

Data Structures and Algorithms

Lab Journal - Lab 5

Name: _____

Enrollment #: _____

Class/Section: _____

Objective

This lab session is aimed at introducing students to singly linked list. In addition, the students will also implement the 'Queue' using a linked list and develop a number of utility functions for singly linked lists.

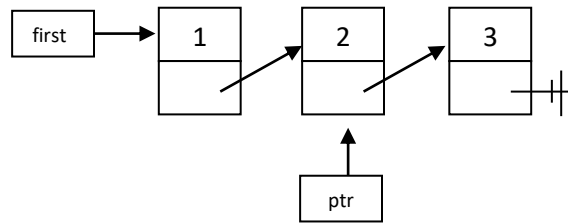
Task 1 :

Give answers to the following.

1.	<p>The following list of names is assigned (in order) to a linear array INFO. Assign value to LINK and START, so that INFO, LINK and START form an alphabetical list.</p> <div style="display: flex; align-items: center; margin-top: 20px;"> <div style="text-align: center; margin-right: 20px;"> <p>START</p> <div style="border: 1px solid black; width: 60px; height: 20px; margin: 0 auto;"></div> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 20%;">INFO</th> <th style="width: 15%;">LINK</th> </tr> </thead> <tbody> <tr><td>1</td><td>Mary</td><td></td></tr> <tr><td>2</td><td>Helen</td><td></td></tr> <tr><td>3</td><td>Barbara</td><td></td></tr> <tr><td>4</td><td>Paula</td><td></td></tr> <tr><td>5</td><td>Diana</td><td></td></tr> <tr><td>6</td><td>Audrey</td><td></td></tr> <tr><td>7</td><td>Karen</td><td></td></tr> <tr><td>8</td><td>Nancy</td><td></td></tr> <tr><td>9</td><td>Ruth</td><td></td></tr> <tr><td>10</td><td>Eileen</td><td></td></tr> </tbody> </table> </div>		INFO	LINK	1	Mary		2	Helen		3	Barbara		4	Paula		5	Diana		6	Audrey		7	Karen		8	Nancy		9	Ruth		10	Eileen	
	INFO	LINK																																
1	Mary																																	
2	Helen																																	
3	Barbara																																	
4	Paula																																	
5	Diana																																	
6	Audrey																																	
7	Karen																																	
8	Nancy																																	
9	Ruth																																	
10	Eileen																																	

2.

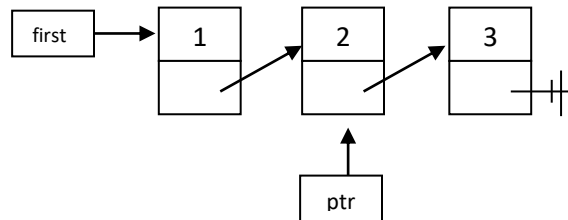
Given the following linked list, state what does each of the following statements refer to.



<code>first->data;</code>	
<code>first->next->next->data;</code>	
<code>ptr->next->data;</code>	
<code>ptr->next->next;</code>	
<code>first->next->next->next;</code>	

3.

Redraw the following list after the given instructions are executed:



```

first -> next = first -> next -> next;
ptr -> next -> next = ptr;
ptr->next = NULL;
  
```

Task 2 :

Implement the following exercises.

Exercise 1

Implement the class Linked List to create a list of integers. You need to provide the implementation of the member functions as described in the following.

```
class List
{
private:
    Node * head;
public:
    List();
    ~List();
    // Checks if the list is empty or not
    bool emptyList();

    // Inserts a new node with value 'newV' after the node
    // containing value 'oldV'. If a node with value 'oldV' does
    // not exist, inserts the new node at the end.
    void insertafter(int oldV, int newV);

    // Deletes the node containing the specified value
    void deleteNode(int value);

    // Inserts a new node at the start of the list
    void insert_begin(int value);

    // Inserts a new node at the end of the list
    void insert_end(int value);

    // Displays the values stored in the list
    void traverse();
};
```

Exercise 2

Linked lists allow efficient implementation of a number of data structures. For instance, Queues can be implemented using a linked list to store the data values. The first node can serve as the 'front' while the last node can be regarded as 'rear' of the queue (Figure 1). The Enqueue() operation is equivalent to adding a node at the end of the list while the Dequeue() operation removes the first element from the list. Implement the 'Queue' class using a linked list for data storage.

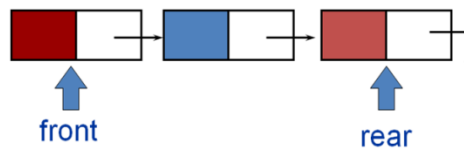


Figure 1 Linked list based implementation of queue

Exercise 3

Write the following C++ functions to realize the indicated functionality on a singly linked list of integers.

- A function which accepts a pointer to the first node and returns the maximum value in the list.
- A function that counts the total number of nodes in the list
- A function to search a given value in the list and return the node number where the queried value is found
- A function to display the elements of the list using recursion.
- A function `swap(Node *p1, Node *p2)` that swaps the data in the nodes p1 and p2.

Implement the given exercises and get them checked by your instructor. If you are unable to complete the tasks in the lab session, deposit this journal alongwith your programs (printed or handwritten) before the start of the next lab session.

S No.	Exercise	Checked By:
1.	Exercise 1	
2.	Exercise 2	
3.	Exercise 3	

+++++