#include <stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*prev;

struct node \*next;

};

struct node \*head=NULL;

void create();

void display();

void insert\_beg();

void insert\_end();

void delete\_beginning();

void delete\_end();

void insert\_specific();

void delete\_specific();

void find();

void find\_prev();

void find\_next();

int main()

{

int x=1;

while(x)

{

int choice;

printf("\n\*\*\*\*\*MENU\*\*\*\*\*\n1.create\n2.insert at beginning\n3.insert at end\n4.delete from beginning\n5.delete from end\n6.delete specific\n7.insert\_specific\n8.find element\n9.find previous\n10.find next\n11.display\n12.exit");

printf("\nenter choice");

scanf("%d",&choice);

switch(choice)

{

case 1:

create();

break;

case 2:

insert\_beg();

break;

case 3:

insert\_end();

break;

case 4:

delete\_beginning();

break;

case 5:

delete\_end();

break;

case 6:

delete\_specific();

break;

case 7:

insert\_specific();

break;

case 8:

find();

break;

case 9:

find\_prev();

break;

case 10:

find\_next();

break;

case 11:

display();

break;

case 12:

x=0;

printf("\nexitingg....\n");

break;

}

}

}

void create()

{

struct node \*temp;

temp=head;

int x=1,item;

while(x)

{

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

printf("enter element");

scanf("%d",&item);

newnode->data=item;

newnode->next=NULL;

if(head==NULL)

{

head=temp=newnode;

head->prev=NULL;

}

else

{

temp->next=newnode;

newnode->prev=temp;

}

temp=newnode;

printf("add more?(1/0)");

scanf("%d",&x);

}

printf("\nlist created!!!");

}

void display()

{

struct node \*temp;

temp=head;

if(head==NULL)

{

printf("List is empty!!\n");

}

else

{

while(temp->next!=NULL)

{

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

temp=temp->next;

}

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

}

}

void insert\_beg()

{

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

printf("enter element to insert");

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

newnode->next=head;

newnode->prev=NULL;

head=newnode;

printf("one node inserted");

}

void insert\_end()

{

struct node \*temp;

temp=head;

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

printf("enter element to insert");

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

newnode->next=NULL;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=newnode;

newnode->prev=temp;

printf("\nOne node inserted!!\n");

}

void delete\_beginning()

{

head->next->prev=NULL;

head=head->next;

printf("\n one node deleted\n");

}

void delete\_end()

{

struct node \*temp;

temp=head;

while(temp->next!=0)

{

temp=temp->next;

}

temp->prev->next=NULL;

printf("\n one node deleted!\n");

}

void insert\_specific()

{

int pos;

struct node \*temp;

temp=head;

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

printf("enter position:");

scanf("%d",&pos);

printf("enter element:");

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

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

{

temp=temp->next;

}

newnode->next=temp->next;

newnode->prev=temp;

temp->next=newnode;

temp->next->prev=newnode;

printf("\none node inserted\n");

}

void delete\_specific()

{

int pos;

struct node \*temp;

temp=head;

printf("enter pos to delete:");

scanf("%d",&pos);

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

{

temp=temp->next;

}

temp->next->prev=temp->prev;

temp->prev->next=temp->next;

printf("\n node deleted!\n");

}

void find()

{

int ele,i=1;

struct node \*temp;

temp=head;

printf("enter element to find:");

scanf("%d",&ele);

while(temp->data!=ele)

{

temp=temp->next;

i=i+1;

}

printf("element found at %d",i);

}

void find\_prev()

{

int ele,i=1;

struct node \*temp;

temp=head;

printf("enter the previous of element to be found:");

scanf("%d",&ele);

while(temp->data!=ele)

{

temp=temp->next;

}

printf("the element is: %d",temp->prev->data);

}

void find\_next()

{

int ele,i=1;

struct node \*temp;

temp=head;

printf("enter the previous of element to be found:");

scanf("%d",&ele);

while(temp->data!=ele)

{

temp=temp->next;

}

printf("the element is: %d",temp->next->data);

}