National University

Of Computer and Emerging Sciences

CL2001 – Data Structure Lab
Lab Task # 04

Note:

- Plagiarism will not be **tolerated!!**
- Variable names should be meaningful.
- Use comments wherever applicable.
- Submit a pdf file containing all of your C++ code with all possible screenshots of every task output on Google Classroom. The name of file should be your roll no followed by your name (roll-no-name.pdf) i.e., (23P-1234-Ali.pdf).

Problem: 1

Suppose you are tasked with creating a student registration system for a university using C++. Each student record includes the student's ID number, name, and GPA. You need to implement this system using a singly circular linked list data structure for efficient management of student records.

1. Design a class to represent a student record. It should include private members for the student's ID, name, and GPA, along with appropriate member functions.

2. Develop a class to manage the linked list of students. Include functions to add a student, remove a student by ID, display all student records, search for a student by ID and display their information, and calculate the average GPA of all students.

3. Write a sample program demonstrating the usage of your class. The program should allow users to interactively add students, remove students, search for students, display all student records, and calculate the average GPA.

Ensure that your implementation gracefully handles edge cases such as an empty list or attempting to remove a non-existent student.

Sample Output:

```
Welcome to the Student Registration System!

1. Add a student
2. Remove a student
3. Search for a student
4. Display all students
5. Calculate average GPA
6. Exit

Enter your choice: 1
Enter student ID: 1001
Enter student name: John Doe
Enter student GPA: 3.8
```

Student added successfully.

Enter your choice: 1
Enter student ID: 1002
Enter student name: Jane Smith
Enter student GPA: 3.5
Student added successfully.

Enter your choice: 1
Enter student ID: 1003
Enter student name: Alice Johnson
Enter student GPA: 3.9
Student added successfully.

Enter your choice: 4
Student Records:
ID: 1001, Name: John Doe, GPA: 3.8
ID: 1002, Name: Jane Smith, GPA: 3.5
ID: 1003, Name: Alice Johnson, GPA: 3.9

Enter your choice: 3
Enter student ID to search: 1002
Student found:
ID: 1002, Name: Jane Smith, GPA: 3.5

Enter your choice: 2
Enter student ID to remove:
1002 Student removed
successfully.

Enter your choice: 4
Student Records:
ID: 1001, Name: John Doe, GPA: 3.8
ID: 1003, Name: Alice Johnson, GPA: 3.9

Enter your choice: 5
Average GPA: 3.85

Enter your choice: 6
Goodbye!

Problem: 2

Create two linked lists. After creating the two linked lists, find how many values are common in both linked lists.

Example :

Linked List 1: 10 → 20 → 30 → 40 → 50

Linked List 2: 30 → 50 → 70 → 90

Common values: 30, 50

Total common values = 2

Problem: 3

Create a Singly Circular linked list. After creating the list, remove all duplicate values so that each element appears only once.

Example:

Singly Circular Linked List (before): 10 → 20 → 30 → 20 → 40 → 30

Singly Circular Linked List (after removing duplicates): 10 → 20 →30 → 40