**Practical no. 5**

AIM- To write a menu-driven program to implement Singly Linked List Operations

PROGRAM-

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*t,\*x,\*y;

struct node \*s=NULL;

struct node\* insb()

{

int dt;

printf("Enter element data:");

scanf("%d",&dt);

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

t->data=dt;

t->next=s;

s=t;

return s;

}

void inse()

{

int dt;

printf("Enter element data:");

scanf("%d",&dt);

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

t->data=dt;

x=s;

while(x->next!=NULL)

x=x->next;

x->next=t;

t->next=NULL;

}

void insp()

{

int dt,p;

printf("Enter element data:");

scanf("%d",&dt);

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

t->data=dt;

printf("Enter position :");

scanf("%d",&p);

p--;

y=s;

while(p>0)

{

x=y;

y=y->next;

p--;

}

x->next=t;

t->next=y;

}

void delb()

{

if(s==NULL)

printf("No node!");

else

{

t=s;

s=s->next;

free(t);

}

}

void dele()

{

if(s==NULL)

printf("No node!");

else

{

x=s;

while(x->next!=NULL)

{

y=x;

x=x->next;

}

y->next=NULL;

free(x);

}

}

void delp()

{ int p;

if(s==NULL)

printf("No node!");

else

{

printf("Enter position :");

scanf("%d",&p);

p--;

y=s;

while(p>0)

{

x=y;

y=y->next;

p--;

}

x->next=y->next;

free(y);

}

}

void disp()

{

x=s;

while(x->next!=NULL)

{

printf("\n%d",x->data);

x=x->next;

}

printf("\n%d",x->data);

}

void search()

{

int c=0,e;

x=s;

printf("Enter element to be searched:");

scanf("%d",&e);

for(x=s;x->next!=NULL;x=x->next)

{

if(x->data==e)

{

printf("Element found at position:%d",c+1);

break;

}

c++;

}

if(x->next==NULL)

{

if(x->data==e)

printf("Element found at position:%d",c+1);

else

printf("Element not found!");

}

}

void main()

{

int o,ch;

do

{

printf("MENU\n1.Add at beginning\n2.Add at end\n3.Add at position\n4.Delete from beginning\n5.Delete at end\n6.Delete at position\n7.Display\n8.Search");

printf("\n Enter choice:");

scanf("%d",&ch);

switch(ch)

{

case 1: s=insb();

disp();

break;

case 2: inse();

disp();

break;

case 3: insp();

disp();

break;

case 4: delb();

disp();

break;

case 5: dele();

disp();

break;

case 6: delp();

disp();

break;

case 7: disp();

break;

case 8: search();

break;

}

printf("Enter 1 to continue:");

scanf("%d",&o);

}while(o==1);

}

OUTPUT-

1.Add at beginning

2.Add at end

3.Add at position

4.Delete from beginning

5.Delete at end

6.Delete at position

7.Display

8.Search

Enter choice:1

Enter element data:12

12

Enter 1 to continue:1

1.Add at beginning

2.Add at end

3.Add at position

4.Delete from beginning

5.Delete at end

6.Delete at position

7.Display

8.Search

Enter choice:2

Enter element data:13

12

13

Enter 1 to continue:1

1.Add at beginning

2.Add at end

3.Add at position

4.Delete from beginning

5.Delete at end

6.Delete at position

7.Display

8.Search

Enter choice:3

Enter element data:14

Enter position :3

12

13

14

Enter 1 to continue:1

1.Add at beginning

2.Add at end

3.Add at position

4.Delete from beginning

5.Delete at end

6.Delete at position

7.Display

8.Search

Enter choice:4

13

14

Enter 1 to continue:1

1.Add at beginning

2.Add at end

3.Add at position

4.Delete from beginning

5.Delete at end

6.Delete at position

7.Display

8.Search

Enter choice:5

13

Enter 1 to continue:0