#include<stdio.h>

#include<stdlib.h>

//structure of single linked list

struct node

{

int data;

struct node\*link;

}\*p;

// add at head function

void add\_beg(int d){

if(p==NULL){

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

ptr->data=d;

ptr->link=NULL;

p=ptr;

}

else{

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

temp->data=d;

temp->link=p;

p=temp;

}}

// add at end function

void add\_end(int d){

if(p==NULL){

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

ptr->data=d;

ptr->link=NULL;

p=ptr;

}else{

struct node\*temp;

temp=p;

while(temp->link!=NULL){

temp=temp->link;

}struct node\*end=(struct node\*)malloc(sizeof(struct node));

end->data=d;

temp->link=end;

end->link=NULL;}

}

// add at specific position

void at\_pos(int d,int pos){

int c=0;

if(p==NULL){

if(pos==1){

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

ptr->data=d;

ptr->link=NULL;

p=ptr;

}

else{

printf("invalid position");

}

}else{

struct node\*temp,\*prev;

temp=p;

if(pos==1){

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

ptr->data=d;

ptr->link=temp;

p=ptr;

}else{

while(c!=pos-1){

c++;

prev=temp;

temp=temp->link;

}

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

ptr->data=d;

ptr->link=temp;

prev->link=ptr;

}

}}

// delete at end function

void del\_end()

{

struct node\*temp;

if(p!=NULL){

temp=p;

struct node\*ptr;

while(temp->link!=NULL){

ptr=temp;

temp=temp->link;

}

ptr->link=NULL;

free(temp);

}}

// delete at head function

void del\_beg(){

struct node \*temp;

if(p!=NULL){

temp=p;

p=p->link;

free(temp);

}

else{

printf("linked list not exit");

}}

// delete at specific position

void del\_atpos(int pos){

struct node\* temp = p;

int i;

if (pos == 0) {

printf("\nElement deleted is : %d\n", temp->data);

p = p->link;

temp->link = NULL;

free(temp); // Node is deleted

}

else {

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

temp = temp->link;

}

struct node\* del= temp->link;

temp->link = temp->link->link;

printf("\nElement deleted is : %d\n", del->data);

del->link = NULL;

free(del); // Node is deleted

}

}

// finding middle node

void middle(){

int c=0,mid=0;

struct node \*s=p;

while(s!=NULL){

c++;

s=s->link;

}

mid=c/2;

if(c&1==0){

mid-=1;

}

s=p;

while(mid>0){

s=s->link;

mid-=1;

}printf("middle value is:");

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

}

// mid by Two Pointer

void TwoPointer(){

struct node \*A,\*B;

A=p;

B=p;

while(B!=NULL &&(B->link!=NULL && B->link && B->link->link!=NULL)){

A=A->link;

B=B->link->link;

}

printf("middle node is:");

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

}

// nth node

void NthLastNode(int n)

{

int c=0;

struct node\*s=p;

while(s!=NULL){

c++;

s=s->link;

}

struct node \*front, \*back;

int i;

front = back = p;

if(n > c)

{

printf("Error : n is greater than length of Linked List\n");

}

for(i = 0; i < n-1; i++){

front = front->link;

}

/\* Now, move both pointers together till front reaches last node of linked list \*/

while(front->link != NULL){

front = front->link;

back = back->link;

}

printf("nth data is %d",back->data);

}

// print function

void print\_list(){

if(p==NULL){

printf("linked list does not exit.");

}

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

temp=NULL;

temp=p;

while(temp->link!=NULL)

{

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

temp=temp->link;

}

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

}

// driver function

int main(){

int choice;

while(1){

printf("\nchoose below:");

printf("\n1. add\_beg\n2. add\_end\n3. del\_beg\n4. del\_end\n5. middle\n6. mid by two pointer\n7. insert SpecificPosition\n8. delete SpecificPosition\n9. nth node\n10. print list\n11. Exit the Program.\n");

printf("\nEnter your Choice : ");

scanf("%d",&choice);

if(choice!=0){

switch(choice){

int m,i,p;

int n,d,pos;

case 1:

printf("\nenter how many data you want to enter:",m);

scanf("%d",&m);

for(i=1;i<=m;i++){

printf("\nenter element at %d:",i);

scanf("%d",&p);

add\_beg(p); }

break;

case 2:

printf("\nenter how many data you want to enter:",n);

scanf("%d",&n);

for(i=1;i<=n;i++){

printf("\nenter element at %d:",i);

scanf("%d",&d);

add\_end(d);}

break;

case 3:

del\_beg();

break;

case 4:

del\_end();

break;

case 5:

middle();

break;

case 6:

TwoPointer();

break;

case 7:

printf("enter position to insert:");

scanf("%d",&pos);

printf("enter data to insert:");

scanf("%d",&d);

at\_pos(d,pos);

break;

case 8:

printf("enter position to delete:");

scanf("%d",&pos);

del\_atpos(pos);

break;

case 9:

printf("enter value nth position:");

scanf("%d",&n);

NthLastNode(n);

break;

case 10:

print\_list();

break;

case 11:

printf("exit\n");

break;

default:

printf("\nPlease Select only 1-8 option ----\n");

break;

}}}}