LAB 7

Insertion and deletion in a doubly linked list

#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*prev;

struct node \*next;

};

struct node \*head = NULL;

void insert();

void delete();

void display();

void main()

{

int ch;

printf("Press - 1.Insert\n 2.Delete\n 3.Display\n");

while(ch!=4)

{

printf("Enter choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

}

}

}

void insert()

{

int pos,i=1;

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

struct node \*ptr=head;

printf("Enter data:");

scanf("%d",&new->data);

if(head == NULL)

{

new -> prev = NULL;

new -> next = NULL;

head = new;

printf("Node inserted\n");

return;

}

printf("Enter the position:");

scanf("%d",&pos);

if(pos==1)

{

new -> prev =NULL;

new->next = head;

head -> prev =new;

head=new;

printf("Node inserted\n");

}

else{

for(i=0;i<pos-1;i++)

{

ptr = ptr->next;

}

new->prev = ptr->prev;

new->next = ptr;

ptr->prev->next=new;

ptr->prev = new;

printf("Node inserted\n");

}

}

void delete()

{

int val;

printf("Enter the value:");

scanf("%d",&val);

struct node \*ptr=head;

if(head->data == val)

{

head = ptr->next;

free(ptr);

printf("Node deleted\n");

return;

}

while(ptr->data != val)

{

ptr = ptr->next;

if(ptr->next == NULL)

{

ptr->prev->next = NULL;

free(ptr);

printf("Node deleted\n");

return;

}

}

ptr->prev->next = ptr->next;

ptr->next->prev = ptr->prev;

free(ptr);

printf("Node deleted\n");

}

void display()

{

struct node \*p=head;

while(p != NULL)

{

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

p=p->next;

}

printf("NULL\n");

}

OUTPUT:

