**Experiment 1**

**Aim: Create Tensors and perform basic operations with tensors**

**Description :**

Tensors are a generalization of scalars, vectors, and matrices to higher dimensions. Tensors are often denoted by boldface letters (e.g., T). The order (or rank) of a tensor is the number of dimensions.

For example:

1. A scalar is a 0th-order tensor.
2. A vector is a 1st-order tensor.
3. A matrix is a 2nd-order tensor.
4. A 3rd-order tensor might look like a cube of numbers, and so on.

The program imports the NumPy library, which is essential for numerical computations in Python. NumPy provides support for large multi-dimensional arrays and matrices along with a collection of mathematical functions to operate on these arrays.

**Different Types of Operations Performed**

Array Creation:

* Function: numpy.array
* Description: The program initializes a 2D NumPy array with specific values. This is the first step in setting up data structures needed for numerical computations.

Element-wise Multiplication:

* Function: numpy.multiply
* Description: Each element of the array is multiplied by a scalar value. This operation scales the values within the array uniformly, which is useful for normalization and other data transformations.

Element-wise Addition:

* Function: numpy.add
* Description: A scalar value is added to each element of the array. This shifts all the values in the array by a constant amount, which can be useful for adjusting data ranges or centering data around a specific value.

Element-wise Exponentiation:

* Function: numpy.power
* Description: Each element of the array is raised to a specified power. This operation is used for mathematical transformations such as squaring each element, which can help in feature engineering tasks in machine learning.

Array Transposition:

* Function: numpy.transpose
* Description: The array is transposed, meaning its rows and columns are swapped. Transposition is a common operation in linear algebra and is essential for aligning data in matrix computations and preparing data for various algorithms.

**Application**

1. Data Preprocessing: Cleaning and transforming raw data into a suitable format for analysis or machine learning models.
2. Mathematical Computations: Performing efficient numerical operations necessary for implementing algorithms and conducting experiments.
3. Feature Engineering: Creating and modifying features to improve the performance of machine learning models.
4. Scientific Computing: Conducting simulations and experiments that require complex numerical computations.