#include<iostream>

using namespace std;

#define STACKSIZE 5

struct bufferstack

{

int stk[STACKSIZE];

int top; // We will use it as pointer to top of the stack

} pointer; // Here pointer is struct variable, you can see here how to implement structure in C

void push(); // To push elements in stack

int pop(); // To Pop elements in stack

void display();

int main()

{

int c;

pointer.top=-1; //Set pointer to -1

int x=1;

while(x) //While loop to keep program in loop

{

//cout<<"\n Pointer is at : \t"<<pointer.top+1<<endl; //this line is to test, the position of pointer

cout<<"\n Please enter your choice :"<<endl<<"1: Push"<<endl<<"2: Pop"<<endl<<"3: Display"<<endl<<"4: Exit"<<endl;

cin>>c;

switch(c)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

return 0;

}

cout<<" \n Do you want to cotin....? press 1 or 0"<<endl;

cin>>x;

}

}

void push()

{

int num;

if(pointer.top==(STACKSIZE-1))

{

cout<<"\n Sorry You can't push any element into stack .... ,Stack is full";

}

else

{

cout<<"\n Enter the number to push into stack";

cin>>num;

pointer.top+=1;

pointer.stk[pointer.top]= num;

}

}

int pop()

{

int num;

if(pointer.top==-1)

{

cout<<"\nstack is empty "<<endl;

}

else

{

num=pointer.top;

cout<<"\n Poped number is : "<<pointer.stk[num];

pointer.stk[num]=0; //To delete top element from stack

pointer.top-=1;

}

return num;

}

void display()

{

if(pointer.top==-1)

{

cout<<"\n Stack is empty"<<endl;

}

else

{

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

{

cout<<"\n"<<i<<":"<<pointer.stk[i]<<endl;

}

// cout<<pointer.top;

}

}

#include<iostream>

using namespace std;

const int Max\_Len=100;

//enum boolean {false ,true };

struct Stack

{

char s[Max\_Len];

int top;

};

void reset (Stack \*stk)

{

(\*stk).top=0;

}

void push (char c, Stack \*stk)

{

(\*stk).top++;

(\*stk).s[(\*stk).top]=c;

cout<<(\*stk).s[(\*stk).top];

}

char pop (Stack \*stk)

{

return (stk->s[stk->top--]);

}

int main ()

{

Stack s1;

char x='a';

push(x,&s1);

return 0;

}