**LINKED LIST**

**(Algorithm + Pseudocode)**

1. **First create your structure with data and structure type pointer (self-referential structure).**

**struct node{**

**int data;**

**struct node \*link;**

**}**

1. **Create a Head Pointer and make it to point to NULL.**

**struct node \*head = NULL;**

1. **Now create your new node to insert into the linked list using the malloc function. The malloc function allocates memory and returns a void pointer. So we have to type cast the void pointer into a structure type pointer. (to access the structure data use \*)**

**struct node \*new\_node = (struct node\*)malloc(sizeof(struct node));**

1. **Now there are two options for insertion**
2. **When the Head is NULL(Linked List is empty)**

**In this case simply make the head pointer to**

**point to the new node.**

**head = new\_node;**

1. **When the Head is not NULL(Linked List not empty**

**Here again we have two options for insertion**

1. **Insertion in the middle**

* **Traverse till our desired location/position.**

**struct node\* temp = head;**

**while((\*temp).link != NULL){**

**temp = (\*temp).link;**

**}**

* **Now make the new node link to point to the current node link**

**(\*new\_node).link = (\*temp).link;**

* **Then make the current node**

**(\*temp).link = new\_node;**

1. **Insertion at the end**

* **Traverse till the end of the Linked list.**
* **Make the current node link to point the new node.**
* **Make the new node link to NULL.**

1. **Deletion has 3 options. We have to traverse to the desired position first and then do the operation:**
2. **At the beginning:**

**Just change the head pointer to the next and its done.**

**head = (\*head).link;**

1. **At the end:**

**Go to the second last position. Then make it to point to NULL;**

1. **In the middle:**

**Go to the (n–1)th node. Here n is the node to be deleted. Now make the (n-1)th node to point to the (n+1)th node.**

**struct node \*temp = head;**

**int cnt = 0;**

**while(cnt < n – 1){**

**temp = (\*temp).link;**

**}**

**(\*temp).link = (\*(\*temp).link).link;**