苏州城市学院实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算科学与人工智能学院 | | 年级专业 | | 22物联网 | | 姓名 | 王子超 | 学号 | 2200443011 |
| 课程名称 | | 数据结构 | | | | | | | 成绩 |  |
| 指导教师 | |  | | 同组实验者 | | 无 | | 实验日期 | 2024/3/25 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验四 实现顺序线性栈 |

顺序栈代码：

#include<iostream>

#include<malloc.h>

using namespace std;

#define N 100

#define add 10

typedef struct {

int \*base;

int \*top;

int stacksize;

}SeqStack;

void init(SeqStack &L){

L.base = (int \*)malloc(sizeof(int)\*N);

L.top = L.base;

L.stacksize = N;

}

bool empty(SeqStack &L){

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

return true;

}

return false;

}

void getelem(SeqStack &L){

if(!empty(L)){

int x = \*L.top;

cout<<"栈顶元素是:"<<x<<endl;

}else{

cout<<"栈顶为空，没有元素!"<<endl;

}

}

int push(SeqStack &L,int x){

if(L.top-L.base > N){

L.base = (int \*)realloc(L.base,(sizeof(int)\*(L.stacksize+add)));

L.top = L.base + L.stacksize;

L.stacksize +=add;

}

L.top++;

\*L.top = x;

}

bool pop(SeqStack &L){

if(empty(L)){

cout<<"栈顶为空，没有元素!"<<endl;

return true;

}

int x = \*L.top;

L.top--;

cout << "删除的栈顶元素是:"<<x<<endl;

return true;

}

int main(){

SeqStack A;

init(A);//初始化

push(A,1);//插入元素

getelem(A);//获取栈顶元素

pop(A);//删除栈顶元素

getelem(A);

return 0;

}

括号配对代码：

#include<iostream>

#include<string>

#include<malloc.h>

using namespace std;

#define N 100

#define add 10

typedef struct {

char \*base;

char \*top;

int stacksize;

}SeqStack;

void init(SeqStack &L){

L.base = (char \*)malloc(sizeof(int)\*N);

L.top = L.base;

L.stacksize = N;

}

bool empty(SeqStack &L){

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

return true;

}

return false;

}

void getelem(SeqStack &L){

if(!empty(L)){

char x = \*L.top;

cout<<"栈顶元素是:"<<x<<endl;

}else{

cout<<"栈顶为空，没有元素!"<<endl;

}

}

int push(SeqStack &L,char x){

if(L.top-L.base > N){

L.base = (char \*)realloc(L.base,(sizeof(int)\*(L.stacksize+add)));

L.top = L.base + L.stacksize;

L.stacksize +=add;

}

L.top++;

\*L.top = x;

}

bool pop(SeqStack &L){

if(empty(L)){

cout<<"栈顶为空，没有元素!"<<endl;

return true;

}

char x = \*L.top;

L.top--;

cout << "删除的栈顶元素是:"<<x<<endl;

return true;

}

bool judge(string a/\*char a[]\*/,int length){

SeqStack A;

init(A);

for(int i=0;i<length;i++){

char left;

if(a[i]=='(' || a[i]=='['){

push(A,a[i]);

}else if(a[i]==')' || a[i]==']'){

if(empty(A))

return false;

left = \*A.top;

A.top--;

if( (left == ')' && a[i] == '(') || (left == ']' && a[i] == '[') ){

pop(A);

}

}

}

if(empty(A)){

cout<<"匹配成功!"<<endl;

return true;

}

else{

cout<<"匹配失败!"<<endl;

return false;

}

}

int strlength(char a[]){

int count = 0;

while(a[count]!='\0'){

count++;

}

return count;

}

int main(){

// char a[]= "(1+[1+2\*7-(3+4)/7]";

// int length = strlength();

// judge(a,length);

string s;

getline(cin, s);

int length = s.length();

judge(s,length);

return 0;

}

后缀表达式代码：

#include<iostream>

#include<string>

#include<malloc.h>

using namespace std;

#define N 100

#define add 10

typedef struct {

char \*base;

char \*top;

int stacksize;

}SeqStack;

void init(SeqStack &L){

L.base = (char \*)malloc(sizeof(int)\*N);

L.top = L.base;

L.stacksize = N;

}

bool empty(SeqStack &L){

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

return true;

}

return false;

}

int getelem(SeqStack &L){

if(!empty(L)){

int x = \*L.top;

return x;

}else{

cout<<"栈顶为空，没有元素!"<<endl;

}

return 0;

}

int push(SeqStack &L,int x){

if(L.top-L.base > N){

L.base = (char \*)realloc(L.base,(sizeof(int)\*(L.stacksize+add)));

L.top = L.base + L.stacksize;

L.stacksize +=add;

}

L.top++;

\*L.top = x;

}

bool pop(SeqStack &L){

if(empty(L)){

cout<<"栈顶为空，没有元素!"<<endl;

return true;

}

int x = \*L.top;

