.5+0.j -0.5-0.j]
[0.5+0.j -0. +0.5j -0.5-0.j 0. -0.5j]]
Type Matrix_QFT = <class 'numpy.ndarray'>
Type Matrix_IQFT = <class 'numpy.ndarray'>
Shape Q = (4, 4)
Shape IQ = (4, 4)
QFT(2) @ IQFT(2):
[[1.+0.j -0.-0.j 0.-0.j 0.-0.j]
[-0.+0.j 1.+0.j -0.-0.j 0.-0.j]
[0.+0.j -0.+0.j 1.+0.j -0.-0.j]
[0.+0.j 0.+0.j -0.+0.j 1.+0.j]]
[[1. 0. 0. 0.]
[0. 1. 0. 0.]
[0. 0. 1. 0.]
[0. 0. 0. 1.]]
Identity: True
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>

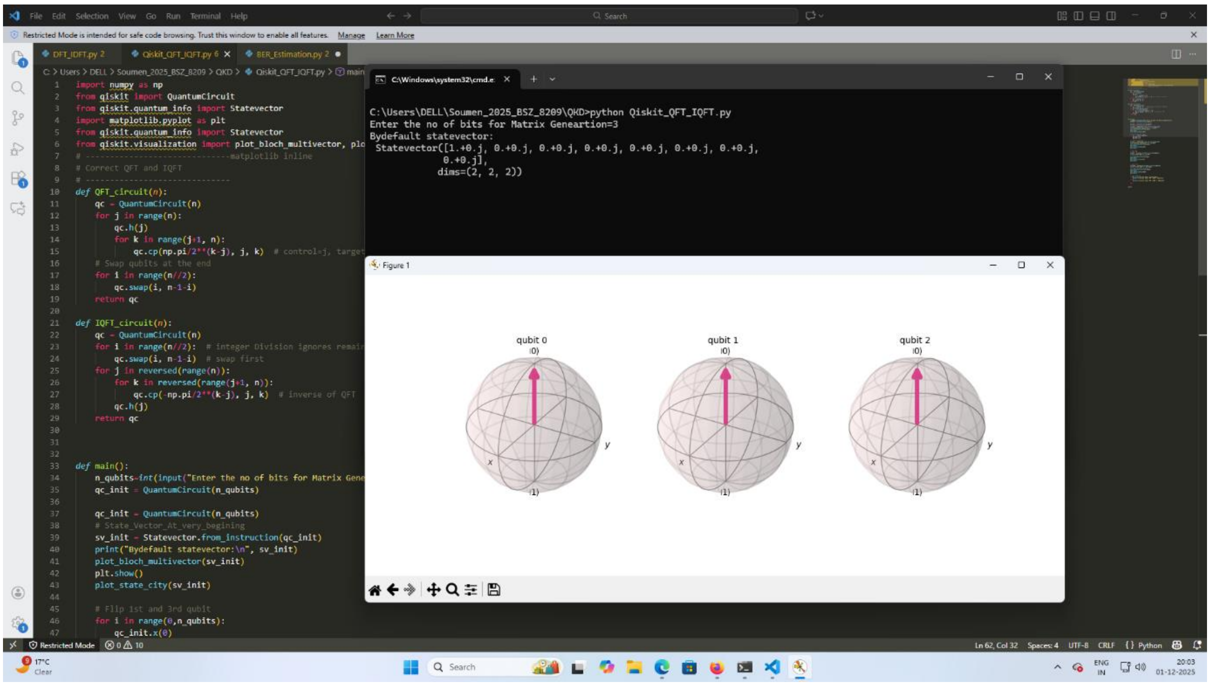

C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python DFT_IDFT.py
Enter the no of bits for Matrix Geneartion=1
[[0.70710678+0.00000000e+00j 0.70710678+0.00000000e+00j]
[0.70710678+0.00000000e+00j -0.70710678+8.65956056e-17j]]
Matrix_QFT for 1 qubits (size 2×2):
[[0.707+0.j 0.707+0.j]
[0.707+0.j -0.707+0.j]]
[[0.707+0.j 0.707+0.j]
[0.707+0.j -0.707-0.j]]
Matrix_IQFT for 1 qubits (size 2×2):
[[0.707+0.j 0.707+0.j]
[0.707+0.j -0.707-0.j]]
Type Matrix_QFT = <class 'numpy.ndarray'>
Type Matrix_IQFT = <class 'numpy.ndarray'>
Shape Q = (2, 2)
Shape IQ = (2, 2)
QFT(1) @ IQFT(1):
[[1.+0.j -0.-0.j]
[-0.+0.j 1.+0.j]]
[[1. 0.]
[0. 1.]]
Identity: True
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>

