//STACK USING LINKED LIST

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*top=NULL;

void push();

void display();

void pop();

void search();

int main()

{

int opt;

do

{

printf("\n menu\n1.push\n2.pop\n3.display\n4.search\n5.exit\n");

printf("enter option\n");

scanf("%d",&opt);

switch(opt)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

search();

break;

}

}

while(opt!=5);

}

void push()

{

int x;

struct node \*ne;

printf("enter element\n");

scanf("%d",&x);

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

if(ne==NULL)

{

printf("stack overflow");

return;

}

ne->data=x;

ne->next=top;

top=ne;

return;

}

void pop()

{

struct node \*ptr;

if(top==NULL)

{

printf("stack is empty");

return;

}

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

ptr=top;

top=top->next;

free(ptr);

return;

}

void display()

{

struct node \*ptr=top;

if(top==NULL)

{

printf("stack empty");

return;

}

while(ptr!=NULL)

{

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

ptr=ptr->next;

}

}

void search()

{

int s;

struct node \*ptr=top;

printf("enter element to be searched");

scanf("%d",&s);

if(top==NULL)

{

printf("stack empty");

return;

}

while(ptr!=NULL)

{

if(s==ptr->data){

printf("element found");

break;

}

ptr=ptr->next;

}

}

OUTPUT:



