CL2001 – Data Structure Lab

Home Work # 01

Linked List

Note: Carefully read the following instructions.

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Submit a pdf file containing all of your C++ code with all possible screenshots of every task outputs. Submit all .cpp files as well on Google Classroom.
7. First think about statement problems and then write/draw your logic on copy.
8. After copy pencil work, code the problem statement.
9. Please submit your file in this format (20P-8743-Zain).
10. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem: 1 | Merging Ordered Lists

Write a function that merges two ordered list objects of integers into a single ordered list object of integers. Function merge should receive references to each of the list objects to be merged and reference to a list object into which the merged elements will be placed.

Problem: 2 | Copying a List in Reverse Order

Write a program that creates a linked list object of 10 characters and creates a second list object containing a copy of the first list, but in reverse order.

Problem: 3 | Inserting middle

Write a program that creates an even number of links in a link list and then finds the middle of the link list and insert a data item in the list.

Example:

Given: 1->2->3->4->NULL

Inserting 7 at the middle of linked list

Updated: 1->2->**7**->3->4->NULL

Problem: 4 |

Write a C++ program to move the last node to the front of a Singly Linked List.

Example:

Given: 1->2->3->4->NULL

Return: 4->1->2->3->NULL

Problem: 5 | Linked List the Palindrome

Write a function to check whether the given Singly Linked List is Palindrome or not.

Problem: 6 | Remove Duplicates

Write a RemoveDuplicates() function which takes a list sorted in increasing order and deletes any duplicate nodes from the list. Ideally, the list should only be traversed once.
