Assignment No. 3

**TITLE : Singly Linked List Operations**

Name : METHIKA M Date : 27/08/2025

UNo : UIT2024936

Class: IT- I DIV

Batch : I2  
  
  
**SOURCE CODE:**

/\*

\* SLL.c

\*

\* Created on: Aug 14, 2025

\* Author: administrator

\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

void accept(struct node \*\*fhead,struct node \*\*flast)

{

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*newnode=NULL;

char ch;

int k=0;

head=\*fhead;

last=\*flast;

do

{

newnode=(struct node \*)calloc(1,sizeof(struct node));

printf("\nEnter data : ");

scanf("%d",&k);

newnode -> data = k;

newnode -> next = NULL;

if(head==NULL)

{

head = newnode;

last = newnode;

}

else

{

last -> next = newnode;

last = newnode;

}

printf("\nDo you want to add another node? (y/n)");

printf("\nEnter your Choice : ");

scanf(" %c",&ch);

}while(ch=='y' || ch=='Y');

\*fhead=head;

\*flast=last;

}

void display(struct node \*\*fhead)

{

struct node \*head=NULL;

struct node \*i=NULL;

head=\*fhead;

for(i=head;i!=NULL;i=i->next)

{

printf("\t%d",i->data);

}

printf("\n");

}

int menu()

{

int ch=0;

printf("\nSINGLY LINKED LIST OPERATIONS");

printf("\nEnter operation to perform on SSL :");

printf("\n1. Insert \n2. Delete \n0. Exit");

printf("\nEnter your choice :");

scanf("%d",&ch);

return ch;

}

void insert(struct node \*\*fhead,struct node \*\*flast)

{

int how=0;

int where=0;

int which=0;

int count=1;

int k=0;

struct node \*i=NULL;

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*newnode=NULL;

head=\*fhead;

last=\*flast;

newnode=(struct node \*)calloc(1,sizeof(struct node));

printf("\nEnter data to insert :");

scanf("%d",&k);

newnode -> data = k;

newnode -> next = NULL;

printf("\nHow do you want to insert?");

printf("\n1.Using Data");

printf("\n2.Using Position");

printf("\nEnter your choice :");

scanf("%d",&how);

switch(how)

{

case 1:

printf("\nWhere do you want to insert? :");

printf("\n1.Before Data");

printf("\n2.After Data");

printf("\nEnter your choice :");

scanf("%d",&where);

if(where==1)

{

printf("\nBefore which data do you want to insert? :");

printf("\nEnter your choice :");

scanf("%d",&which);

if(head->data==which)

{

newnode->next=head;

head=newnode;

}

else

{

for(i=head;i->next->data!=which;i=i->next)

{

continue;

}

newnode->next=i->next;

i->next=newnode;

}

}

if(where==2)

{

printf("\nAfter which data do you want to insert? :");

printf("\nEnter your choice :");

scanf("%d",&which);

for(i=head;i->data!=which;i=i->next)

{

continue;

}

newnode->next=i->next;

i->next=newnode;

}

break;

case 2:

printf("\nWhich position do you want to insert? :");

printf("\nEnter your choice :");

scanf("%d",&which);

if(which==1)

{

newnode->next=head;

head=newnode;

}

else

{

for(i=head;count!=(which-1);i=i->next)

{

count++;

}

newnode->next=i->next;

i->next=newnode;

if(newnode->next==NULL)

{

last=newnode;

}

}

break;

}

\*fhead=head;

\*flast=last;

}

void delete(struct node \*\*fhead,struct node \*\*flast)

{

int how=0;

int which=0;

int count=1;

struct node \*i=NULL;

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*track=NULL;

head=\*fhead;

last=\*flast;

printf("\nHow do you want to delete data?");

printf("\n1.Using Data");

printf("\n2.Using Position");

printf("\nEnter your choice :");

scanf("%d",&how);

switch(how)

{

case 1:

printf("\nWhich data do you want to delete? :");

printf("\nEnter your choice :");

scanf("%d",&which);

if(head->data==which)

{

track=head;

head=head->next;

free(track);

}

else

{

for(i=head;i->next->data!=which;i=i->next)

{

continue;

}

track=i->next;

i->next=i->next->next;

free(track);

if(i->next==NULL)

{

last=i;

}

}

break;

case 2:

printf("\nWhich position do you want to delete? :");

printf("\nEnter your choice :");

scanf("%d",&which);

if(which==1)

{

track=head;

head=head->next;

free(track);

}

else

{

for(i=head;count!=(which-1);i=i->next)

{

count++;

}

track=i->next;

i->next=i->next->next;

free(track);

if(i->next==NULL)

{

last=i;

}

}

break;

}

\*fhead=head;

\*flast=last;

}

int main(void)

{

struct node \*head=NULL;

struct node \*last=NULL;

int ch=0;

int count=1;

accept(&head,&last);

printf("\nEntered Linked List :");

display(&head);

do

{

ch=menu();

switch(ch)

{

case 1:

insert(&head,&last);

printf("\nUpdated Linked List :");

display(&head);

break;

case 2:

delete(&head,&last);

printf("\nUpdated Linked List :");

display(&head);

break;

case 0:

count=0;

printf("\nExiting...\nThank You");

break;

}

}while(count==1);

return 0;

}  
  
**OUTPUT**







