1:Stack Using Linked List

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*top=NULL;

void push()

{

struct node \*ne;

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

if(ne==NULL)

printf("\nStack is full\n");

else

{

printf("\nEnter the element to insert:");

int i;

scanf("%d",&i);

ne->data=i;

ne->next=NULL;

if(top==NULL)

{

top=ne;

}

else

{

ne->next=top;

top=ne;

}

}

}

void display()

{

struct node \*top1=top;

while(top1->next!=NULL)

{

printf("%d->",top1->data);

top1=top1->next;

}

printf("%d",top1->data);

}

void pop()

{

if(top==NULL)

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

else

{

struct node \*temp=top;

printf("%d is poped!",top->data);

top=top->next;

free(temp);

}

}

void search()

{

struct node \*top2=top;

int x,c=1;

printf("\nEnter the element to find:");

scanf("%d",&x);

while(top2!=NULL)

{

if(top2->data==x)

{

printf("\nElement found!! and it is %dth term!",c);

break;

}

top2=top2->next;

c++;

}

if(top2==NULL)

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

}

void main()

{

int c,x;

while(c>0)

{

printf("\nSelect:\n1:Push\n2:Pop\n3:Display\n4:Search\n5:Exit\n");

scanf("%d",&c);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

switch(c)

{

case 1:push();

break;

case 2:pop();

break;

case 3:display();

break;

case 4:search();

break;

case 5:c=-1;

break;

default:c=-1;

}

}

}

OUTPUT:

