#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef char Elemtype;

typedef struct node {

Elemtype data;//数据域

struct node \*next;//指针域

} SqStack;

SqStack \*top;

SqStack \*InitStack() {

SqStack \*top;

top = NULL;

return top;

}

void DestroyStack() {

SqStack \*temp;

while (top != NULL) {

temp = top;

top = top->next;

free(temp);

}

}

void Push(Elemtype e) {

SqStack \*node;

node = (Elemtype \*)malloc(sizeof(Elemtype));

if (node != NULL) {

node->next = top;

node->data = e;

top = node;

} else

printf("Push error!\n");

}

Elemtype Pop() {

Elemtype x;

SqStack \*temp;

temp = top;

x = top->data;

top = top->next;

free(temp);

return x;

}

Elemtype GetTop() {

Elemtype x;

x = top->data;

return x;

}

int StackEmpty() { //判断是否为空栈,是返回1，否 返回0

if (top == NULL)

return 1;

else

return 0;

}

int StackTraverse() {//链栈里面暂时只能倒序逐个输出，要想顺序输出的话，可以建成双向链表

SqStack \*temp = top;

if (top == NULL) {

printf("栈为空!\n");

return 0;

} else {

while (temp != NULL) {

printf("%c ", temp->data);

temp = temp->next;

}

return 1;

}

}

main() {

printf("测试链栈初始化的操作：");

top = InitStack();

if (top == NULL)

printf("初始化成功！\n");

Elemtype e, ch[15], ch1[15];

puts("请输入要压入栈中的字符串:");

scanf("%s", ch);

for (int i = 0; i < strlen(ch); i++) {

printf("存入第%d个字符：%c\n", i + 1, ch[i]);

Push(ch[i]);

}

printf("测试从栈中访问元素:\n");

Elemtype e1;

// e1 = GetTop(top);

// printf("栈顶的元素为：%c\n", e1);

printf("请确认您的输入:");

StackTraverse();

printf("\n");

///

printf("测试逐个弹出栈顶元素:\n");

int a;//储存每次弹出的元素

while (StackEmpty() != 1) { //如果栈非空

a = Pop();

printf("%c ", a);

}

printf("\n");

//测试ClearStack(SqStack \*s)清空栈操作

puts("重新为栈赋值:");

puts("请输入要压入栈中的字符串:");

scanf("%s", ch1);

for (int i = 0; i < strlen(ch1); i++) {

printf("存入第%d个字符：%c\n", i + 1, ch1[i]);

Push(ch1[i]);

}

//验证下是否重新入栈成功

printf("请确认您的输入:");

StackTraverse();

printf("\n");

DestroyStack();

printf("执行后结果:");

StackTraverse();

}