Assignment - 4

1. Why list occupies 2 blocks of memory?

A **list** in Python is **mutable**

Its elements can change, can be added or can be removed.
So, Python stores a list in **two parts**:

1. **One block**: stores information like the list's length, type, and a pointer to its elements.
2. **Second block**: stores pointers to each actual element.

Example:

a = [10, 20, 30]

- The **first block** stores: "I am a list of length 3."
- The **second block** stores: pointers to 10, 20, 30.

Here, one for the list structure, one for its data

2. Why Tuple occupies one block of Memory?

A **tuple** is **immutable** once created it never changes.

Python can store all the elements **in a single, fixed block of memory**, without needing a separate flexible structure.

Example:

b = (10, 20, 30)

It stored in one continuous block since it will never change.

3. Describe Hashable and Unhashable Data Types?

Hashable:

- It Has a **fixed memory address/value**
- It never changes
- It Can be used as a **key in dictionaries** or an element in **sets**

Hashable (immutable) types:

- int
- float
- string
- tuple
- bool

Example:

my_dict = {(1, 2): "tuple key", "name": "Priyanka"}

Unhashable (mutable) types:

- list
- set
- dict

These can't be used as dictionary keys or in sets because their contents can change that would break the hash.

Example:

my_dict = {[1, 2]: "list key"} # Error: unhashable type: 'list'