**Doubly Linked List**

In single linked list we don’t have the facility to traverse or display the list in a backward direction i.e. we can’t travel from last node to first node. Double linked is a type of linked list where each node have link of both previous and next node.

Here each node have three components one data part and two link part.

Struct node{

int data;

struct node \*prev;

struct node \*next;

};



**Creation of node of Doubly Linked List:**

struct node\* start = NULL;

start = (struct node\*)malloc(sizeof(struct node));

start->prev = NULL;

start->next = NULL;

This is a node which have 3 part 2 link part which contains NULL and one data part where we can store value.

**Operation on Doubly Linked list:**

1. Add node in the list
2. Add at beginning
3. Add at any location
4. Add at the end of the list
5. Length of the list
6. Display the list
7. Delete the element in the list
8. Delete at beginning
9. Delete at any location
10. Delete at end
11. Search an element
12. Reverse the list

**Add an element in the list:**

To add an element we have to create a node first. Then we have to add it at beginning, end or at any particular position

1. Add at beginning:

If the list is empty then there is no doubt of adding at any position. Since the node should be pointed by start so the code is:

struct node \*temp;

temp = start;

temp = (struct node\*)malloc(sizeof(struct node));

printf(“Enter the data“);

scanf(“%d”,&temp->data);

temp->prev = NULL;

temp->next = NULL;

if(start == null)