

Name:- Anurag Halder Subject:- Data Structures
Lab

Class:- 5P-13

Roll no:- 2215055

Enrollment No:- MITU21BT1T0010

Lab Write-Up (Ques- Ans)
Practical No:- 7

Q.1) What is SLL?

→ Singly linked list is a set of nodes where each node has two fields. "data" and "link". The data field stores actual piece of information and 'link' field is used to point to next node. Basically the link field stores the address of the next node.

First node of singly linked list is called head. A singly linked list is a unidirectional linked list where you can traverse from head to last node contains NULL as the limit.

Q.2) What is the structure of node of SLL?

→ Each element in SLL is stored in node structure or node in singly linked list is two sub-elements called data and link. The data part stores the information and the link part stores the address of next node.

The last node of SLL is containing the NULL as the link; that

determines the end of linked list.

Syntax:-

struct Node

{ int data;

struct Node link

}

Q.3) How can we delete node from SLL?

→ We can delete node from SLL in three locations:-

[A] Beginning

[B] Middle

[C] End.

1) Beginning

// deleting the node from beginning

temp = head;

head = head → next;

free(temp);

2) Middle -

// deleting the node from middle

temp = head;

prev = head;

for (int i = 0; i < pos; i++)

{ if (i == 0 && pos == 1)

head = head → next;


```
else {  
    if (i == pos-1 && temp)  
        prev → next = temp → next;  
        free(temp);  
}  
else {  
    prev = temp;  
    if (prev == NULL)  
        break;  
    temp = temp → next;  
}
```

3) End :-

// deleting node from end.

```
struct node * end = head;  
while (end → next)  
{  
    prev = end;  
    end = end → next;  
}  
prev → next = NULL;  
free(end);
```

Q.4) How can we insert node in SLL?
→ We can insert node from SLL in
3 location.

A) Beginning

// insert the element at beginning

struct node * new;

struct node * next node;

new = head;

new → next = head → next;

head = next node

B) Middle :-

// inserting at the middle

struct node * new node, * temp

if (head == NULL)

head = New Node;

else {

temp = head;

while (temp → next != NULL)

temp = temp → next;

temp → next = new node;

}

c) End -

// inserting at the end

struct node * new node; * end

end = end → next

end → next = NULL;