TASK 2: Pauli Matrices and Eigenvalues/Eigenvectors

Aim:
To analyze Pauli matrices through application on qubit states and eigenvalue decomposition.
Algorithm:
· Define Pauli-X, Y, and Z matrices.
· Apply these matrices to |0⟩ and |1⟩ states.
· Use linear algebra to compute eigenvalues and eigenvectors.
· Print matrix properties.

Program :-
```
import numpy as np
from numpy.linalg import eig

print("\n" + "="*50)
print("TASK 2: PAULI MATRICES AND EIGEN-ANALYSIS")
print("="*50)

# Define Pauli matrices
pauli_x = np.array([[0, 1], [1, 0]])
pauli_y = np.array([[0, -1j], [1j, 0]])
pauli_z = np.array([[1, 0], [0, -1]])

print("Pauli-X matrix:")
print(pauli_x)
print("\nPauli-Y matrix:")
print(pauli_y)
print("\nPauli-Z matrix:")
```

```
print(pauli_z)

# Define qubit states
qubit_0 = np.array([1, 0])  # |0⟩
qubit_1 = np.array([0, 1])  # |1⟩

print("\nApplying Pauli-X to |0⟩:", pauli_x @ qubit_0)
print("Applying Pauli-X to |1⟩:", pauli_x @ qubit_1)

# Eigen-analysis function
def analyze_operator(matrix, name):
    eigenvals, eigenvecs = eig(matrix)
    print(f"\n{name} Eigenvalues:", eigenvals)
    print(f"{name} Eigenvectors:")
    for i, vec in enumerate(eigenvecs.T):
        print(f"  λ = {eigenvals[i]} : {vec}")

# Analyze all Pauli matrices
analyze_operator(pauli_x, "Pauli-X")
analyze_operator(pauli_y, "Pauli-Y")
analyze_operator(pauli_z, "Pauli-Z")
```