LAB-4

Question 1: Detecting Cycles in TEDx-IIITDelhi Event Planning During the planning phase, TEDx-IIITDelhi uses a dependency list where each task points to the next one. However, an accidental loop in the task list can lead to an infinite dependency cycle, preventing the event from moving forward. Write a function to detect if such a cycle exists in a linked list representing tasks.
Example: 6 Input: Task1 -> Task2 -> Task3 -> Task4 -> Task2 (loop back)
Output: Cycle detected

Question 2: Write a function reverseDList() which takes a doubly linked list and reverses it.
For example: Input: 1 <-> 2 <-> 3 <-> 4 <-> 5 Output: 5 <-> 4 <-> 3 <-> 2 <-> 1

Question 3: Imagine you are managing two separate trains running on parallel tracks. Each train has passenger coaches arranged in ascending order based on seat numbers. Your job is to merge these two trains into a single train while ensuring that the seat numbers remain in sorted order.

- Both List are sorted in ascending order
- Merge list must maintain the sorted order

Train A (Linked List 1): 1 → 3 → 5 → 7
Train B (Linked List 2): 2 → 4 → 6 → 8
After merging: Final Train: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8