

CL2001 – Data Structure Lab

Lab Task # 03

Note:

- Copied task will be awarded **zero** marks.
- Use comments wherever applicable.
- Submit a pdf file containing all 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., (24P-1234-Ali.pdf).
- Variables and functions names should be meaningful.

Problem: 1

Write a program to implement a singly linked list in C++. The program should allow the user to insert new node after a specific target value, but only when it appears for the third time in the list.

Instructions:

First, create a singly linked list containing integer values.

Ask the user to enter the target value.

Traverse the list and insert new node after third occurrence of the target value (if it exists). Display the updated linked list after insertion.

Example:

Given the linked list:

$2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 4$

If the user enters the target value 4, then the new node will insert after the third occurrence of 4. The resulting linked list should be:

$2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 4$

Problem: 2

Write a program to implement a singly linked list in C++. Your task is to insert new node after the middle element of the linked list.

Instructions:

Assume linked list contain total odd number of nodes.

Create a singly linked list containing integer values.

Traverse the list to determine and print the middle element.

Example:

For the linked list:

10 → 20 → 30 → 40 → 50

The middle element is: 30

Problem: 3

Write a program to implement a singly linked list in C++. Your task is to count how many nodes meet the following two conditions:

The node contains an even number and is located at an even index (starting from index 0).

The node contains an odd number and is located at an odd index.

Instructions:

Create a singly linked list containing integer values.

Traverse the list and count:

How many nodes have even values and are at even

indices. How many nodes have odd values and are at odd

indices. Display both counts at the end.

Example:

For the linked list:

3 → 4 → 6 → 9 → 8 → 11 → 14 → 15

The result might be:

Even numbers at even indices: 2

Odd numbers at odd indices: 3