Assignment 3
Deadline: 5:30 PM on Dec 4

For Assignment 3, you will work in groups of up to three members. Ensure you form a different group from your assignment 1 & 2. If you prefer to work alone, that's acceptable as well.

Requirements

Imagine you are developing a system for a hospital to manage its patient records. The hospital staff needs to add new patients, remove discharged patients, and display the current list of patients. Your task is to implement this system using linked lists.
Tasks:

1. **Linked List Implementation:**
   - Implement a singly linked list in C.
   - Include operations for:
     1. **Insertion (at the beginning, end, and a specific position)**

        a. At the Beginning: Add a new patient at the start of the linked list.

        Before: Head -> [Patient1] -> [Patient2] -> NULL

        Insert "NewPatient" at the beginning

        After: Head -> [NewPatient] -> [Patient1] -> [Patient2] -> NULL

        b. At the End: Add a new patient at the end of the linked list.

        Before: Head -> [Patient1] -> [Patient2] -> NULL

        Insert "NewPatient" at the end

        After: Head -> [Patient1] -> [Patient2] -> [NewPatient] -> NULL

        c. At a Specific Position: Add a new patient at a specified position in the linked list.

        Before: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

        Insert "NewPatient" at position 2

        After: Head -> [Patient1] -> [Patient2] -> [NewPatient] -> [Patient3] -> NULL

     2. **Deletion (by value and by position)**

        a. By Value: Remove the first patient that matches the specified name.

        Before: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

        Delete patient with name "Patient2"

        After: Head -> [Patient1] -> [Patient3] -> NULL

        b. By Position: Remove the patient at the specified position.

        Before: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

Delete patient at position 1

After: Head -> [Patient1] -> [Patient3] -> NULL

3. **Display the list**

   Display Traverse the linked list and print the name of each patient.

   List: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

   Output: Patient1, Patient2, Patient3

2. **Testing:**
   - Create a test suite to validate your linked list operations.
   - Include test cases for inserting, deleting, and displaying elements.

Example Test Case

Insertion at End:

- **Input:** Insert “Patient4” at the end of the list.
- **Expected Output:**

  Before: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

  After: Head -> [Patient1] -> [Patient2] -> [Patient3] -> [Patient4] -> NULL

Deletion by Value:

- **Input:** Delete patient with name “Patient2”.
- **Expected Output:**

  Before: Head -> [Patient1] -> [Patient2] -> [Patient3] -> NULL

  After: Head -> [Patient1] -> [Patient3] -> NULL

Display List:

- **Input:** Display the list.
- **Expected Output:**

  List: Head -> [Patient1] -> [Patient3] -> NULL

  Output: Patient1, Patient3

Submission Files

- Submit your all files to the learning hub before the deadline.

Grading Criteria

- **Correct Solution:** 10 points
- **Error Handling:** 5 points

- **Peer Evaluation:** 5 points

Example Run Command

```
./<executable> input.txt output.txt
```

Notes

- There can be duplicates.
- Values in the input file are not sorted.
- Handle corner cases.
- Ensure no empty lines or invalid data in the input file.
- Print “Error” and exit if any error is detected.
- Do not add random trailing new lines or spaces. One new line character at the end of the output file is acceptable.