# 实验3 栈的应用

班级：2019级 计算机科学与技术 专业3班 学号 20190521340 姓名 周吉瑞

**实验日期**：2020 年 4 月 14 日 **报告日期**：2020 年 4 月 14 日

### 实验目的

掌握栈的结构特性及其入栈，出栈操作；会利用栈实现各种算法。

### 实验预习

复习以下概念

顺序栈：顺序栈是栈的顺序实现。顺序栈是指利用顺序存储结构实现的栈。采用地址连续的存储空间（数组)依次存储栈中数据元素，由于人栈和出栈运算都是在栈顶进行，而栈底位置是固定不变的，可以将栈底位置设置在数组空间的起始处；栈顶位置是随入栈和出栈操作而变化的，故需用一个整型变量top来记录当前栈顶元素在数组中的位置。

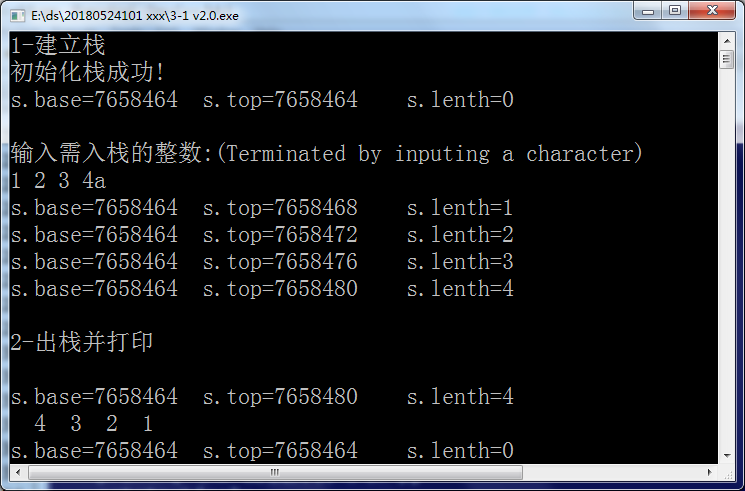
链栈：链式栈是一种数据存储结构，可以通过单链表的方式来实现，使用链式栈的优点在于它能够克服用数组实现的顺序栈空间利用率不高的特点，但是需要为每个栈元素分配额外的指针空间用来存放指针域。

循环队列：为充分利用向量空间，克服"[假溢出](https://baike.baidu.com/item/%E5%81%87%E6%BA%A2%E5%87%BA" \t "_blank)"现象的方法是：将向量空间想象为一个首尾相接的圆环，并称这种向量为循环向量。存储在其中的队列称为循环队列（Circular Queue）。循环队列是把[顺序队列](https://baike.baidu.com/item/%E9%A1%BA%E5%BA%8F%E9%98%9F%E5%88%97/20832734" \t "_blank)首尾相连，把存储队列元素的表从逻辑上看成一个环，成为循环队列。

链队：队列的链式存储结构，其实就是线性表的单链表，只不过它只是**尾进头出**而已，我们把它简称为链队列。为了操作上的方便，我们将队头指针指向链队列的头结点，而队尾指针指向终端节点。

### 实验内容和要求

1.阅读下面程序，将函数Push、函数Pop、及其他函数补充完整。若输入元素序列1 2 3 4a，运行结果如下所示。



（1）头文件**intstack.h：**

/\*顺序栈的实现

文件名：intstack.h

\*/

#include<iostream>

#include<fstream>

using namespace std;

//顺序栈定义

#define OK 1

#define ERROR 0

#define OVERFLOW -2

#define MAXSIZE 100//顺序栈存储空间的初始分配量

typedef int Status;

typedef int SElemType;

typedef struct {

SElemType \*base;//栈底指针

SElemType \*top;//栈顶指针

int stacksize;//栈可用的最大容量

} SqStack;

//算法3.1　顺序栈的初始化

Status InitStack(SqStack &S) {

//构造一个空栈S

S.base = new SElemType[MAXSIZE];//为顺序栈动态分配一个最大容量为MAXSIZE的数组空间

if (!S.base)

exit(OVERFLOW); //存储分配失败

S.top = S.base ; //top初始为base，空栈

S.stacksize = MAXSIZE; //stacksize置为栈的最大容量MAXSIZE

return OK;

}

//算法3.2　顺序栈的入栈

Status Push(SqStack &S, SElemType e) {

// 插入元素e为新的栈顶元素

if (S.top - S.base == MAXSIZE)

return ERROR;