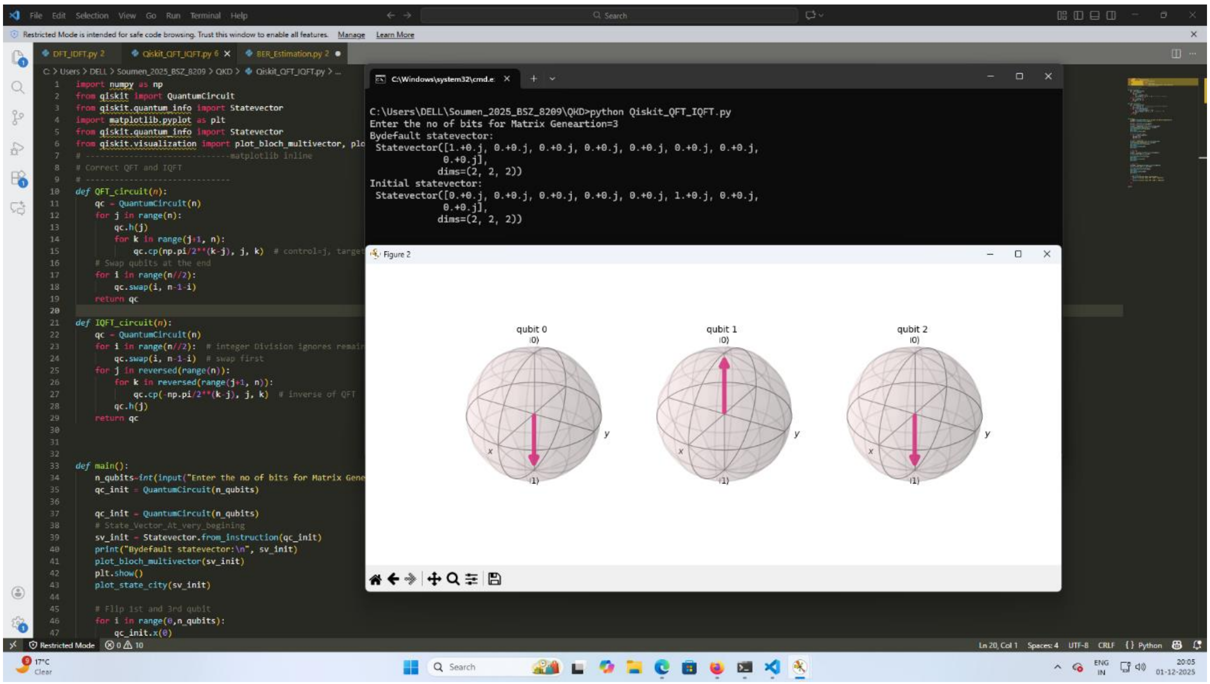

C:\Windows\system32\cmd.e
[0.354+0.j 0. +0.354j -0.354+0.j 0. -0.354j 0.354+0.j
0. +0.354j -0.354+0.j 0. -0.354j]
[0.354+0.j 0.25 +0.25j 0. +0.354j -0.25 +0.25j -0.354+0.j
-0.25 -0.25j 0. -0.354j 0.25 -0.25j]]
Matrix_IQFT for 3 qubits (size 8x8):
[[0.354+0.j 0.354+0.j 0.354+0.j 0.354+0.j 0.354+0.j
0.354+0.j 0.354+0.j 0.354+0.j]
[0.354+0.j 0.25 -0.25j 0. -0.354j -0.25 -0.25j -0.354+0.j
-0.25 +0.25j 0. +0.354j 0.25 +0.25j]
[0.354+0.j 0. -0.354j -0.354+0.j 0. +0.354j 0.354+0.j
0. -0.354j -0.354+0.j 0. +0.354j]
[0.354+0.j -0.25 -0.25j 0. +0.354j 0.25 -0.25j -0.354+0.j
0.25 +0.25j 0. -0.354j -0.25 +0.25j]
[0.354+0.j -0.354+0.j 0.354+0.j -0.354+0.j 0.354+0.j
-0.354+0.j 0.354+0.j -0.354+0.j]
[0.354+0.j -0.25 +0.25j 0. -0.354j 0.25 +0.25j -0.354+0.j
0.25 -0.25j 0. +0.354j -0.25 -0.25j]
[0.354+0.j 0. +0.354j -0.354+0.j 0. -0.354j 0.354+0.j
0. +0.354j -0.354+0.j 0. -0.354j]
[0.354+0.j 0.25 +0.25j 0. +0.354j -0.25 +0.25j -0.354+0.j
