Exp 03-04: Familiarization with linked list and performing different operations on it.

a) Objectives:

- To get clear understanding of linked list
- Linked list construction, traversal
- Finding the length of a linked list
- Performing insert/delete/search operations in linked list
- Reversing a linked list

b) Prerequisites:

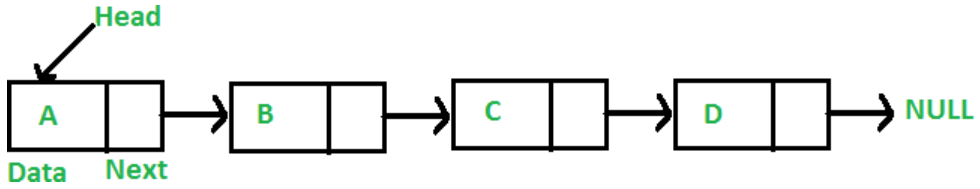
- Pointers
- Structure/Class
- Recursion

c) Theory:

- Data processing frequently involves storing and processing data organized into lists.
- One common way to store such data by means of array
- It is relatively expensive to perform insert/delete operations on array O(n)
- For performing insertion/deletion operations efficiently we can use linked list

Linked list:

- A linked list, or one-way list is a linear collection of data elements, called nodes, where the linear order is given by means of pointers.



- Each node is divided into two fields:
  - i) Data filed: contains information of the element
  - ii) Next pointer field: contains the address of next node in the list
- Representation:

```
struct Node {
  int data;
  Node* next;
};
```

Task1:

Description:	Construct a linked list by taking inputs from user. Stop adding nodes when user enters -1. Then print the entire list. (Note: we will consider this sample input for the later tasks)
Sample Input:	6 3 2 -1
Sample Output:	6 3 2

Task2:

Description:	Find the length of the constructed linked list.
Sample Input:	
Sample Output:	Length = 3

Task3:

Description:	Search for an element in the constructed linked list.	
Sample Input:	3	4
Sample Output:	Found	Not found

Task4:

Description:	Insert an element after P nodes with value X. Then print the entire list again. Insert(P X)
Sample Input:	2 7
Sample Output:	6 3 7 2

Task5:

Description:	Delete first occurrence of X from the linked list. Then print the entire list again. Delete(X)
Sample Input:	3
Sample Output:	6 7 2

Task6:

Description:	Reverse the linked list and then print it.
Sample Input:	
Sample Output:	2 7 6

d) Discussion:

- Implementing stack/queue using linked list
- All codes of the lab will be uploaded to google classroom.

e) Extra Tasks:

Try to implement the following tasks on your own.

1) Find out if a linked list is sorted or not.
2) Insert an element in a sorted linked list. After inserting the element, the list must remain sorted.
3) Find out if a linked list is palindrome or not.
4) Implement stack using linked list.
5) Implement queue using linked list.