Data Structure Using Python Assignment 1.

Name: Mohammed. Janaliye.

Q.1.   why we require Numpy? (Mention at least 4 points).

Numpy is an essential library in python for scientific computing and data analysis.

i. Efficient multidimensional Array operations:
Numpy provides support for large, multi-dimensional arrays and matrices. It offers a wide range of mathematical functions that operate on these array efficiently, making it possible to perform complex operations with minimal code and maximum performance.

ii. Performance.
Numpy arrays are more efficient than python lists for numerical operations because they are implemented in c and can perform operations faster by avoiding the overhead of python loops. Numpy uses contiguous memory blocks which makes processing large data sets faster and more efficient.

iii. Vectorization.
Numpy allows for vectorized operations, meaning that operations on entire arrays can be performed without the need for explicit loops in Python. This leads to cleaner, more readable code and often significant performance improvements.

iv. Integration with Other Libraries.

Numpy is the Foundational package for many other scientific libraries in Python, Such as Scipy, Pandas, Matplotlib, and Tensorflow. These libraries rely on Numpy arrays for handling data, making Numpy indispensable for any kind of scientific computing or data analysis in Python.

Q.2 what is the difference between list, tuple, array?

a. List.

   i. Definition: A list is a mutable, ordered collection of items in Python. Items in a list can be of different types, and the list itself can be modified (e.g., elements can be added or removed).

   ii. Syntax: List are created using square brackets e.g., my_list = [1, 2, 3, "apple"].

   iii. Mutability: List are mutable, meaning that their contents can be changed after creation.

   iv. Use case: List are versatile and can store mixed data types, making them suitable for general-purpose collection

b. Tuple

   i. Definition: A tuple is an immutable, ordered collection of items in Python. Like lists, items in a tuple can be of different types.

ii. **Syntex:** Tuples are created using parantheses, e.g., my_tuple = (1, 2, 3, "apple").

iii. **Mutability:** Tuples are immutables, meaning that their contents cannot be changed after creation.

iv. **Use Case:** Tuples are used when a fixed collection of items in needed, where the integrity of the data should be maintained.
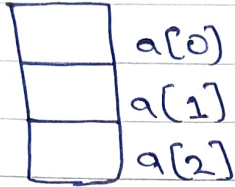
C. **Array (Numpy Array):**

i. **Definition:** An array in Numpy is a grid values of the same type, and is indexed by a tuple of non-negative integers. Arrays in Numpy are homogenous, meaning all elements must be of the same data type.

ii. **Syntax:** Arrays are created using the numpy.array() function. e.g., my_array = np.array([[1, 2, 3]]).

iii. **Mutability** ~~stability~~: Numpy arrays are mutable, meaning that their contents can be changed. However, they are more efficient in terms of storage and performance for numerical computations.

iv. Use Case: Arrays are used in scientific computing and data analysis where large datasets need to be processed efficiently, especially when performing element-wise operations, mathematical functions, or linear algebra.
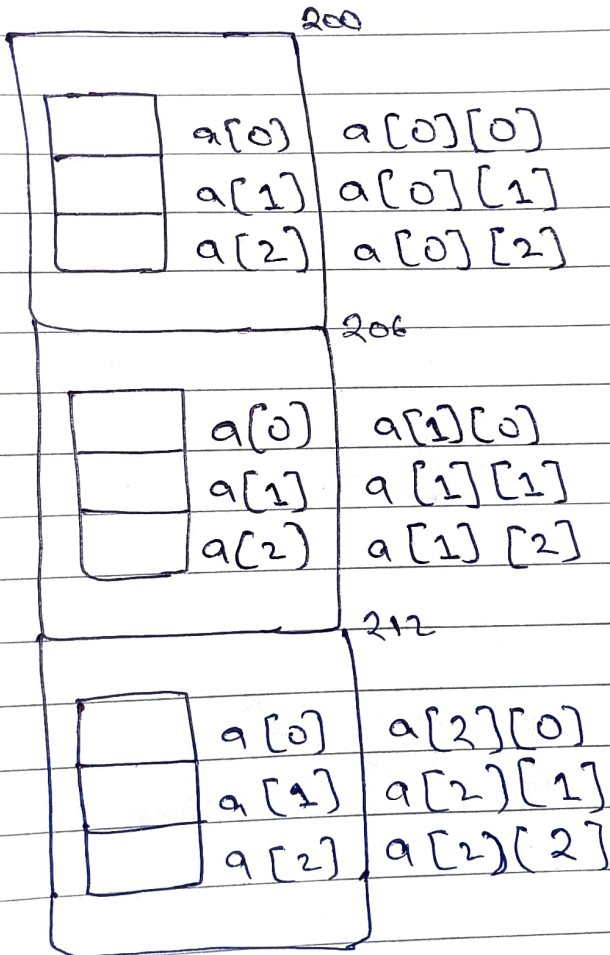
Arrays naming.

→ a[3]   1-D array

a[0]
a[1]
a[2]

→   2-D arrays.

a[3][3]   200

a[0]   a[0][0]
a[1]   a[0][1]
a[2]   a[0][2]

                          206

a[0]   a[1][0]
a[1]   a[1][1]
a[2]   a[1][2]

                          212

a[0]   a[2][0]
a[1]   a[2][1]
a[2]   a[2][2]

→ 3-D arrays

a[3][3][3]

	a[0]	a[0][0]	a[0][0][0]
	a[1]	a[0][1]	a[0][0][1]
	a[2]	a[0][2]	a[0][0][2]
	a[0]	a[1][0]	a[0][1][0]
	a[1]	a[1][1]	a[0][1][1]
	a[2]	a[1][2]	a[0][1][2]
	a[0]	a[2][0]	a[0][2][0]
	a[1]	a[2][1]	a[0][2][1]
	a[2]	a[2][2]	a[0][2][2]
	a[0]	a[0][0]	a[1][0][0]
	a[1]	a[0][1]	a[1][0][1]
	a[2]	a[0][2]	a[1][0][2]
	a[0]	a[1][0]	a[1][1][0]
	a[1]	a[1][1]	a[1][1][1]
	a[2]	a[1][2]	a[1][1][2]
	a[0]	a[2][0]	a[1][2][0]
	a[1]	a[2][1]	a[1][2][1]
	a[2]	a[2][2]	a[1][2][2]
	a[0]	a[0][0]	a[2][0][0]
	a[1]	a[0][1]	a[2][0][1]
	a[2]	a[0][2]	a[2][0][2]
	a[0]	a[1][0]	a[2][1][0]
	a[1]	a[1][1]	a[2][1][1]
	a[2]	a[1][2]	a[2][1][2]
	a[0]	a[2][0]	a[2][2][0]
	a[1]	a[2][1]	a[2][2][1]
	a[2]	a[2][2]	a[2][2][2]