//implementaion of stack using linked list

#include <stdio.h>

#include <stdlib.h>

void push();

void pop();

void display();

struct node

{

int val;

struct node \*next;

};

struct node \*head;

int main()

{

int choice;

while(1)

{

printf("Operation performed by stack");

printf("\n1.push\n2.pop\n3.display\n");

scanf("%d",&choice);

switch(choice)

{

case 1: push(); //push operator

break;

case 2: pop(); //pop operator

break;

case 3: display(); //display operator

break;

case 4: exit(0); //terminates the program

default:printf("\n Invalid option");

}}}

void push()

{

int val;

struct node \*ptr=(struct node\*)malloc(sizeof(struct node)); //allocating the memory

if(ptr==NULL)

{

printf("Not able to push the elment");

}

else

{

printf("Enter the value \n"); //entering the value to be pushed into the stack

scanf("%d",&val);

if(head==NULL)

{

ptr->val=val;

ptr->next=NULL;

head=ptr;

}

else

{

ptr->val=val;

ptr->next=head;

head=ptr;

}

printf("\n Item pushed \n"); //item or the value is being is pushed

}}

void pop()

{

int item;

struct node \*ptr;

if(head==NULL)

{

printf("\nUnderflow \n"); //underflow condition

}

else

{

item=head->val;

ptr=head;

head=head->next;

free(ptr);

printf("\n Item popped \n"); //item is being popped out of the stack

}}

void display()

{

int i;

struct node\*ptr;

ptr=head;

if(ptr==NULL)

{

printf("\n Stack is empty \n"); //displays the stack is empty

}

else

{

printf("\n Printing stack element \t \n");

while(ptr!=NULL)

{

printf("%d\t",ptr->val);

ptr=ptr->next;

}}}