Assignment No. 4

**TITLE : Doubly Linked List Operations**

Name : METHIKA M Date : 29/08/2025

UNo : UIT2024936

Class: IT- I DIV

Batch : I2  
  
  
**SOURCE CODE**#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

};

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

{

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*newnode=NULL;

head=\*fhead;

last=\*flast;

int k=0;

char ch;

do

{

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

printf("\nEnter data: ");

scanf("%d", &k);

newnode->data=k;

newnode->next=NULL;

newnode->prev=NULL;

if (head==NULL)

{

head=newnode;

last=newnode;

}

else

{

newnode->prev=last;

last->next=newnode;

last=newnode;

}

printf("\nDo you want to add another elements?(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);

}

}

int menu (void)

{

int ch=0;

printf("\nDOUBLY LINKED LIST OPERATIONS");

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

printf("\n1. Insert \n2. Delete \n3. Reverse \n4. Revert \n0. Exit\n");

printf("\nEnter your choice: ");

scanf("%d",&ch);

return ch;

}

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

{

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*newnode=NULL;

struct node \*i=NULL;

int count =1;

int k=0;

head=\*fhead;

last=\*flast;

int how=0;

int where=0;

int which=0;

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

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

scanf("%d",&k);

newnode->prev=NULL;

newnode -> data = k;

newnode -> next = NULL;

printf("\nHow do you wish to insert the data? ");

printf("\n1. Using data \n2. Using position\n");

printf("\nEnter your choice: ");

scanf("%d", &how);

switch (how)

{

case 1:

printf("Where do you wish to enter the data? ");

printf("\n1. Before Data \n2. After data\n");

printf("Enter your choice: ");

scanf ("%d", &where);

if (where==1)

{

printf("Before which data do you want to insert? ");

printf("Enter your choice: ");

scanf ("%d", &which);

if (head->data==which)

{

newnode->prev=NULL;

newnode->next=head;

head->prev=newnode;

head=newnode;

}

else

{

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

{

continue;

}

newnode->prev=i;

newnode->next=i->next;

i->next->prev = newnode;

i->next=newnode;

}

}

if (where==2)

{

printf("After which data do you want to insert? ");

printf("Enter your choice: ");

scanf ("%d", &which);

if (last->data==which)

{

newnode->prev=last;

newnode->next=NULL;

last->next=newnode;

last=newnode;

}

else

{

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

{

continue;

}

newnode->prev=i;

newnode->next=i->next;

if (i->next != NULL)

{

i->next->prev = newnode;

}

else

{

last = newnode;

}

i->next=newnode;

}

}

break;

case 2:

printf("At what position do you wish to insert? ");

printf("Enter your choice: ");

scanf("%d",&which);

if (which==1)

{

newnode->prev=NULL;

newnode->next=head;

head->prev=newnode;

head=newnode;

}

else

{

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

{

count++;

}

newnode->prev=i;

newnode->next=i->next;

i->next->prev = newnode;

i->next=newnode;

}

break;

}

\*fhead=head;

\*flast=last;

}

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

{

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*temp=NULL;

struct node \*i=NULL;

int count =1;

int k=0;

head=\*fhead;

last=\*flast;

int how=0;

int which=0;

printf("How would you like to delete your data? ");

printf("\n1. Using data \n2. Using position");

printf("\nEnter your choice : ");

scanf("%d", &how);

switch (how)

{

case 1:

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

printf("\nEnter your choice: ");

scanf("%d", &which);

if (head->data==which)

{

temp=head;

head=head->next;

head->prev=NULL;

free (temp);

}

else

{

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

{

continue;

}

temp=i->next;

i->next=temp->next;

if (temp->next != NULL)

{

temp->next->prev = i;

}

else

{

last = i;

}

free(temp);

}

break;

case 2:

printf("\nWhich position data would you like to delete? ");

printf("\nEnter your choice: ");

scanf("%d", &which);

if (which==1)

{

temp=head;

head=head->next;

head->prev=NULL;

free (temp);

}

else

{

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

{

count++;

}

temp=i->next;

i->next=temp->next;

if (temp->next != NULL)

{

temp->next->prev = i;

}

else

{

last = i;

}

free(temp);

}

break;

}

\*fhead=head;

\*flast=last;

}

void reverse (struct node \*\*flast)

{

struct node \*last=NULL;

struct node \*i=NULL;

last=\*flast;

for (i=last;i!=NULL;i=i->prev)

{

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

}

}

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

{

struct node \*head=NULL;

struct node \*last=NULL;

struct node \*h1=NULL;

struct node \*l1=NULL;

struct node \*dnode=NULL;

head=\*fhead;

last=\*flast;

while(head!=NULL && last!=NULL)

{

dnode = head;

head= head->next;

dnode->next=NULL;

if (h1==NULL)

{

h1=dnode;

l1=dnode;

}

else

{

dnode->next=h1;

h1=dnode;

}

}

head=h1;

last=l1;

\*fhead=head;

\*flast=last;

}

int main(void)

{

struct node \*head=NULL;

struct node \*last=NULL;

accept (&head, &last);

printf("\nThe entered linked list is: ");

display(&head);

int ch=0;

int count=1;

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 3:

printf("\nThe reversed Linked list is: ");

reverse(&last);

break;

case 4:

printf("\nThe reverted Linked list is: ");

revert(&head, &last);

display (&head);

break;

case 0:

printf("\nExiting...");

printf("\nThank you! ");

count=0;

}

}while(count==1);

return 0;

}

**OUTPUT**