-0.25 -0.25j 0. -0.354j 0.25 -0.25j]]
Type Matrix_QFT = <class 'numpy.ndarray'>
Type Matrix_IQFT = <class 'numpy.ndarray'>
Shape Q = (8, 8)
Shape IQ = (8, 8)
QFT(3) @ IQFT(3):
[[1.+0.j -0.+0.j -0.+0.j -0.+0.j -0.+0.j -0.-0.j -0.-0.j -0.-0.j]
[-0.-0.j 1.-0.j -0.+0.j -0.+0.j -0.+0.j -0.+0.j -0.-0.j -0.-0.j]
[-0.-0.j -0.-0.j 1.+0.j -0.+0.j -0.+0.j -0.+0.j -0.+0.j -0.-0.j]
[-0.-0.j -0.-0.j -0.-0.j 1.+0.j -0.+0.j -0.+0.j -0.+0.j -0.-0.j]
[-0.+0.j -0.-0.j -0.-0.j -0.-0.j 1.+0.j -0.+0.j -0.+0.j -0.+0.j]
[-0.+0.j -0.+0.j -0.-0.j -0.-0.j -0.-0.j 1.-0.j -0.+0.j -0.+0.j]
[-0.+0.j -0.+0.j -0.+0.j -0.-0.j -0.-0.j -0.-0.j 1.+0.j -0.+0.j]
[-0.+0.j -0.+0.j -0.+0.j -0.+0.j -0.-0.j -0.-0.j -0.-0.j 1.+0.j]]
[[1. 0. 0. 0. 0. 0. 0. 0.]
[0. 1. 0. 0. 0. 0. 0. 0.]
[0. 0. 1. 0. 0. 0. 0. 0.]
[0. 0. 0. 1. 0. 0. 0. 0.]
[0. 0. 0. 0. 1. 0. 0. 0.]
[0. 0. 0. 0. 0. 1. 0. 0.]
[0. 0. 0. 0. 0. 0. 1. 0.]
[0. 0. 0. 0. 0. 0. 0. 1.]]
Identity: True
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>

