《数据结构》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 | 2016010207 | 姓名 | 孙博文 | 专业 | 石油工程 | 时间 | 2018.5.13 |

**一、实验题目 顺序栈基本操作**

**二、实验目的**

**1.熟练掌握顺序栈的实现和基本操作**

**2.理解栈后进先出的特点**

**3.熟练应用顺序栈解决实际问题**

**三、算法设计分析**

（一）数据结构的定义

（二）总体设计

实验由主函数、栈初始化函数、返回栈顶元素函数、入栈函数、出栈函数、销毁函数6个函数共同组成。其功能描述如下：

（1）主函数：实现功能选择界面，统筹调用各个函数以实现相应功能，

void main()

（2）栈初始化函数：初始化一个空栈

status Initstack(SqStack &s)

1. 返回栈顶元素函数：返回栈顶元素的值

status GetTop(SqStack s,SElemType &e)

1. 入栈函数：将一个输入元素入栈

status Push(SqStack &s,SElemType &e)

1. 出栈函数：将栈顶元素从栈顶弹出，用一个变量保存它的值

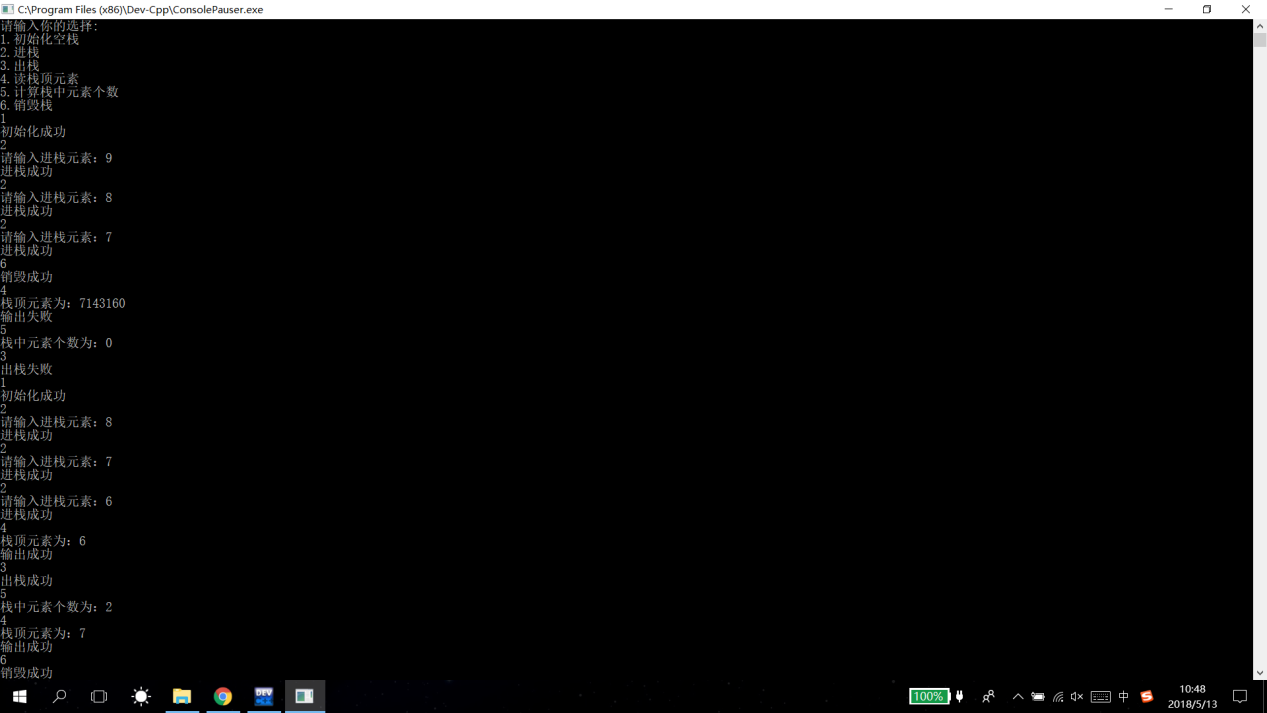
status Pop(SqStack &s,SElemType &e)

