Assignment - 1

CSE 2216 (L)

Some instructions:

- Try to solve the problems yourself. If you face difficulties, you may reach out to our Lab UA for guidance.
- Each problem carries the marks assigned to it.
- For each code, **include the given example run** showing how your solution works. This demonstration will carry **0.5 marks** in your assignment.
- SLL = Singly Linked List, DLL = Doubly Linked List
- Do not copy codes from others; otherwise, marks will be deducted.
- Submit within the deadline; late submissions will result in a deduction of marks.
    - Last date: **September 20, 2025**

Exercise - 1 [2.5]

Create a function using SLL that will merge two SLLs into a single sorted SLL.

- The function will take two SLL as input.
- Based on this input, the function will return a sorted SLL.

Example 1:

Input: 30 → 10 → 20 → 50 → 80, 70 → 90 → 60 → 100 → 40
Output: 10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100

Example 2:

Input: 30 → 10 → 20, 30 → 10 → 20
Output: 10 → 10 → 20 → 20 → 30 → 30

Exercise - 2 [2.5]

Create a function using SLL that will delete duplicate values.

- You can assume the input SLL is already sorted.
- The function will take the sorted SLL as input.
- It will return a sorted SLL where every value is unique.

Example 1:

Input: 10 → 20 → 30 → 40 → 50 (No change)
Output: 10 → 20 → 30 → 40 → 50

Example 2:

Input: 10 → 10 → 20 → 20 → 30 → 30
Output: 10 → 20 → 30

Exercise - 3 [2.5]

Create a function using a DLL that will check if the DLL is a palindrome.

- You may use a tail pointer.
- Do not create an extra DLL; check using the same DLL.
- The function will take the DLL as input.
- It will return `True` if the DLL is a palindrome, otherwise `False`.

Example 1:

Input: 50 ↔ 20 ↔ 30 ↔ 20 ↔ 50
Output: True

Example 2:

Input: 50 ↔ 20 ↔ 20 ↔ 50
Output: True

Example 3:

Input: a ↔ b ↔ c ↔ c ↔ a
Output: False

Exercise - 4 [2.5]

Create a function using a DLL that will perform bubble sort.

- You may use a tail pointer.
- The function will take the DLL as input.
- It will return the DLL sorted in ascending order.

Example 1:

Input: 4 ↔ 2 ↔ 5 ↔ 1 ↔ 3
Output: 1 ↔ 2 ↔ 3 ↔ 4 ↔ 5