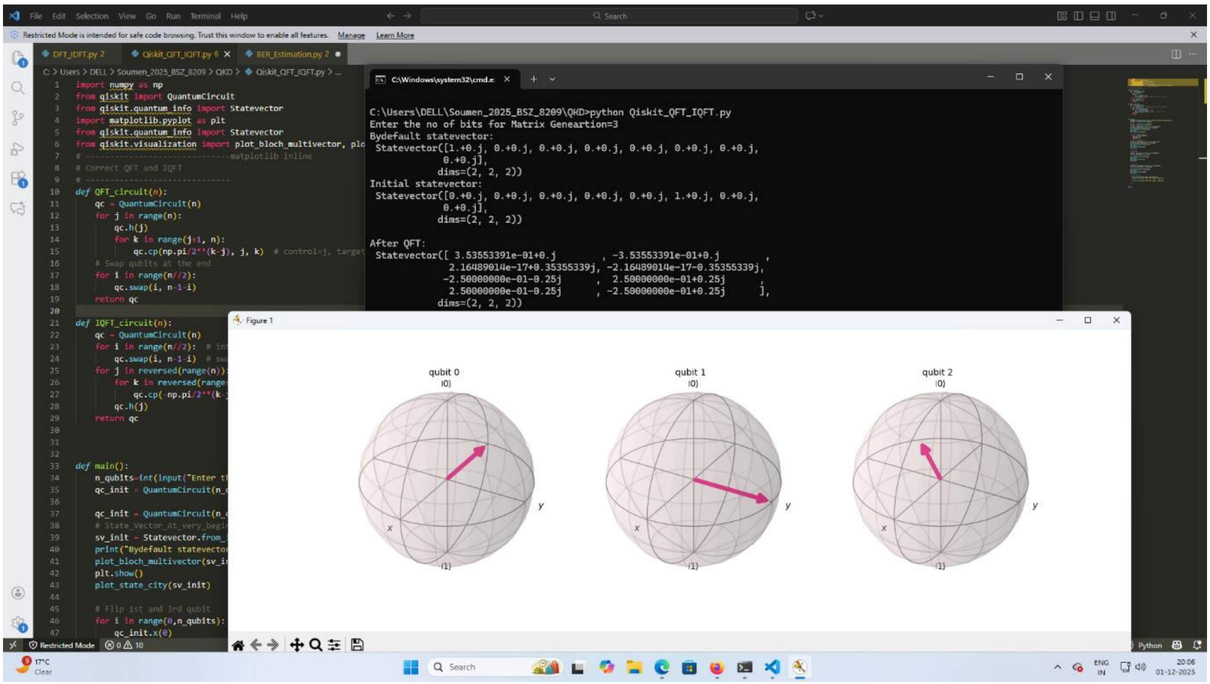

C:\Windows\system32\cmd.e
Type Matrix_QFT = <class 'numpy.ndarray'>
Type Matrix_IQFT = <class 'numpy.ndarray'>
Shape Q = (16, 16)
Shape IQ = (16, 16)
QFT(4) @ IQFT(4):
Identity: True
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>