\*(S.top) = e;

S.top++;

return OK;

}

//算法3.3　顺序栈的出栈

Status Pop(SqStack &S, SElemType &e) {

//删除S的栈顶元素，用e返回其值

if (S.top == S.base)

return ERROR;

S.top--;

e = \*(S.top);

return OK;

}

//算法3.4　取顺序栈的栈顶元素

SElemType GetTop(SqStack S) {//返回S的栈顶元素，不修改栈顶指针

if (S.top != S.base) //栈非空

return \*(S.top - 1); //返回栈顶元素的值，栈顶指针不变

}

////算法3.5　判断栈是否为空

int StackEmpty(SqStack S)

//若栈为空，返回1 ，否则返回0

{if ( S.top == S.base )

return 1;

else

return 0;

}

void prt\_base\_top\_length(SqStack s)

//输出栈的参数

{

printf("s.base=%d s.top=%d s.length=%d\n",s.base,s.top,s.top-s.base);

}

（2）CPP文件：（请命名为： 学号+姓名+实验3-1.cpp）

#include<stdio.h>

#include "intstack.h"

void PrintStack(SqStack &S){

SElemType e;

while(!StackEmpty(S))

{ Pop(S,e);

printf("%3d",e);

}

printf("\n");

}/\*Pop\_and\_Print\*/

int CreateStack(SqStack &S){

int e;

if(InitStack(S))

{printf("初始化栈成功!\n");

prt\_base\_top\_length(S);

}

else{

printf("初始化失败!\n");

return ERROR;

}

printf("\n输入需要入栈的各个整数:(Terminated by inputing a character)\n");

while(scanf("%d",&e))

{ Push(S,e);

prt\_base\_top\_length(S);

};

return OK;

}/\*CreateStack\*/

int main(){

SqStack ss;

printf("1-建立栈\n");

CreateStack(ss);

printf("\n2-出栈并打印\n");

prt\_base\_top\_length(ss);

PrintStack(ss);

prt\_base\_top\_length(ss);

PrintStack(ss);

return 0;

}

**算法分析：**

1. 输入元素序列1 2 3 4，为什么输出序列为4 3 2 1？体现了栈的什么特性？

体现了栈的“先进后出”特性

1. 为什么第二次输出栈时，栈为空栈？ 如果需要保持栈的原样，应该如何修改程序？

因为输出前执行了出栈函数，修改了top指针，最终变为了空栈，要修改须要将出栈函数替换为读取栈元素的函数（即：返回栈元素的值，但不要修改指针）

如下：将Pop函数改为Get函数：

SElemType Get(SqStack S, int n)

