

```
/******SUBHAS NATH*****/
/******
/******PROGRAM OF LINEAR LINKED LIST DELETION*****/
```

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define NULL 0
struct link
{
    int info;
    struct link *next;
};
typedef struct link node;
void main()
{
    node *head;
    clrscr();
    void create(node *);
    void show(node *);
    node *del_fst(node *);
    node *del_lst(node *);
    head=(node *)malloc(sizeof(node)); //CREATE STORAGE FOR FIRST NODE
    printf("\n\tCreate the linked list:\n");
    create(head);
    printf("\n\tDisplay the linked list:\n");
    show(head);
    printf("\n\n\tDeletion from 1st position.\n");
    head=del_fst(head);
    show(head);
    printf("\n\n\tDeletion from last position.\n");
    head=del_lst(head);
    show(head);
    getch();
}
```

```
/******CREATE LINKED LIST*****/
```

```
void create(node *temp)
{
    char ans;
    printf("\nEnter the info:- ");
    scanf("%d",&temp->info);
    temp->next=NULL;
```

```

printf("\nWant Another? ");
fflush(stdin); //CLEAR THE BUFFER
ans=getchar();
if(ans!='y')
    return;
else
{
    temp->next=(node *)malloc(sizeof(node)); //CREATE STORAGE FOR NEXT NODE
    create(temp->next); //RECURSIVE CALL TO CREATE NEXT NODE
}
}

```

/******DISPLAY LINKED LIST******/

```

void show(node *temp)
{
    if(temp->next==NULL)
        printf("%d",temp->info); //DISPLAY LAST NODE
    else
    {
        printf("%d->",temp->info); //DISPLAY CURRENT INFO
        show(temp->next); //RECURSIVE CALL TO DISPLAY NEXT INFO
    }
}

```

/******DELETION FROM FIRST POSITION******/

```

node *del_fst(node * temp)
{
    node * list;
    if(temp==NULL)
        printf("\n\tUNDERFLOW.\n");
    if(temp!=NULL)
    {
        list=temp->next;
        free(temp);
        temp=list;
    }
    return(temp);
}

```

```
/******DELETION FROM LAST POSITION******/
```

```
node *del_lst(node *temp)
{
    node *last,*first;
    first=temp;
    while(temp->next->next!=NULL)
    {
        temp=temp->next;
    }
    if(temp->next->next==NULL)
    {
        last=temp->next;
        free(last);
        temp->next=NULL;
    }
    return(first);
}
```