**AIM**: To implement a C Program for Singly Linked List.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

***/\*----Structure of one Node----\*/***

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

***/\*--Insertion from beginning---\*/***

void insert\_beg(int ele)

{

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

temp->data=ele;

temp->next=head;

head=temp;

}

***/\*--Insertion from the end---\*/***

void insert\_end(int ele)

{

struct node \*temp1=head;

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

temp->data = ele;

temp->next = NULL;

if(head==NULL)

insert\_beg(ele);

else

{

while(temp1->next!=NULL)

{

temp1=temp1->next;

}

temp1->next=temp;

}

}

***/\*---Deleting from beginning---\*/***

void delete\_beg()

{

struct node \*temp=head;

if(head==NULL)

printf("Cannot delete element. Underflow\n");

else

{

head = temp->next;

free(temp);

}

}

***/\*--Deleting from the end---\*/***

void delete\_end()

{

struct node \*temp,\*temp1;

temp=temp1=head;

if(head==NULL)

printf("Underflow Condition.\n");

else if(head->next==NULL)

{

temp=head;

free(temp);

head=NULL;

}

else

{

while(temp->next!=NULL)

{

temp1=temp;

temp = temp->next;

}

temp1->next=NULL;

free(temp);

}

}

***/\*--- count( ): Counts the number of elements in the linked list.---\*/***

int count(void)

{

struct node \*temp=head;

int count=1;

while(temp->next!=NULL)

{

temp=temp->next;

count++;

}

return (count);

}

***/\*---Intermediate Insertion---\*/***

void insert\_any(int ele, int pstn)

{

int k,i;

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

temp->data=ele;

temp->next=NULL;

k=count();

if(pstn<=k+1)

{

if(pstn==1)

insert\_beg(ele);

else if(pstn==k+1)

insert\_end(ele);

else

{

struct node \*temp1=head;

for(i=0;i<pstn-2;i++)

{

temp1=temp1->next;

}

temp->next=temp1->next;

temp1->next=temp;

}

}

else

{

printf("\nThe last position of the element is %d. Hence, cannot insert %d at %d position.", k,ele,pstn);

}

}

***/\*---Intermediate Deletion---\*/***

void delete\_any(int pstn)

{

int k,i;

struct node \*temp=head;

struct node \*temp1=head;

k=count();

if(pstn<=k)

{

if(pstn==1)

delete\_beg();

else if(pstn==k)

delete\_end();

else

{

for(i=0;i<pstn-2;i++)

temp=temp->next;

temp1=temp->next;

temp->next=temp1->next;

printf("\nDeleted [ %d | %d ] node.\n", temp1->data, temp1->next);

temp1->next=NULL;

free(temp1);

}

}

else

{

printf("\nTotal elements: %d. Hence there is not element at %d position.\n", k, pstn);

}

}

***/\*-------Display Function-------\*/***

void display()

{

struct node \*temp=head;

printf("\nHead(%d)", head);

while(temp!=NULL)

{

printf("-->[%d(%d) | %d]", temp->data,temp,temp->next);

temp=temp->next;

}

printf("\n");

}

***/\*------MAIN FUNCTION-----\*/***

int ele,pstn;

void main()

{

int ch;

clrscr();

do

{

printf("\t=====MENU=======\n");

printf("\t1] Insert from Beginning\n");

printf("\t2] Delete from Beginning\n");

printf("\t3] Insert from End\n");

printf("\t4] Delete from End\n");

printf("\t5] Insert at any position\n");

printf("\t6] Delete from any position\n");

printf("\t7] Display.\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("\nEnter the element you want to insert: ");

scanf("%d", &ele);

insert\_beg(ele);

break;

case 2:

delete\_beg();

break;

case 3:

printf("\nEnter the element you want to insert: ");

scanf("%d", &ele);

insert\_end(ele);

break;

case 4:

delete\_end();

break;

case 5:

printf("\nEnter the element to insert: ");

scanf("%d", &ele);

printf("\nWhere do you want to insert %d? ", ele);

scanf("%d", &pstn);

insert\_any(ele,pstn);

break;

case 6:

printf("\nEnter the position of element you want to delete.");

scanf("%d", &pstn);

delete\_any(pstn);

break;

case 7:

display();

break;

}

}

while(ch);

getch();

}