2) List Operations.

- **Common list operations: concatenation, repetition, membership.**

Concatenation :

The process of combining two or more lists to create a new list.

- Key Characteristics:
  - Uses the + operator.
  - Does not modify the original lists.
  - Creates a new list containing all elements from both lists in order.
  - Time complexity: O(n) where n is the total number of elements.

Repetition :

Creating a new list by repeating the elements of an existing list multiple times.

- Key Characteristics:

    - Uses the * operator with an integer.
    - The original list remains unchanged.
    - The integer determines how many times the list is repeated.
    - Time complexity: O(n*k) where n is list length and k is repetition count.

Membership :

Checking whether an element exists in a list.

- Key Characteristics:

    - Uses in and not in operators.
    - Returns a boolean value (True/False).
    - Performs linear search (checks elements one by one).
    - Time complexity: O(n) in worst case.
    - Case-sensitive for strings.
    - Uses equality comparison (==) for matching.

- **Understanding list methods like append(), insert(), remove(), pop().**

1. **append()**

Adds a single element to the end of the list.

Characteristics:

- Modifies the original list.
- Accepts exactly one argument.
- Time complexity: O(1) (amortized).
- No return value (returns None).

```
list.append(element)
```

2. **insert()**

Inserts an element at a specified position.

Characteristics:

- Takes two arguments: index and element
- Shifts subsequent elements to the right
- Time complexity: O(n) in worst case

- Accepts negative indices (counts from end)
- No return value (returns None)

```
list.insert(index, element)
```

3. remove()

Removes the first occurrence of a specified value.

Characteristics:

- Takes one argument: value to remove.
- Raises ValueError if element not found.
- Time complexity: O(n).
- Performs linear search for the value.
- No return value (returns None).

```
list.remove(element)
```

4. pop()

Removes and returns an element at a given index.

Characteristics:

- Optional index argument (default: -1).
- Returns the removed element.
- Raises IndexError for invalid indices.
- Time complexity: O(1) for end, O(n) for other positions.
- Without arguments, removes last element.

```
list.pop(index)
```