**Exercise 7: Write a menu driven program to implement following operations on the singly linked list. (a) Insert a node at the front of the linked list.**

**(b) Insert a node at the end of the linked list.**

**(c) Insert a node such that linked list is in ascending order (according to info. Field).**

**(d) Delete a first node of the linked list.**

**(e) Delete a node before specified position.**

**(f) Delete a node after specified position**

#include<stdio.h>

#include<conio.h>

struct node

{

int data;

struct node \*next;

}\*start=NULL,\*q,\*t;

int main()

{

int ch;

void insert\_beg();

void insert\_end();

int insert\_pos();

void display();

void delete\_beg();

void delete\_end();

int delete\_pos();

while(1)

{

printf("\n\n---- Singly Linked List(SLL) Menu ----");

printf("\n1.Insert\n2.Display\n3.Delete\n4.Exit\n\n");

printf("Enter your choice(1-4):");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("\n---- Insert Menu ----");

printf("\n1.Insert at beginning\n2.Insert at end\n3.Insert at specified position\n4.Exit");

printf("\n\nEnter your choice(1-4):");

scanf("%d",&ch);

switch(ch)

{

case 1: insert\_beg();

break;

case 2: insert\_end();

break;

case 3: insert\_pos();

break;

case 4: exit(0);

default: printf("Wrong Choice!!");

}

break;

case 2: display();

break;

case 3: printf("\n---- Delete Menu ----");

printf("\n1.Delete from beginning\n2.Delete from end\n3.Delete from specified position\n4.Exit");

printf("\n\nEnter your choice(1-4):");

scanf("%d",&ch);

switch(ch)

{

case 1: delete\_beg();

break;

case 2: delete\_end();

break;

case 3: delete\_pos();

break;

case 4: exit(0);

default: printf("Wrong Choice!!");

}

break;

case 4: exit(0);

default: printf("Wrong Choice!!");

}

}

return 0;

}

void insert\_beg()

{

int num;

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

printf("Enter data:");

scanf("%d",&num);

t->data=num;

if(start==NULL) //If list is empty

{

t->next=NULL;

start=t;

}

else

{

t->next=start;

start=t;

}

}

void insert\_end()

{

int num;

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

printf("Enter data:");

scanf("%d",&num);

t->data=num;

t->next=NULL;

if(start==NULL) //If list is empty

{

start=t;

}

else

{

q=start;

while(q->next!=NULL)

q=q->next;

q->next=t;

}

}

int insert\_pos()

{

int pos,i,num;

if(start==NULL)

{

printf("List is empty!!");

return 0;

}

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

printf("Enter data:");

scanf("%d",&num);

printf("Enter position to insert:");

scanf("%d",&pos);

t->data=num;

q=start;

for(i=1;i<pos-1;i++)

{

if(q->next==NULL)

{

printf("There are less elements!!");

return 0;

}

q=q->next;

}

t->next=q->next;

q->next=t;

return 0;

}

void display()

{

if(start==NULL)

{

printf("List is empty!!");

}

else

{

q=start;

printf("The linked list is:\n");

while(q!=NULL)

{

printf("%d->",q->data);

q=q->next;

}

}

}

void delete\_beg()

{

if(start==NULL)

{

printf("The list is empty!!");

}

else

{

q=start;

start=start->next;

printf("Deleted element is %d",q->data);

free(q);

}

}

void delete\_end()

{

if(start==NULL)

{

printf("The list is empty!!");

}

else

{

q=start;

while(q->next->next!=NULL)

q=q->next;

t=q->next;

q->next=NULL;

printf("Deleted element is %d",t->data);

free(t);

}

}

int delete\_pos()

{

int pos,i;

if(start==NULL)

{

printf("List is empty!!");

return 0;

}

printf("Enter position to delete:");

scanf("%d",&pos);

q=start;

for(i=1;i<pos-1;i++)

{

if(q->next==NULL)

{

printf("There are less elements!!");

return 0;

}

q=q->next;

}

t=q->next;

q->next=t->next;

printf("Deleted element is %d",t->data);

free(t);

return 0;

}