# Laboratory 7

1. Questions
   1. Implement a linked list and illustrate the following operations.
      1. Insert a node at the beginning
      2. Insert a node at the end
      3. Print the linked list
   2. Write a program to create a linked list and delete the element entered by a user.
2. Algorithm

**2.1 In a linked list, insert a node at the beginning - Insert a node at the end - Print the linked list**

Step1: start

Step2: to add node at the front.

1. allocate node
2. put in the data
3. Make next of new node as head
4. move the head to point to the new node

step3: to add node at the end

1. allocate node
2. put in the data
3. This new node is going to be the last node, so make next of it as NULL
4. If the Linked List is empty, then make the new node as head
5. Else traverse till the last node
6. Change the next of last node

Step4: call the function accordingly in main body

Step5: stop

**2.2 a program to create a linked list and delete the element entered by a user.**

Step1: start

Step2: make a function for pushing and printing

Step3: delete the element entered by a user

1. Store head node
2. If head node itself holds the key to be deleted, then change head & free old head
3. Search for the key to be deleted, keep track of the previous node as we need to change 'prev->next'
4. If key was not present in linked list, then temp == NULL
5. Unlink the node from linked list
6. Free memory

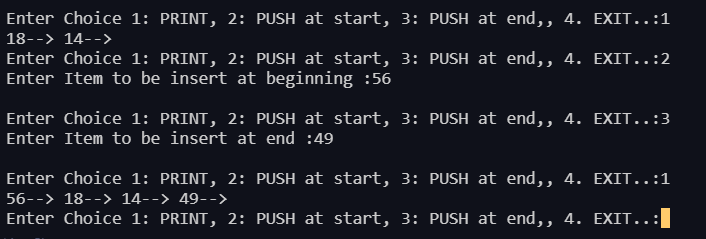
Step4: call the function accordingly in main body

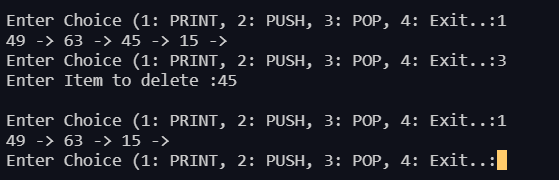
Step5: stop

1. Program

1. Presentation of Results





1. Conclusions

Learning happened:

A **linked list** is a linear data structure where each element is a separate object. Each element (we will call it a node) of a **list** is comprising of two items - the data and a reference to the next node. The last node has a reference to null. The entry point into a **linked list** is called the head of the **list**.

* to add node at the front.

1. allocate node
2. put in the data
3. Make next of new node as head
4. move the head to point to the new node

* to add node at the end

1. allocate node
2. put in the data
3. This new node is going to be the last node, so make next of it as NULL
4. If the Linked List is empty, then make the new node as head
5. Else traverse till the last node
6. Change the next of last node

* delete the element entered by a user

1. Store head node
2. If head node itself holds the key to be deleted, then change head & free old head
3. Search for the key to be deleted, keep track of the previous node as we need to change 'prev->next'
4. If key was not present in linked list, then temp == NULL
5. Unlink the node from linked list
6. Free memory