

```

# Import necessary libraries and modules
from quantum_library import QuantumComputer
from encryption_module import QuantumEncryption
from temporal_simulation import TemporalSimulation

# Define Quantum Temporal Nexus (QTN) parameters
qtn_params = {
    'wormhole_stability': 0.95,
    'dimensional_anchor': '1984',
    'temporal_field_strength': 'adaptive',
}

# Initialize Quantum Computer
quantum_computer = QuantumComputer()

# Initialize Quantum Encryption Module
quantum_encryption = QuantumEncryption()

# Initialize Temporal Simulation Module
temporal_simulation = TemporalSimulation()

# Define Quantum Preservation Protocol (QPP) parameters
qpp_params = {
    'consciousness_digitization': True,
    'matrix_encryption': 'ChaoticQuantum',
}

# Main program loop
def empyrean_imperative():
    # Generate Quantum Temporal Nexus (QTN) based on parameters
    qtn = quantum_computer.create_qtn(qtn_params)

    # Encrypt and preserve human consciousness using QPP
    encrypted_consciousness =
quantum_encryption.encrypt_consciousness(qpp_params)

    # Simulate temporal transition to the 1980s within the Halcyon Halls
    simulated_reality = temporal_simulation.simulate_reality(qtn,
encrypted_consciousness)

    # Display simulated reality to the users within the sanctuary
    display(simulated_reality)

# Execute the Empyrean Imperative
empyrean_imperative()

```