#include<stdio.h>

#include<stdlib.h>

#define OK 1

#define ERROR 0

#define TRUE 1

#define FALSE 0

#define MAXSIZE

typedef int Status;//函数返回值的状态

typedef int SElemType;//SElemType表示栈的数据元素，在这里定义

//链栈结构

typedef struct StackNode

{

SElemType data;

struct StackNode \*next;

}StackNode, \*LinkStackPtr;//LinkStackPtr被定义成：“struct StackNode \*”，即struct StackNode的指针

typedef struct LinkStack

{

LinkStackPtr top;

int count;

}LinkStack;

//输出元素e

Status visit(SElemType e)

{

printf("%d ", e);

return OK;

}

//构造一个空栈S

Status InitStack(LinkStack \*S)

{

S->top = (LinkStackPtr)malloc(sizeof(StackNode));

if (!S->top)

return ERROR;

S->top = NULL;

S->count = 0;

return OK;

}

//把栈置为空栈

Status ClearStack(LinkStack \*s)

{

LinkStackPtr p, q;

p = s->top;

while (p)

{

q = p;

p = p->next;

free(q);//释放内存

}

s->count = 0;

return OK;

}

//ClearStack错误代码如下，编译不会报错，但s.count的值，在主函数里的栈长度值（count)不会改变。有兴趣的可以试试

/\*Status ClearStack(LinkStack s)

{

LinkStackPtr p, q;

p = s.top;

while (p)

{

q = p;

p = p->next;

free(q);//释放内存

}

s.count = 0;

return OK;

}\*/

//判断是否是空栈

Status StackEmpty(LinkStack s)

{

if (s.count == 0)

return TRUE;

else

return FALSE;

}

//返回栈的元素个数，即长度

int StackLength(LinkStack s)

{

return s.count;

}

//若栈不为空则返回元素e,并返回OK,否则返回FALSE

Status GetTop(LinkStack s, SElemType \*e)

{

if (s.top != NULL)

{

\*e = s.top->data;

return OK;

}

return FALSE;

}

//插入元素e为新的栈顶元素

Status Push(LinkStack \*S, SElemType e)

{

LinkStackPtr s = (LinkStackPtr)malloc(sizeof(StackNode));

s->data = e;

s->next = S->top;//把当前的栈顶元素赋值给新结点的直接后继

S->top = s;//将新结点赋值给栈顶指针

S->count++;

return OK;

}

//若栈不为空，则删除S的栈顶元素，用e返回其值，并返回OK,否则返回ERROR

Status Pop(LinkStack \*S, SElemType \*e)

{

LinkStackPtr p;

if (S->top == NULL)

return ERROR;

\*e = S->top->data;

p = S->top;//将栈顶赋值给p

S->top = S->top->next;//使得栈顶指针下移一位，指向后一结点

free(p);

S->count--;

return OK;

}

//输出栈元素

Status StackTraverse(LinkStack s)

{

LinkStackPtr p;

p = s.top;

while (p)

{

visit(p->data);

p = p->next;

}

printf("\n");

return OK;

}

int main()

{

int j;

LinkStack s;

int e;

if (InitStack(&s) == OK)

{

for (j = 1; j <= 10; j++)

Push(&s, j);

}

printf("栈中元素依次为：");

StackTraverse(s);

Pop(&s, &e);

printf("弹出的栈顶元素 e=%d\n", e);

printf("栈是否为空：%d(1:空 0：否)\n", StackEmpty(s));

GetTop(s, &e);

printf("栈顶元素 e=%d 栈的长度为%d\n", e, StackLength(s));

ClearStack(&s);

printf("栈清空后，栈空否：%d(1:空 0：否)\n", StackEmpty(s));

system("pause");

return 0;

}