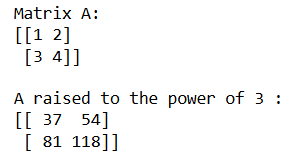
**Task-2:** **Statistical Analysis**

1. **Description**

Implement matrix exponentiation for a square matrix using numpy.

1. **Output**

****

1. **Algorithm Used in Task**

This code demonstrates how to compute the power of a square matrix using NumPy's matrix\_power function. It simplifies matrix exponentiation, a fundamental operation in linear algebra, by automating the repeated multiplication of the matrix by itself. The result is calculated efficiently and displayed clearly, making it easy to understand the behavior of matrix powers.

**Explanation of Libraries and Their Usage:**

**NumPy (import numpy as np)**

* + A library used for numerical computing in Python.
  + Provides support for multi-dimensional arrays and mathematical functions.
  + **Key Function Used:**
    - np.array: Creates a NumPy array, representing the square matrix A.
    - np.linalg.matrix\_power: Computes the matrix A raised to a given power n.

**Approach:**

1. Define a square matrix A using NumPy.
2. Specify the exponent n to which A should be raised.
3. Use the function np.linalg.matrix\_power() to compute An.
4. Print the input matrix AAA and the resulting matrix An.

**Algorithm:**

1. Import the NumPy library.
2. Define a square matrix A as a 2D NumPy array.
3. Specify the exponent n.
4. Use np.linalg.matrix\_power(A, n) to compute the result of multiplying A by itself n times.
5. Display the original matrix A and the computed result An using print() statements.