**Assignment on Collection (List)**

1. Remove Duplicates from an ArrayList (Without using Set or Streams).

Input: [10, 20, 10, 30, 20, 40] Expected Output: [10, 20, 30, 40]

Hint: Use a nested loop or a new ArrayList with contains() check

2. Rotate an ArrayList k Times (Circular Rotation).

Input: List = [1, 2, 3, 4, 5], k = 2 Expected Output: [4, 5, 1, 2, 3]

Hint: Use modulo and sublists.

3. Top K Frequent Elements using ArrayList & HashMap.

Input: List = [4, 4, 1, 2, 2, 2, 3, 3], k = 2 Expected Output: 2 → 3 times 4 → 2 times

Hint: Count frequencies using HashMap, sort by values.

4. First Non-Repeating Element in an ArrayList.

Input: [3, 5, 3, 4, 5, 6] Expected Output: 4

Hint: Use HashMap to count frequencies, then iterate list.

5. Find All Pairs with Given Sum (Two-Sum Variant).

Input: List = [2, 4, 3, 5, 7], sum = 7 Expected Output: (2, 5) and (4, 3)

Hint: Use HashSet to store complements.

6. Merge Two Sorted ArrayLists into One Sorted ArrayList.

Input: arraylist\_1 = [1, 3, 5, 7] , arraylist\_2 = [2, 4, 6, 8] Expected Output: [1, 2, 3, 4, 5, 6, 7, 8]

Hint: Use two-pointer technique.

7. Find the Longest Increasing Subsequence in an ArrayList.

Input: [10, 9, 2, 5, 3, 7, 101, 18] Expected Output: [2, 3, 7, 101]

Hint: Use Dynamic Programming or Binary Search approach.

8. Remove All Strings with Length < 3 .

Input: ["a", "ab", "abc", "java", "is", "fun", "ok", "yes"] Expected Output: [abc, java, fun, yes]

Hint: Use Iterator or removeIf().

9. Count Occurrences of Each Element in an ArrayList.

Input: ["apple", "banana", "apple", "orange", "banana", "apple"]

Expected Output: apple → 3 banana → 2 orange → 1

Hint: Use HashMap to store frequencies.