## Page 1

CC-213L – Data Structures and Algorithms

Fall 2024

LAB-09

Issue Date: May 9, 2025

Faculty of Computing and Information Technology

Dr. Madiha Khalid

Page 2 of 4

The objective of this lab is to:

Derive and write recursive solutions by identifying base cases and recursive cases, ensuring

correctness and efficiency.

ALERT!

1. This is an individual lab. You are strictly NOT allowed to collaborate with others, share screens, or

communicate answers in any form.

2. Use of AI tools (e.g., ChatGPT, Copilot, etc.) is strictly prohibited. Any AI-generated content will

be treated as academic dishonesty.

3. Anyone caught in act of cheating would be awarded an “F” grade in this Lab.

Task 01:

[30 Marks]

Rotate Linked List in Chunks (Singly) Using Recursion

Problem Statement

Write a recursive function to rotate a singly linked list in chunks of size k. Each chunk of k nodes should be

reversed, and the entire list should be processed recursively. If the number of nodes is not a multiple of k, the

remaining nodes should remain in their original order.

Function Prototype

Node\* rotateListInChunks (Node\* head, int k); // global function

Sample Execution 1

Input:

Head: 1 → 2 → 3 → 4 → 5 → 6 → NULL

k = 2

Output:

2 → 1 → 4 → 3 → 6 → 5

Explanation:

Chunks [1,2] → [2,1], [3,4] → [4,3], [5,6] → [6,5].

Sample Execution 2

Input:

Head: 1 → 2 → 3 → 4 → 5 → NULL

k = 3

Output:

3 → 2 → 1 → 4 → 5 → NULL

Explanation:

Chunk [1,2,3] → [3,2,1], [4,5] remains as it is.

Sample Execution 3

Input:

Head: 1 → 2 → 3,

k = 5

Output:

1 → 2 → 3

Explanation:

Since k > list length, no rotation occurs.

## Page 2

CC-213L – Data Structures and Algorithms

Fall 2024

LAB-09

Issue Date: May 9, 2025

Faculty of Computing and Information Technology

Dr. Madiha Khalid

Page 2 of 4

Task 02:

[30 Marks]

Check for Balanced Parentheses Without Stack

Problem Statement

Write a recursive function to check if a string containing parentheses ( and ) is balanced. Instead of using an explicit

stack, use the call stack to simulate the counting of open and closed parentheses.

Function Prototype

bool isBalancedParentheses (string& str);

// global function

Input

A string containing parentheses ( and ).

Output

true if the parentheses are balanced, false otherwise.

Sample Executions



Input: "((()))"

Output: true

Explanation: All parentheses are properly nested.



Input: "(()"

Output: false

Explanation: There is an unmatched opening parenthesis.



Input: ")("

Output: false

Explanation: A closing parenthesis appears before an opening one.

Task 03:

[30 Marks]

Evaluate Postfix Expression Using Stack Recursively

Problem Statement

Write a recursive function to evaluate a postfix expression using the call stack to simulate a stack. A postfix

expression is a mathematical expression where operators follow their operands (e.g., 2 3 + means 2 + 3).

Function Prototype

int evaluatePostfix (const string& expression);

// global function

Input

expression: A string representing a valid postfix expression with single-digit operands and operators +, -, \*, /.

Output

The result of evaluating the postfix expression as an integer.

Sample Executions



Input: "2 3 1 \* + 9 -"

Output: -4

Explanation: (2 + (3 \* 1)) - 9 = (2 + 3) - 9 = 5 - 9 = -4.



Input: "5 1 2 + 4 \* + 3 -"

Output: 14

Explanation: (5 + ((1 + 2) \* 4)) - 3 = (5 + (3 \* 4)) - 3 = (5 + 12) - 3 = 17 - 3 = 14.

## Page 3

CC-213L – Data Structures and Algorithms

Fall 2024

LAB-09

Issue Date: May 9, 2025

Faculty of Computing and Information Technology

Dr. Madiha Khalid

Page 2 of 4

Task 04:

[30 Marks]

Permutations of a String

Problem Statement

Write a recursive function to generate all permutations of a given string by swapping characters. A permutation is a

rearrangement of the characters in the string.

Function Prototype

vector<string> generatePermutations (const string& str);

// global function

Input

str: A string containing distinct characters.

Output

A vector of strings containing all possible permutations of the input string.

Sample Executions



Input: str = "abc"

Output: ["abc", "acb", "bac", "bca", "cab", "cba"]

Explanation: All possible arrangements of characters a, b, c.



Input: str = "ab"

Output: ["ab", "ba"]

Explanation: Two possible arrangements.



Input: str = "a"

Output: ["a"]

Explanation: Only one permutation for a single character.

Task 05:

[30 Marks]

Check if a Matrix is Symmetric Using Recursion

Problem Statement

Write a recursive function to check if a given square matrix is symmetric. A matrix is symmetric if it is equal to its

transpose (i.e., matrix[i][j] == matrix[j][i] for all i, j).

Function Prototype

bool isSymmetricMatrix (int \*\*matrix, int n);

Input



matrix: A 2D vector representing an n x n square matrix.



n: The size of the matrix (number of rows/columns).

Output

true if the matrix is symmetric, false otherwise.

Sample Executions



Input:

[[1, 2, 3],

[2, 4, 5],

[3, 5, 6]]

Output:

True

Explanation: The matrix is symmetric as matrix[i][j] == matrix[j][i] for all i, j.

## Page 4

CC-213L – Data Structures and Algorithms

Fall 2024

LAB-09

Issue Date: May 9, 2025

Faculty of Computing and Information Technology

Dr. Madiha Khalid

Page 2 of 4



Input:

[[1, 2, 3],

[2, 4, 5],

[4, 5, 6]]

Output:

false

Explanation: matrix[2][0] != matrix[0][2] (4 != 3).



Input:

[[1]]

Output:

true

Explanation: A 1x1 matrix is always symmetric.