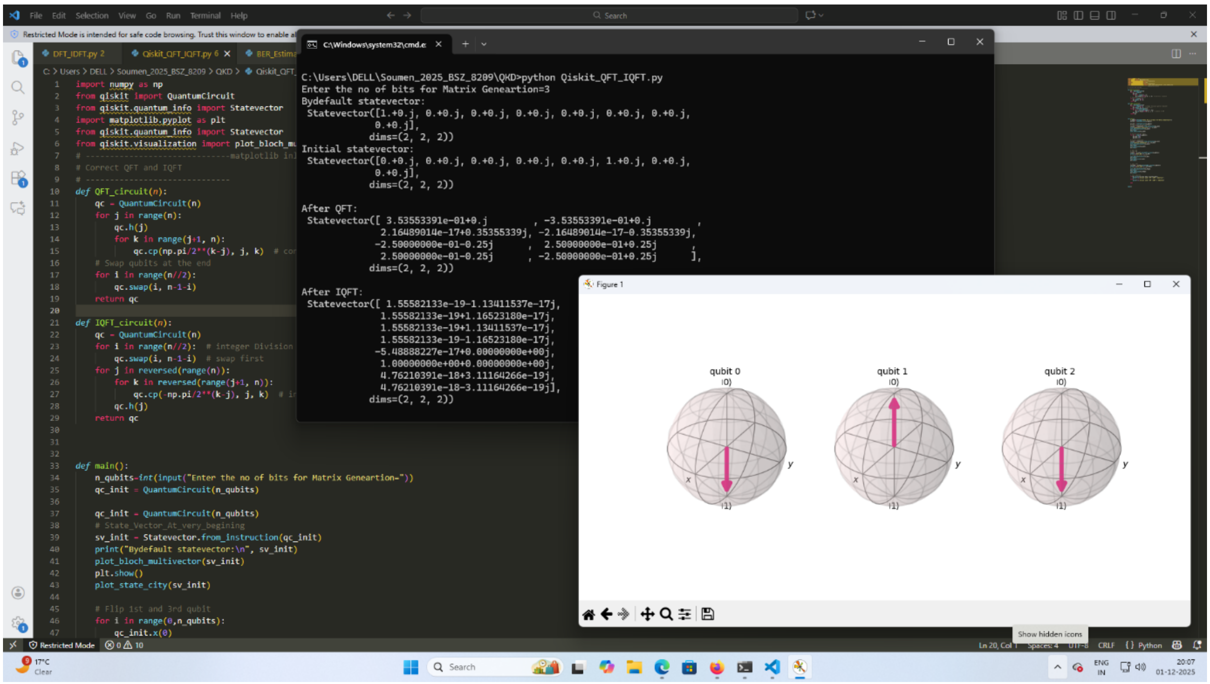


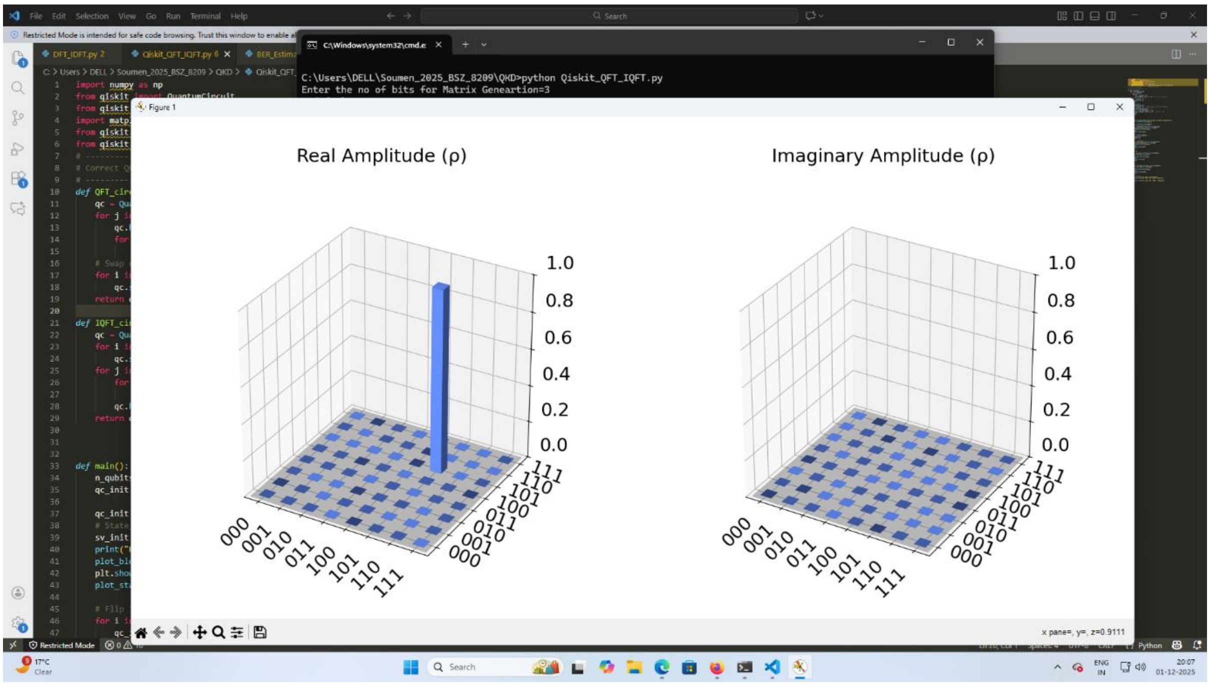


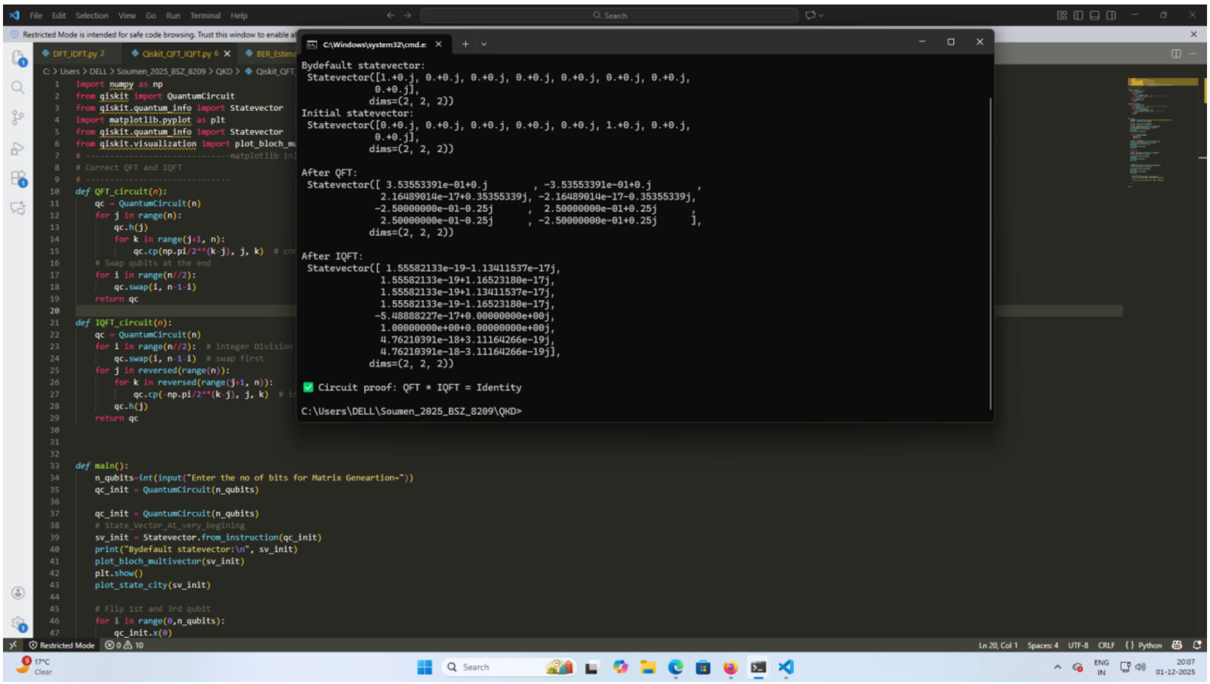

C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python Qiskit_QFT_IQFT.py
Enter the no of bits for Matrix Geneartion=3
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Figure 1
qubit 0
qubit 1
qubit 2
import numpy as np
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector
import matplotlib.pyplot as plt
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_bloch_multivector, plo
# ----------------------------matplotlib inline
# Correct QFT and IQFT
# ----------------------------
def QFT_circuit(n):
qc = QuantumCircuit(n)
for j in range(n):
qc.h(j)
for k in range(j+1, n):
qc.cp(np.pi/2**(k-j), j, k) # control=j, target
# Swap qubits at the end
for i in range(n//2):
qc.swap(i, n-1-i)
return qc
def IQFT_circuit(n):
qc = QuantumCircuit(n)
for i in range(n//2): # integer division ignores remai
qc.swap(i, n-1-i) # swap first
for j in reversed(range(n)):
for k in reversed(range(j+1, n)):
qc.cp(-np.pi/2**(k-j), j, k) # inverse of QFT
qc.h(j)
return qc
def main():
n_qubits=int(input("Enter the no of bits for Matrix Gene
qc_init = QuantumCircuit(n_qubits)
qc_init = QuantumCircuit(n_qubits)
# State_Vector_At_very_begining
sv_init = Statevector.from_instruction(qc_init)
print("Bydefault statevector:\n", sv_init)
plot_bloch_multivector(sv_init)
plt.show()
plot_state_city(sv_init)
# Flip 1st and 3rd qubit
for i in range(0,n_qubits):
qc_init.x(0)

