***Stack using array***

#include <stdio.h>

#define max 10

int stack[max], top;

void display(int []);

void push(int [], int);

int pop(int []);

int main() {

int i = 0;

int choice=0, res;

top=-1;

while(1){

printf("\nEnter choice\n");

printf("1: Display\n");

printf("2: Push\n");

printf("3: Pop\n");

scanf("%d",&choice);

switch (choice) {

case 1:

display(stack);

break;

case 2:

printf("Enter element to insert: " );

scanf("%d",&i);

push(stack,i);

break;

case 3:

res=pop(stack);

if(res==-1)

printf("No element to pop");

else

printf("Element removed");

break;

default: printf("Invalid choice" );

break;

}

}

return 0;

}

void display(int stack[]){

int i=0;

if (top==-1){

printf("stack is empty\n");

return;

}

for (int i = top; i >= 0; i--) {

printf("\n%d",stack[i]);

}

}

void push(int stack[], int item){

if(top==max-1){

printf("\nStack full");

return;

}

top++;

stack[top]=item;

}

int pop(int stack[]){

int deletedi;

if (top==-1){

printf("stack is empty");

return -1;

}

deletedi=stack[top];

top--;

return deletedi;

}

***Stack using structure***

#include <stdio.h>

#define max 10

struct stackStr{

int stack[max];

int top;

};

typedef struct stackStr STACK;

STACK s;

void display();

void push();

int pop();

int main() {

int i = 0;

int choice=0, res;

s.top=-1;

while(1){

printf("\nEnter choice\n");

printf("1: Display\n");

printf("2: Push\n");

printf("3: Pop\n");

scanf("%d",&choice);

switch (choice) {

case 1:

display();

break;

case 2:

printf("Enter element to insert: " );

scanf("%d",&i);

push(i);

break;

case 3:

res=pop();

if(res==-1)

printf("No element to pop");

else

printf("Element removed");

break;

default: printf("Invalid choice" );

break;

}

}

return 0;

}

void display(){

int i=0;

if (s.top==-1){

printf("stack is empty\n");

return;

}

for (int i = s.top; i >= 0; i--) {

printf("\n%d",s.stack[i]);

}

}

void push(int ele){

if(s.top==max-1){

printf("\nStack full");

return;

}

s.top++;

s.stack[s.top]=ele;

}

int pop(){

int deletedi;

if (s.top==-1){

printf("stack is empty");

return -1;

}

deletedi=s.stack[s.top];

s.top--;

return deletedi;

}