#include<stdio.h>

#include<stdlib.h>

void createlist(int n);

void insert\_first(void);

void insert\_middle(void);

void insert\_last();

void delete\_first(void);

void delete\_middle(void);

void delete\_last();

void searching();

void finding\_even();

void reversing();

void sorting();

void copy();

void delete\_dup();

void display(void);

void newnode(void);

struct node{

int data;

struct node\* prev;

struct node\* next;

}\*temp,\*head,\*trav,\*tail,\*new\_temp,\*new\_head,\*new\_tail,\*trac;

int main(){

int press;

int n,i;

printf("Enter the number to create list:");

scanf("%d",&n);

createlist(n);

do{

printf("\n1.insert first\n2.insert middle\n3.insert last");

printf("\n4.delete first\n5.delete middle\n6.delete last");

printf("\n7.searching\n8.finding\_even\n9.reversing\n10.sorting\n11.copy");

int option;

printf("\nEnter the option:");

scanf("%d",&option);

switch(option)

{

case 1:

insert\_first();

break;

case 2:

insert\_middle();

break;

case 3:

insert\_last();

break;

case 4:

delete\_first();

break;

case 5:

delete\_middle();

break;

case 6:

delete\_last();

break;

case 7:

searching();

break;

case 8:

finding\_even();

break;

case 9:

reversing();

break;

case 10:

sorting();

break;

case 11:

copy();

break;

case 12:

delete\_dup();

break;

default:

printf("Enter the valid number;");

break;

}

display();

printf("\n\nIf you want to continue please Enter 's':");

scanf("%d",&press);

}while(press==1);

}

void createlist(int n)

{

int num;

for(int i=0;i<n;i++)

{

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

printf("Enter the node number:");

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

temp->prev=NULL;

temp->next=NULL;

if(temp==NULL)

{

printf("list is empty");

}

else{

if(head==NULL){

head=temp;

tail=temp;

}

else{

tail->next=temp;

temp->prev=tail;

tail=temp;

}

}

}

}

void newnode(){

int num;

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

printf("enter the number:");

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

temp->next=NULL;

}

void insert\_first(){

int n;

printf("Enter number to insert how many node;");

scanf("%d",&n);

for(int i=0;i<n;i++){

newnode();

if(head==NULL)

printf("List is empty");

else{

temp->next=head;

head->prev=temp;

head=temp;

}

}

}

void insert\_last(){

int n;

printf("Enter number to insert how many node;");

scanf("%d",&n);

for(int i=0;i<n;i++){

newnode();

if(head==NULL)

printf("List is empty");

else{

trav=head;

while(trav->next!=NULL)

{

trav=trav->next;

}

trav->next=temp;

temp->prev=trav;

}

}

}

void insert\_middle(){

newnode();

int data;

printf("Enter the data in the node to insert after:");

scanf("%d",&data);

if(head==NULL)

printf("List is empty");

else{

trav=head;

while(trav->data!=data)

{

trav=trav->next;

}

temp->next=trav->next;

temp->prev=trav;

trav->next->prev=temp;

trav->next=temp;

}

}

void delete\_first(){

if(head==NULL)

printf("list is empty");

else{

temp=head;

head=temp->next;

head->prev=NULL;

free(temp);

}

}

void delete\_middle(){

int data;

printf("Enter the data to delete:");

scanf("%d\n",&data);

if(head==NULL)

printf("list is empty");

else{

temp=head;

while(temp->data!=data)

{

trav=temp;

temp=temp->next;

}

trav->next=temp->next;

temp->next->prev=trav;

free(temp);

}

}

void delete\_last(){

if(head==NULL)

printf("list is empty");

else{

temp=head;

while(temp->next!=NULL)

{

trav=temp;

temp=temp->next;

}

trav=temp->prev;

trav->next=NULL;

free(temp);

}

}

void searching(){

int count=0,num;

printf("Enter the number to search:");

scanf("%d",&num);

if(head==NULL)

printf("List is empty");

else{

temp=head;

while(temp!=NULL){

if(num==temp->data)

{

count++;

//temp=temp->next;

}

temp=temp->next;

}

}

if(count>0)

printf("The number is found %d times",count);

else

printf("The number is not found in the node");

}

void finding\_even(){

int count=0;

if(head==NULL)

printf("List is empty");

else

temp=head;

while(temp!=NULL){

if(temp->data%2==0){

count++;

printf("The even number is %d\n",temp->data);

}

temp=temp->next;

}

if(count>0)

printf("%d even number is present in the node",count);

else

printf("NO even number is present");

}

void reversing(){

int x;

if(head==NULL)

printf("List is empty");

else{

temp=head;

while(temp!=NULL){

trav=temp->next;

while(trav!=NULL){

x=temp->data;

temp->data=trav->data;

trav->data=x;

trav=trav->next;

}

temp=temp->next;

}

}

}

void sorting(){

int x;

if(head==NULL)

printf("List is empty");

else{

temp=head;

while(temp!=NULL){

trav=temp->next;

while(trav!=NULL)

{

if(temp->data<trav->data)

{

x=temp->data;

temp->data=trav->data;

trav->data=x;

}

trav=trav->next;

}

temp=temp->next;

}

}

}

void copy(){

temp=head;

while(temp!=NULL){

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

new\_temp->data=temp->data;

new\_temp->next=NULL;

if(new\_temp==NULL)

printf("list is empty");

else{

if(new\_head==NULL){

new\_head=new\_temp;

new\_tail=new\_temp;

}

else{

new\_tail->next=new\_temp;

new\_tail=new\_temp;

}

}

temp=temp->next;

}

printf("\nThe copied list:");

if(new\_head==NULL)

{

printf("List is empty");

}

else

{

new\_temp=new\_head;

while(new\_temp!=NULL){

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

new\_temp=new\_temp->next;

}

}

}

void delete\_dup(){

if(head==NULL)

printf("List is empty");

else{

temp=head;

while(temp!=NULL){

trav=temp->next;

while(trav!=NULL){

tail=trav;

if(temp->data==trav->data){

trac=trav;

trav->prev->next=trav->next;

trav->next->prev=trav->prev;

free(trac);

}

trav=trav->next;

}

temp=temp->next;

}

}

}

/\*void occurrence(){

int data;

printf("Enter the data to found how many times in the list:");

scanf("%d",&data);

if(head==NULL)

printf("List is empty");

else{

while(temp!=NULL){

if(temp->data==data){

count++;

}

temp=temp-next;

}

}

}\*/

void display()

{

printf("\nThe list is\n");

if(head==NULL)

printf("List is empty");

else{

temp=head;

while(temp!=NULL)

{

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

temp=temp->next;

}

}

}