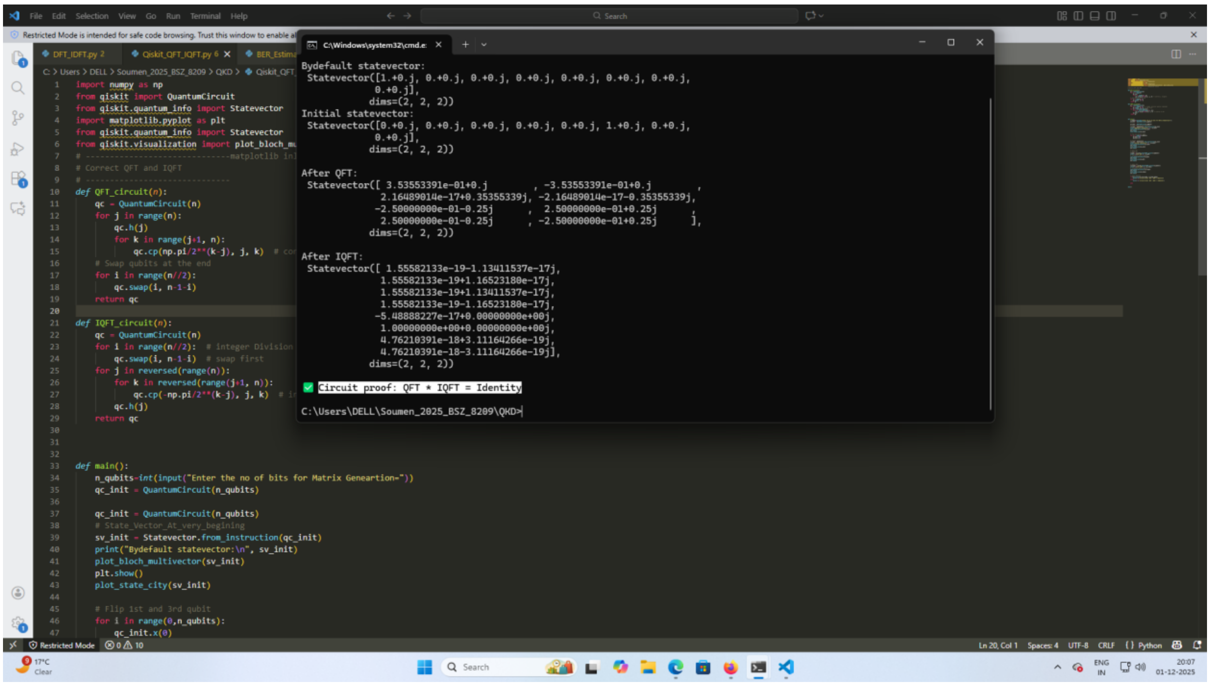

C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python Qiskit_QFT_IQFT.py
Enter the no of bits for Matrix Geneartion=3
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Initial statevector:
Statevector([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Figure 2
qubit 0
qubit 1
qubit 2

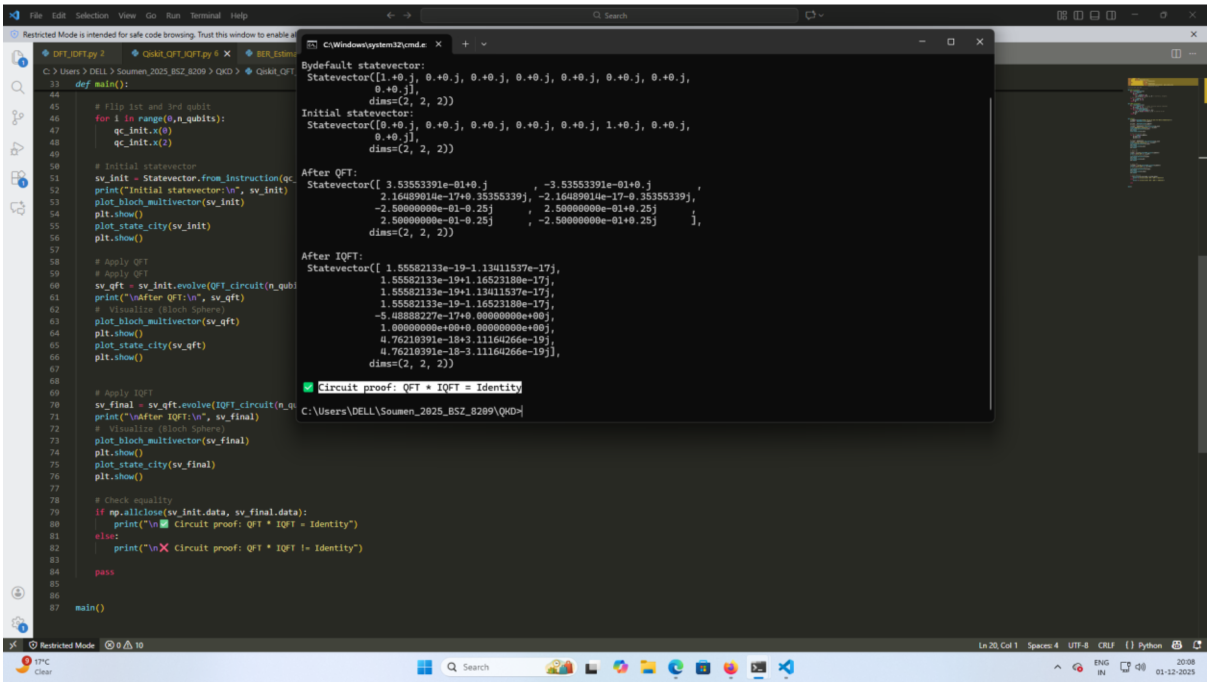

C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python Qiskit_QFT_IQFT.py
Enter the no of bits for Matrix Genearion=3
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Initial statevector:
Statevector([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
After QFT:
Statevector([3.53553391e-01+0.j , -3.53553391e-01+0.j ,
2.16489014e-17+0.35355339j, -2.16489014e-17-0.35355339j,
-2.50000000e-01-0.25j , 2.50000000e-01+0.25j ,
2.50000000e-01-0.25j , -2.50000000e-01+0.25j],
dims=(2, 2, 2))
Figure 1
qubit 0
qubit 1
qubit 2


