Data Structures and Algorithms

Lab Journal - Lab 7

Name: _____

Enrollment #: _____

Class/Section: _____

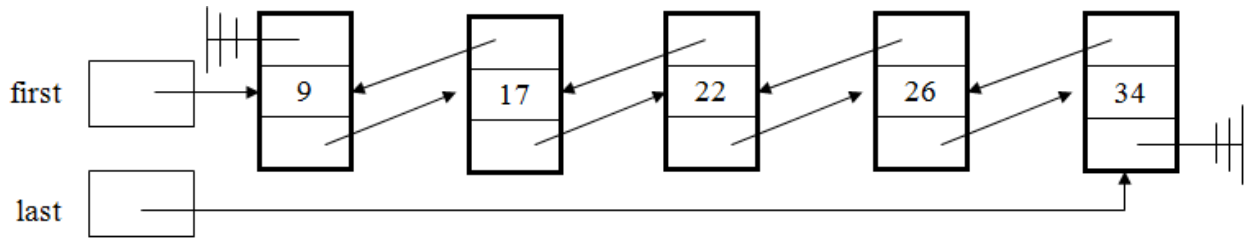
Objective

This lab session is aimed at introducing students to doubly linked list. In addition, the students will also implement a number of utility functions involving doubly linked lists.

Task 1 :

Give answers to the following.

1. Consider the following doubly linked list.



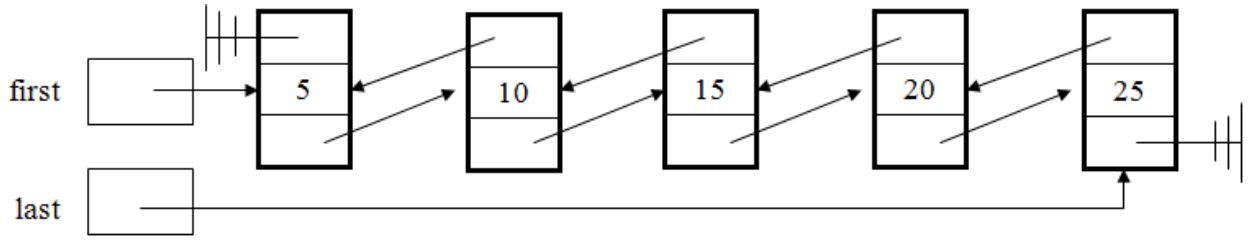
Write C++ statements to:

a. Print the value 26 using the pointer 'last':

b. Print the value 17 using the pointer 'first:

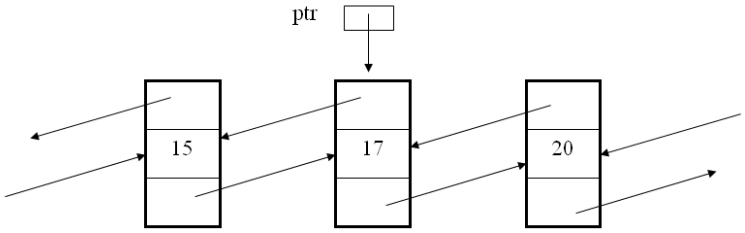
c. Print the address of the node containing value 9 using the pointer 'last':

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



`first->data;`	
`last->next;`	
`first->next->prev;`	
`first->next->next->data;`	
`last->prev->data;`	

3. Redraw the following list after the given instructions are executed:




```
ptr->next->prev = ptr->prev;
ptr->prev->next = ptr->next;
delete ptr;
```

Task 2 :

Implement the following exercises.

Exercise 1

Implement the class Doubly 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

Implement the Stack using a doubly linked list. Provide the standard push, pop and top operations in the class.

Exercise 3

Write a program that stores student ID, name and age in a doubly linked list. Also, write a function to search and display record(s) based on student name.

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	

++++++++++++++++++++++++++