UNITED INTERNATIONAL UNIVERSITY

Department of Computer Science and Engineering (CSE)

CT-01

50 mins

Set A

Course : DSA I Lab | **Course Code: CSE 2216**

Trimester & Year: Spring 2025 | **Section: D**

Name : | **Total Marks: 30**

ID:

SN	Questions	Marks
1		10

You are given a sorted list of integers in non-decreasing order and an integer `x`. Your task is to determine how many times `x` occurs in the list.

Input:

- The first line contains a single integer `n` — the number of elements in the list.
- The second line contains `n` integers `a_1, a_2, ..., a_n`, where the list is sorted in non-decreasing order.
- The third line contains a single integer `x` — the number whose occurrences you need to count.

Output:

- Print a single integer — the number of times `x` appears in the list.

Constraints:

- Time complexity must be less than or equal to O(logn)

Sample Input	Sample Output
6 1 2 2 2 3 4 2	3
Explanation: "listen" can be rearranged to form "silent".	
5 1 1 1 1 1 2	0

Marking Criteria

Logic	**4**
Implementation	**4**
Overall correctness	**2**

SN	Questions	Marks
2	You are given a singly linked list where the elements are sorted in non-decreasing order. Your task is to implement two functions: 1. **Remove Duplicates from a Sorted Linked List**: Modify the linked list so that all duplicate elements are removed, leaving only distinct elements from the original list. 2. **Detect Cycle in a Linked List**: Determine whether the linked list contains a cycle. A cycle occurs when a node's next pointer points to a previous node in the list, forming a loop.	**10*2**