Ans-1 :-

Differences Between Lists and Tuples in Python

Feature	**List (list)**	**Tuple (tuple)**
Mutability	Mutable (can be modified)	Immutable (cannot be modified after creation)
Performance	Slower (because of dynamic resizing and mutability overhead)	Faster (optimized due to immutability)
Memory Usage	Higher (extra memory for modifications)	Lower (no modification overhead)
Methods Available	Many methods (append(), remove(), etc.)	Fewer methods (count(), index())
Iteration Speed	Slower (due to dynamic nature)	Faster (fixed structure)
Use Case	When modifications are needed (e.g., dynamic data structures)	When values should remain constant (e.g., fixed configurations, dictionary keys)

When to Choose a List vs. a Tuple?

Choose a list when:

- You need to modify, add, or remove elements frequently.
- The data structure is dynamic.
- You need built-in list operations (like .append(), .pop(), etc.).

Choose a tuple when:

- You need **faster execution** (e.g., for large datasets).
- The data should **not be changed** after creation.
- You want to use it as a **dictionary key** (since tuples are hashable, unlike lists).
- You need to ensure data **integrity and security**.

Ans-2 :-

Type Conversions in Python

Conversion Type	**Function**	**Example**	**Result**
Int → Float	**float(x)**	**float(5)**	**5.0**
Float → Int	**int(x)**	**int(3.9)**	**3**
Number → String	**str(x)**	**str(100)**	**"100"**

Conversion Type	Function	Example	Result
String → Int	int(x)	int("42")	42
String → Float	float(x)	float("3.14")	3.14
String → List	list(x)	list("abc")	['a', 'b', 'c']
List → String	"".join(list)	"".join(['a', 'b', 'c'])	"abc"