TASK 6: Quantum Error Correction (9-Qubit Code)

Aim:
To demonstrate logical qubit encoding and error protection using the 9-qubit Shor code.
Algorithm:
· Encode logical |0⟩ and |1⟩ into 9-qubit states.
· Simulate bit-flip noise on the encoded qubits.
· Demonstrate noise resilience.
· Output encoded and noisy states.

Program :-

```
import numpy as np

print("\n" + "="*50)
print("TASK 6: QUANTUM ERROR CORRECTION (9-QUBIT CODE)")
print("="*50)

def encode_9_qubit(logical_qubit):
    """Encode logical qubit into 9 physical qubits (simplified Shor code)"""
    encoded = np.zeros(2**9)
    if np.allclose(logical_qubit, [1, 0]):   # |0⟩
        encoded[0] = 1                       # |000000000⟩
    elif np.allclose(logical_qubit, [0, 1]): # |1⟩
        encoded[-1] = 1                      # |111111111⟩
    else:
        raise ValueError("Only |0⟩ or |1⟩ supported in this
```

```
simplified encoding.")
    return encoded

def add_bit_flip_noise(state, error_prob=0.1):
    """Add bit flip noise to one random qubit with given
probability"""
    if np.random.rand() < error_prob:
        qubit_to_flip = np.random.randint(9)  # pick random qubit
        print(f"Bit flip error occurred on qubit {qubit_to_flip}")
        # Flip basis index in computational basis
        new_state = np.zeros_like(state)
        index = np.argmax(state)             # find which basis
state is '1'
        bitstring = list(format(index, "09b"))
        bitstring[qubit_to_flip] = "1" if bitstring[qubit_to_flip] ==
"0" else "0"
        new_index = int("".join(bitstring), 2)
        new_state[new_index] = 1
        return new_state
    else:
        print("No error occurred")
        return state

# Encode logical qubits
logical_0 = np.array([1, 0])
logical_1 = np.array([0, 1])

encoded_0 = encode_9_qubit(logical_0)
encoded_1 = encode_9_qubit(logical_1)
```

```
print("Logical |0⟩ encoded into 9-qubit code → |000000000⟩")
print("Logical |1⟩ encoded into 9-qubit code → |111111111⟩")

# Simulate noise
noisy_0 = add_bit_flip_noise(encoded_0)
print("Error correction protocol: detect and correct single bit
flips (conceptual).")
```