C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python Qiskit_QFT_IQFT.py
Enter the no of bits for Matrix Genearion=3
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Initial statevector:
Statevector([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
After QFT:
Statevector([3.53553391e-01+0.j , -3.53553391e-01+0.j ,
2.16489014e-17+0.35355339j, -2.16489014e-17-0.35355339j,
-2.50000000e-01-0.25j , 2.50000000e-01+0.25j ,
2.50000000e-01-0.25j , -2.50000000e-01+0.25j],
dims=(2, 2, 2))
After IQFT:
Statevector([1.55582133e-19-1.13411537e-17j,
1.55582133e-19+1.16523180e-17j,
1.55582133e-19+1.13411537e-17j,
1.55582133e-19-1.16523180e-17j,
-5.48888227e-17+0.00000000e+00j,
1.00000000e+00+0.00000000e+00j,
4.76210391e-18+3.11164266e-19j,
4.76210391e-18-3.11164266e-19j],
dims=(2, 2, 2))
Figure 1
qubit 0
qubit 1
qubit 2


C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>python Qiskit_QFT_IQFT.py
Enter the no of bits for Matrix Geneartion=3
Figure 1
Real Amplitude (ρ)
Imaginary Amplitude (ρ)
1.0
0.8
0.6
0.4
0.2
0.0
000
001
010
011
100
101
110
111


C:\Windows\system32\cmd.e
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
Initial statevector:
Statevector([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j,
0.+0.j],
dims=(2, 2, 2))
After QFT:
Statevector([3.53553391e-01+0.j , -3.53553391e-01+0.j ,
2.16489014e-17+0.35355339j, -2.16489014e-17-0.35355339j,
-2.50000000e-01-0.25j , 2.50000000e-01+0.25j ,
2.50000000e-01-0.25j , -2.50000000e-01+0.25j],
dims=(2, 2, 2))
After IQFT:
Statevector([1.55582133e-19-1.13411537e-17j,
1.55582133e-19+1.16523180e-17j,
1.55582133e-19+1.13411537e-17j,
1.55582133e-19-1.16523180e-17j,
-5.48888227e-17+0.00000000e+00j,
1.00000000e+00+0.00000000e+00j,
4.76210391e-18+3.11164266e-19j,
4.76210391e-18-3.11164266e-19j],
dims=(2, 2, 2))
Circuit proof: QFT * IQFT = Identity
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>


C:\Windows\system32\cmd.e
Bydefault statevector:
Statevector([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j], dims=(2, 2, 2))
Initial statevector:
Statevector([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 1.+0.j, 0.+0.j, 0.+0.j], dims=(2, 2, 2))
After QFT:
Statevector([3.53553391e-01+0.j , -3.53553391e-01+0.j , 2.16489014e-17+0.35355339j, -2.16489014e-17-0.35355339j, -2.50000000e-01-0.25j , 2.50000000e-01+0.25j , 2.50000000e-01-0.25j , -2.50000000e-01+0.25j], dims=(2, 2, 2))
After IQFT:
Statevector([1.55582133e-19-1.13411537e-17j, 1.55582133e-19+1.16523180e-17j, 1.55582133e-19+1.13411537e-17j, 1.55582133e-19-1.16523180e-17j, -5.48888227e-17+0.00000000e+00j, 1.00000000e+00+0.00000000e+00j, 4.76210391e-18+3.11164266e-19j, 4.76210391e-18-3.11164266e-19j], dims=(2, 2, 2))
Circuit proof: QFT * IQFT = Identity
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>


C:\Windows\system32\cmd.e
Circuit proof: QFT * IQFT = Identity
C:\Users\DELL\Soumen_2025_BSZ_8209\QKD>