

```

*****SUBHAS NATH*****
*****PROGRAM OF DOUBLE LINKED LIST*****

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define NULL 0
struct link
{
    struct link *prev;
    int info;
    struct link *next;
};
typedef struct link node;
void main()
{
    node *head;
    clrscr();
    void create(node *);
    void show(node *);
    node *insert_fst(node *);
    node *insert_lst(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\tInsertion at 1st position.\n");
    head=insert_fst(head);
    show(head);
    printf("\n\n\tInsertion at last position.\n");
    head=insert_lst(head);
    show(head);
    getch();
    printf("\n\n\tDeletion at 1st position.\n");
    head=del_fst(head);
    show(head);
    getch();
    printf("\n\n\tDeletion at last position.\n");
    head=del_lst(head);
    show(head);
    getch();
}

```

```

*****CREATE LINKED LIST*****

void create(node *temp)
{
    char ans;
    node *p;
    temp->prev=NULL;
    printf("\nEnter the info:- ");
    scanf("%d",&temp->info);
    while(1)
    {
        printf("\nWant Another? ");
        fflush(stdin); //CLEAR THE BUFFER
        ans=getchar();
        if(ans!='y')
            break;
        else
        {
            p=temp;
            temp->next=(node *)malloc(sizeof(node));//CREATE STORAGE FOR NEXT
NODE
            temp=temp->next;
            temp->prev=p;
            printf("\nEnter the info:- ");
            scanf("%d",&temp->info);
        }
    }
    temp->next=NULL;
}

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


void show(node *temp)
{
    while(temp->next!=NULL)
    {
        printf("%d<->",temp->info); //DISPLAY UPTO PREVIOUS OF LAST NODE
        temp=temp->next;
    }
    if(temp->next==NULL)
    {
        printf("%d",temp->info);//DISPLAY LAST NODE
    }
    printf("\n");
    while(temp->prev!=NULL)

```

```

{
printf("%d<->",temp->info); //DISPLAY FROM LAST TO SECOND NODE
temp=temp->prev;
}
if(temp->prev==NULL)
{
printf("%d",temp->info); //DISPLAY FIRST NODE
}
}

```

*****INSERTION AT FIRST POSITION*****

```

node *insert_fst(node * temp)
{
node * list;
list=(node *)malloc(sizeof(node));
list->prev=NULL;
printf("\nEnter the element:- ");
scanf("%d",&list->info);
list->next=temp;
temp->prev=list;
temp=list;
return(temp);
}

```

*****INSERTION AT LAST POSITION*****

```

node *insert_lst(node *temp)
{
node *last,*first;
first=temp;
while(temp->next!=NULL)
{
temp=temp->next;
}
if(temp->next==NULL)
{
last=(node *)malloc(sizeof(node));
printf("\nEnter the element:- ");
scanf("%d",&last->info);
temp->next=last;
last->prev=temp;
last->next=NULL;
}
return(first);
}

```

```
*****DELETION FROM FIRST POSITION*****
```

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