Data Structures and Algorithms Lab
CS-F23

Issue Date: March 03, 2025

Total Marks: 40

LAB-05

Start Time: 11:45 PM
End Time: 12:45 PM

The objective of this lab is to:

Understanding of double and circular linked lists

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) You are **not allowed** to use any external libraries.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

Task 01(Doubly Linked List Node Splitting) [20 Marks]

In this problem, each node in the doubly linked list contains a string value. Your task is to **split** each node into **multiple nodes** if the string contains **spaces**. For example, if a node contains "hello world", it should be split into two nodes: one containing "hello" and the other containing "world". After splitting, ensure the prev and next pointers are updated correctly.

Function Prototype: *void splitNodes (Node* head)*

Example:

Input:

"hello world" <-> "this is" <-> "a test"

Output:

"hello" <-> "world" <-> "this" <-> "is" <-> "a" <-> "test"

Constraints:

- Split nodes in-place without creating a new list.
- Handle edge cases like empty strings or nodes with no spaces.

Task 02(Circular Coin Distribution) [20 Marks]

You are given a file (input.txt) containing a **list of houses**, each with some coins. The houses are arranged in a circular manner. The goal is to redistribute the coins so that every house ends up with the **same number of coins** by moving them **only to neighboring houses** (left or right).

You need to find the minimum number of moves required to achieve this.

Function Prototype: *int minimumMoves(Node*head)*

Movement Rules:

- You can only move coins to the next or previous house.
- A move consists of transferring one coin at a time from one house to its neighbor.
- If equal distribution is not possible, return -1.

Input Format (File: input.txt**)**

- The first number in the file represents the **total** number of houses (n).
- The next n numbers represent the **number of coins** in each house.

Provided Code:

You do not need to write the file reading and linked list creation part. Just use the provided function.

➡

```
Node* readAndCreateList(const string &filename)
{
        ifstream inputFile(filename);
        if (!inputFile)
        {
                cout << "Error opening input file!" << endl;
                return nullptr;
        }

        int n;
        inputFile >> n; // Read number of houses

        if (n <= 0)
              return nullptr;

        Node *head = nullptr, *tail = nullptr;
        int val;

        for (int i = 0; i < n; i++)
        {
                inputFile >> val;
                Node *newNode = new Node(val);

                if (!head)
                     head = tail = newNode;
                else
                {
                     tail->next = newNode;
                     newNode->prev = tail;
                     tail = newNode;
                }
        }
        // Make the list circular
        tail->next = head;
        head->prev = tail;

        inputFile.close();
        return head;
}
```

Constraints:

- Use the provided linked list code for input conversion.
- Do not convert the linked list into an array.
- Your function should work for large inputs

Example:

Input:

Houses: [1, 0, 5]

Output:

Minimum moves required: 4

Explanation:

- Move 1 coin from house 3 to house 2 → [1, 1, 4]
- Move 1 coin from house 3 to house 2 → [1, 2, 3]
- Move 1 coin from house 2 to house 1 → [2, 1, 3]
- Move 1 coin from house 3 to house 2 → [2, 2, 2]