1. 销毁函数：销毁目标栈

status Delete(SqStack &s)

**四、实验测试结果及结果分析**

（一）测试结果（此处给出程序运行截图）



（二）结果分析

**五、实验总结**

刚开始我以为是用数字来实现顺序栈，但是后来才明白是用动态内存。销毁栈的操作有些难度，因为我刚开始以为要直接销毁栈底指针，后来又看了一遍数据结构，才发现定义时每一个栈元素都是一个节点，所以需要遍历整个栈，依次释放

**附录 实验程序代码(该部分请加注释)**

#include <bits/stdc++.h>

#define STACK\_INIT\_SIZE 100

#define OK 1

#define ERROR -1

#define OVERFLOW -2

//#define OVERFLOW -2

using namespace std;

typedef int status;

typedef int SElemType;

typedef struct{

SElemType \*base;

SElemType \*top;

int stacksize;

}SqStack;

status Initstack(SqStack &s) { //初始化空栈

s.base=(SElemType\*)malloc(STACK\_INIT\_SIZE\*sizeof(SElemType));

if (!s.base) exit(OVERFLOW);

s.top=s.base;

s.stacksize=STACK\_INIT\_SIZE;

return OK;

}

status GetTop(SqStack s,SElemType &e) { //返回栈顶元素

if (s.top==s.base) return ERROR;

e=\*(s.top-1);

return OK;

}

status Push(SqStack &s,SElemType &e) { //入栈

if (!s.base) exit(OVERFLOW);

if (s.top-s.base>=s.stacksize) { //如果栈即将溢出，就重新分配内存，容量为原先容量的两倍

s.base=(SElemType\*)realloc(s.base,2\*s.stacksize\*sizeof(SElemType));

if (!s.base) exit(OVERFLOW);

s.top=s.base+s.stacksize;

s.stacksize\*=2;

}

\*s.top++=e;

return OK;

}

status Pop(SqStack &s,SElemType &e) { //出栈

if (s.top==s.base) return ERROR;

e=\*(--s.top);

return OK;

}

status Delete(SqStack &s) { //销毁栈

if (s.top==s.base) return OK;

s.top--;

while (s.top!=s.base) {

free(s.top--);

}

free(s.base);

return OK;

}

SqStack s;

int main() {

cout << "请输入你的选择:" <<endl;

cout << "1.初始化空栈" <<endl;

cout << "2.进栈" <<endl;

cout << "3.出栈" <<endl;

cout << "4.读栈顶元素" <<endl;

cout << "5.计算栈中元素个数" <<endl;

cout << "6.销毁栈" <<endl;

int n;

SElemType e;

cin >> n;

while (n) {

switch(n) {

case 1: {

int temp=Initstack(s);

if (temp==OK) printf("初始化成功\n");

else printf("初始化失败\n");

break;

}

case 2: {

printf("请输入进栈元素：");

SElemType t;

cin >> t;

int temp=Push(s,t);

if (temp==OK) printf("进栈成功\n");

else printf("进栈失败\n");

break;

}

case 3: {

int temp=Pop(s,e);

if (temp==OK) printf("出栈成功\n");

else printf("出栈失败\n");

break;

}

case 4: {

printf("栈顶元素为：");

int temp=GetTop(s,e);

cout << e <<endl;

if (temp==OK) printf("输出成功\n");

else printf("输出失败\n");

break;

}

case 5: {

printf("栈中元素个数为：%d\n",s.top-s.base);

break;

}

case 6: {

if (Delete(s)) printf("销毁成功\n");

else printf("销毁失败\n");

break;

}

}

cin >> n;

}

return 0;

}

**实验报告雷同者均视为未做。抄袭请慎重！**

【注意】上机实验报告只提交电子版，要求在课堂上完成并提交。