{

//返回S的元素，不修改top指针

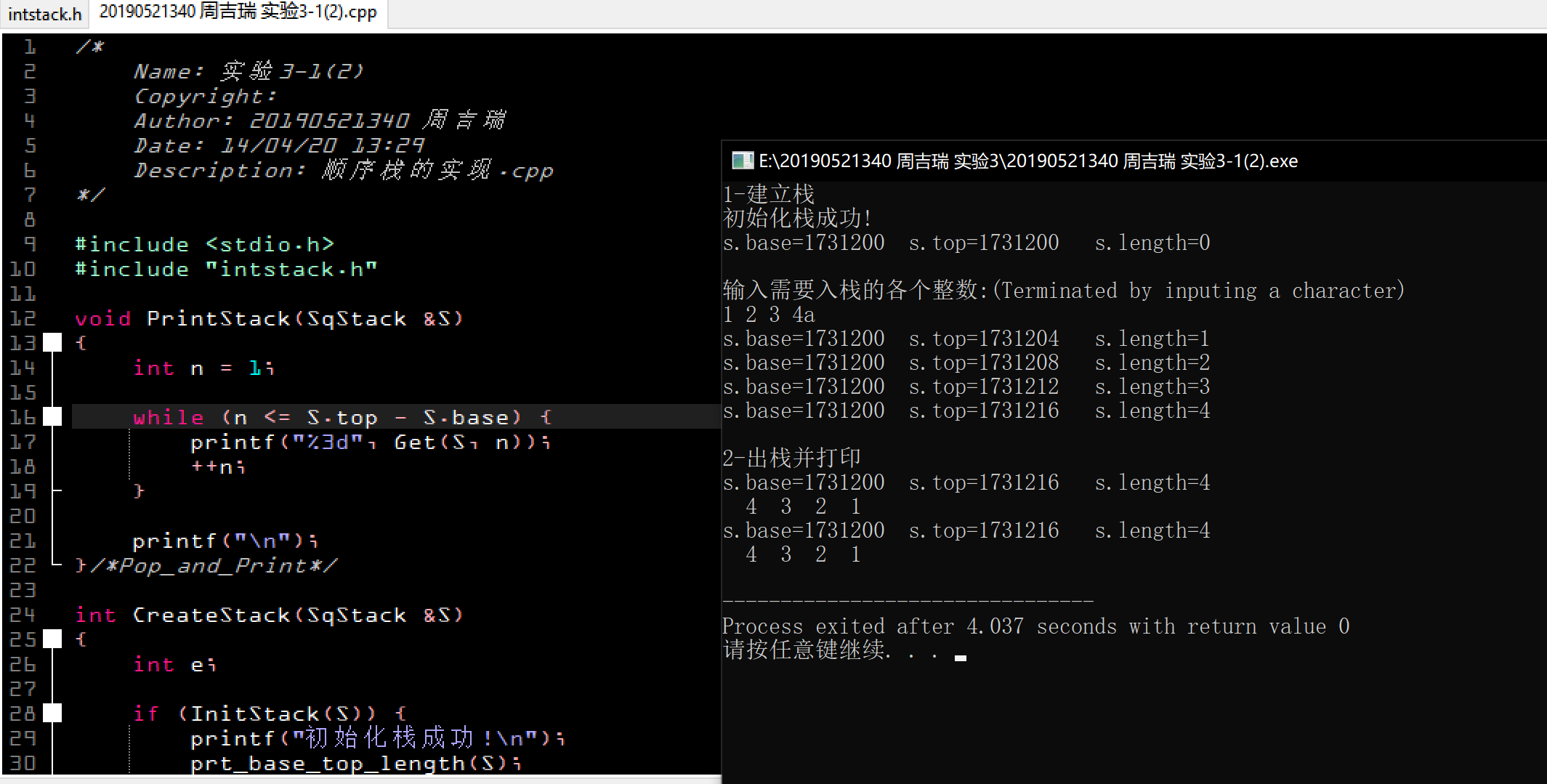
if (S.top != S.base) //栈非空

return \*(S.top - n); //返回栈元素的值，top指针不变

}

//具体代码见：实验3-1(2).cpp intstack.h

运行结果：



2.参考教材算法3.20，编写一个十进制转换为八进制的数制转换程序（要求利用栈来实现），并验证其正确性。如输入1348，输出2504

（请命名为： 学号+姓名+实验3-2.cpp）

提示：可使用intstack.h

实现代码：

/\*

Name: 实验3-2

Copyright:

Author: 20190521340 周吉瑞

Date: 14/04/20 13:54

Description: 编写一个十进制转换为八进制的数制转换程序（要求利用栈来实现），

并验证其正确性。如输入1348，输出2504

\*/

#include <stdio.h>

#include "intstack.h"

int main()

{

int n;

SqStack S;

SElemType e;

printf("input (10 jinzhi):");

scanf("%d", &n);

InitStack(S);

while (n) {

Push(S, n % 8);

n = n / 8;

}

cout << "(8 jinzhi):";

