#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*next;

};

//scanf("%d",&choice);

struct node \*head=NULL;

void add\_beg()

{

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

int val;

printf("Enter value\n");

scanf("%d",&val);

var->data=val;

var->next=NULL;

if(head==NULL)

{

head=var;

}

else

{

var->next=head;//

head=var;

}

}

void add\_end()

{

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

int val;

printf("Enter value\n");

scanf("%d",&val);

var->data=val;

var->next=NULL;

if(head==NULL)

{

head=var;

}

else

{

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

for(temp=head;temp->next!=NULL;temp=temp->next);

temp->next=var;

}

}

void add\_pos()

{

/\*memory allocation not required for temp here, when requerd\*/

int pos,i;

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

printf("Enter position you want to insert\n");

scanf("%d",&pos);

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

int val;

printf("Enter value\n");

scanf("%d",&val);

var->data=val;

for( i=1,temp=head;i<pos-1;temp=temp->next,i++);

//i<pos-1 because after end of loop temp again increases temp=temp->next

var->next=temp->next;

temp->next=var;

}

void ins\_val()

{

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

struct node \*temp;

int val,v,f=0,i,pos;

printf("Enter value after which you want to insert\n");

scanf("%d",&v);

for( i=1,temp=head;temp!=NULL;temp=temp->next,i++)

{

if(temp->data==v)

{

f=1;

break;

//since break is used(i++)no post increment of i

}

}

if(f==0)

{

printf("Value not found\n");

}

else

{

printf("Enter value\n");

scanf("%d",&val);

pos=i-1;

var->data=val;

for(i=1,temp=head;i<=pos;temp=temp->next,i++);

//i<=pos because after end of loop temp again increases temp=temp->next

var->next=temp->next;

temp->next=var;

}

}

void del\_beg()

{

if(head==NULL)

{

printf("The Linked list is empty");

}

else

{

struct node \*temp=head;

head= head->next;

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

temp->next=NULL;//making previous head next null,since we are deleting it anyway

free(temp);

}

}

void del\_end()

{

if(head==NULL)

{

printf("The Linked list is empty");

}

else

{

struct node \*temp,\*p;

for(temp=head;temp->next->next!=NULL;temp=temp->next);

p=temp->next;//(storing last value)

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

temp->next=NULL;

free(p);//(deleting last value)

}

}

void del\_pos()

{

if(head==NULL)

{

printf("The Linked list is empty");

}

else

{

int pos,i;

struct node \*p;

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

printf("Enter position you want to delete\n");

scanf("%d",&pos);

for(i=1,temp=head;i<=pos-2;temp=temp->next,i++);

//to find pos to be deleted

p=temp->next;

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

temp->next=p->next;

p=NULL;

free(p);

}

}

void rev()

{

struct node \*prevnode,\*currentnode,\*nextnode;

prevnode=NULL;

currentnode=nextnode=head;

while(nextnode!=NULL)

{

nextnode=nextnode->next;

currentnode->next=prevnode;

prevnode=currentnode;

currentnode=nextnode;

}

head=prevnode;

printf("The elements of Linked List are reversed\n");

}

void phy\_rev()

{

printf("After reversing the linkedlist: \n");

struct node \*end=NULL;

while(end!=head)

{

struct node \*temp;

for(temp=head;temp->next!=end;temp=temp->next)

{

}

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

end=temp;

}

}

void sel\_sort()

{

printf("The Linked list is sorted by Selection Sort: \n");

struct node \*temp;

struct node \*j;

int t;

for(temp=head;temp->next!=NULL;temp=temp->next)

{

for(j=temp->next;j!=NULL;j=j->next)

{

if(temp->data>j->data)

{

t=temp->data;

temp->data=j->data;

j->data=t;

}

}

}

}

void bub\_sort()

{

printf("The Linked list is sorted by Bubble Sort: \n");

struct node \*end=NULL;

struct node \*temp,\*j;

int t;

while(end!=head)

{

for(j=head;j->next!=end;j=j->next)

{

if(j->data>j->next->data)

{

t=j->data;

j->data=j->next->data;

j->next->data=t;

}

}

end=j;

}

}

void display()

{

printf("The elements of the Linked List are:\n");

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

for(temp=head;temp!=NULL;temp=temp->next)

{

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

}

}

int main()

{

int choice,loop=1;

//struct node \*head=NULL;

printf("Enter 1 to insert at beginning\n");

printf("Enter 2 to insert at ending\n");

printf("Enter 3 to insert at desired position\n");

printf("Enter 4 to insert after a given value\n");

printf("Enter 6 to delete from beginning\n");

printf("Enter 7 to delete from end\n");

printf("Enter 8 to delete from any position\n");

printf("Enter 9 to physically reverse Linked List\n");

printf("Enter 10 to display reverse Linked List\n");

printf("Enter 11 to implement Selecton Sort in Linked List\n");

printf("Enter 12 to implement Bubble Sort in Linked List\n");

printf("Enter 15 to display\n");

printf("Enter 20 to end\n");

while(loop)

{

printf("\nEnter choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

add\_beg();

break;

case 2:

add\_end();

break;

case 3:

add\_pos();

break;

case 4:

ins\_val();

break;

case 5:

del\_beg();

break;

case 6:

add\_end();

break;

case 7:

del\_end();

break;

case 8:

del\_pos();

break;

case 9:

rev();

break;

case 10:

phy\_rev();

break;

case 11:

sel\_sort();

break;

case 12:

bub\_sort();

break;

case 15:

display();

break;

case 20:

loop=0;

printf("Thanks for using Linked List");

break;

default:

printf("Enter valid input\n");

}

}

}