<<<<<<<<<<<INSERTION AND DELETION OF ELEMENTS IN LINKED LIST>>>>>>>>>>>>

#include<stdio.h>

#include<malloc.h>

struct node

{

int data;

struct node\*next;

};

struct node\*head=NULL;

void insert(int e)

{

struct node\*temp;

if(head==NULL)

{

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

head->data=e;

head->next=NULL;

}

else

{

temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=(struct node\*)malloc(sizeof(struct node));

temp->next->data=e;

temp->next->next=NULL;

}

}

void delete(int e)

{

struct node\*temp;

if(head==NULL)

{

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

}

else if(head->data==e)

{

head=head->next;

}

else

{

temp=head;

while(temp->next!=NULL&&temp->next->data!=e)

{

temp=temp->next;

}

if(temp->next==NULL)

{

printf("\nElement not found\n");

}

else

{

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

}

}

}

void disp()

{

struct node\*temp;

if(head==NULL)

{

printf("\nlist is empty");

}

else

{

temp=head;

while(temp!=NULL)

{

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

temp=temp->next;

}

}

}

int main()

{

insert(10);

insert(20);

insert(30);

disp();

delete(20);

printf("\n");

disp();

delete(10);

printf("\n");

disp();

delete(100);

}

<<<<<<<<SEARCHING AN ELEMENT IN LINKED LIST>>>>>>>>>>>>

#include<stdio.h>

#include<malloc.h>

struct node

{

int data;

struct node\*next;

};

struct node\*head=NULL;

void insert(int e)

{

struct node\*temp;

if(head==NULL)

{

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

head->data=e;

head->next=NULL;

}

else

{

temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=(struct node\*)malloc(sizeof(struct node));

temp->next->data=e;

temp->next->next=NULL;

}

}

void search(int key)

{

int flag=0;

struct node \* temp;

temp=head;

while (temp != NULL)

{

if(key==temp->data)

{

flag=1;

break;

}

temp=temp->next;

}

if (flag==1)

{

printf("\nelement found");

}

else

{

printf("\nelement not found");

}

}

void disp()

{

struct node\*temp;

if(head==NULL)

{

printf("\nlist is empty");

}

else

{

temp=head;

while(temp!=NULL)

{

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

temp=temp->next;

}

}

}

int main()

{

int key;

insert(10);

insert(20);

insert(30);

disp();

printf("\nenter key element:");

scanf("%d",&key);

search(key);

}

<<<<<<<<<<<SORTING ELEMENTS IN LINKED LIST>>>>>>>>>>>>

#include<stdio.h>

#include<malloc.h>

struct node

{

int data;

struct node\*next;

};

struct node\*head=NULL;

void insert(int e)

{

struct node\*temp;

if(head==NULL)

{

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

head->data=e;

head->next=NULL;

}

else

{

temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=(struct node\*)malloc(sizeof(struct node));

temp->next->data=e;

temp->next->next=NULL;

}

}

void sort()

{

int t;

struct node\*t1,\*t2;

for(t1=head;t1->next!=NULL;t1=t1->next)

{

for(t2=head;t2->next!=NULL;t2=t2->next)

{

if(t2->data>t2->next->data)

{

t=t2->data;

t2->data=t2->next->data;

t2->next->data=t;

}

}

}

}

void disp()

{

struct node\*temp;

if(head==NULL)

{

printf("\nlist is empty");

}

else

{

temp=head;

while(temp!=NULL)

{

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

temp=temp->next;

}

}

}

int main()

{

int key;

insert(50);

insert(20);

insert(30);

insert(15);

insert(45);

disp();

sort();

printf("\nThe Sorted List is :");

disp();

}