while (!StackEmpty(S)) {

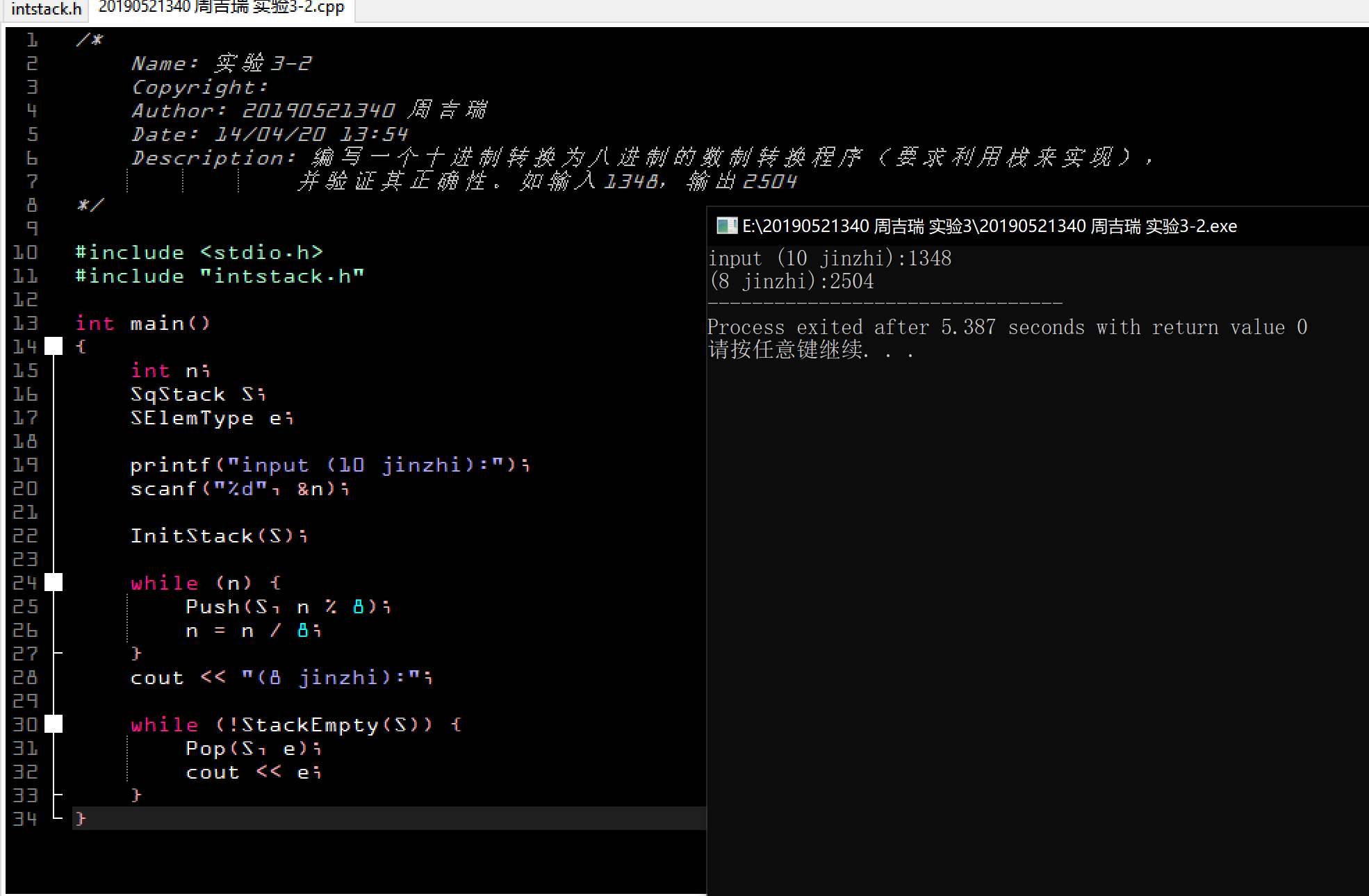
Pop(S, e);

cout << e;

}

}//具体代码见：实验3-2

验证：



3．（选做）参考教材算法3.21，设计一个判断表达式括号是否匹配的算法，并编程实现。

（请命名为： 学号+姓名+实验3-3.cpp）

如： 输入：2+[(c-d)\*6-(f-7)\*a)]/6 输出：不匹配

输入：a-[(c-d)\*6-(s/3-x)]/2 输出：匹配

/\*

Name: 实验3-3

Copyright:

Author: 20190521340 周吉瑞

Date: 14/04/20 14:24

Description: 设计一个判断表达式括号是否匹配的算法，并编程实现。

如： 输入：2+[(c-d)\*6-(f-7)\*a)]/6 输出：NO

输入：a-[(c-d)\*6-(s/3-x)]/2 输出：YES

\*/

#include <stdio.h>

#include "intstack.h"

int main()

{

char ch;

SqStack S;

SElemType e;

int f = 1;

InitStack(S);

printf("input formula Enter is end:");

while ((ch = getchar()) != '\n') {

if (ch == '(' || ch == '[' || ch == '{')

Push(S, ch);

else {

if (ch == ')') {

if (StackEmpty(S))

f = 0;

else {

Pop(S, e);

if (e != '(')

f = 0;

}

}

if (ch == ']') {

if (StackEmpty(S))

f = 0;

else {

Pop(S, e);

if (e != '[')

f = 0;

}

}

if (ch == '}') {

if (StackEmpty(S))

f = 0;

else {

Pop(S, e);

if (e != '{')

f = 0;

}

}

}

}

if (f) {

if (StackEmpty(S))

printf("YES\n");

else

printf("NO\n");

}

else {

printf("NO\n");

}

return 0;

}

### 实验小结