L.top--;

return true;

}

void postfix(string a,int length){

SeqStack A;

init(A);

for(int i=0;i<length;i++){

if(isdigit(a[i])){ //isdigit(a[i])是C++的方法

int num = a[i] - '0';

push(A,num);

}else if(a[i]=='+' || a[i]=='-' ||a[i]=='\*' ||a[i]=='/'){

int num1 = getelem(A);

pop(A);

int num2 = getelem(A);

pop(A);

switch(a[i]){

case '+':

push(A,num1+num2);

break;

case '-':

push(A,num2-num1);

break;

case '\*':

push(A,num1\*num2);

break;

case '/':

push(A,num2/num1);

break;

}

}

}

int n = getelem(A);

cout<<"计算结果是:"<<n<<endl;

}

int main(){

// 1341+\*+5-

string s;

getline(cin, s);

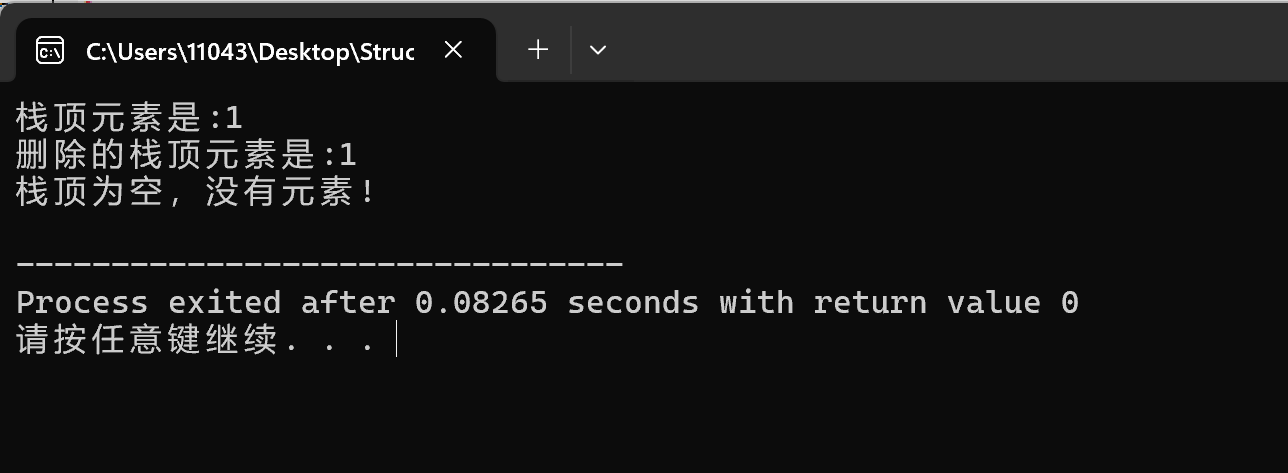
int length = s.length();

postfix(s,length);

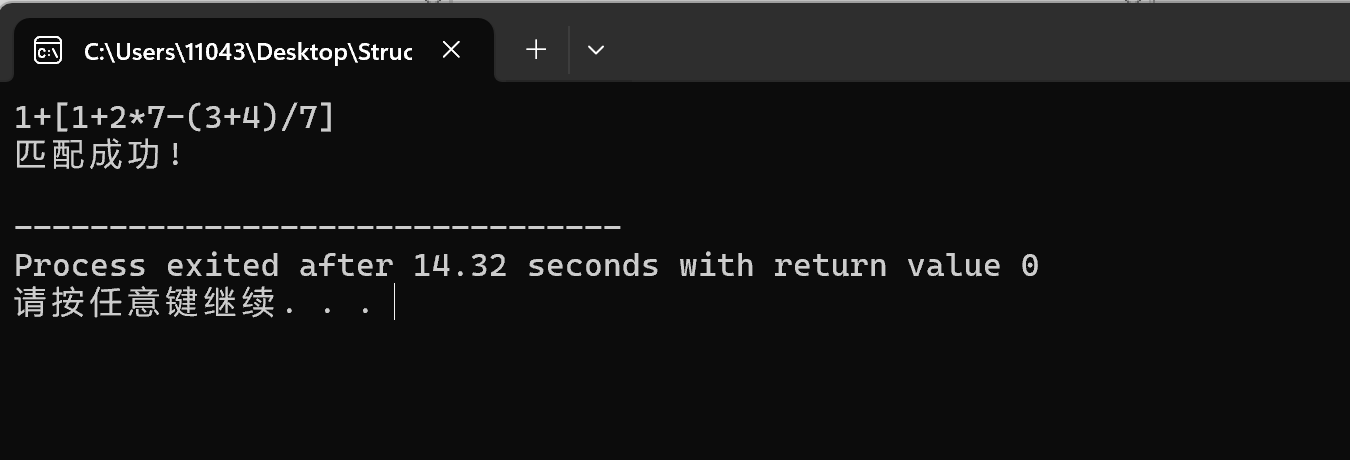
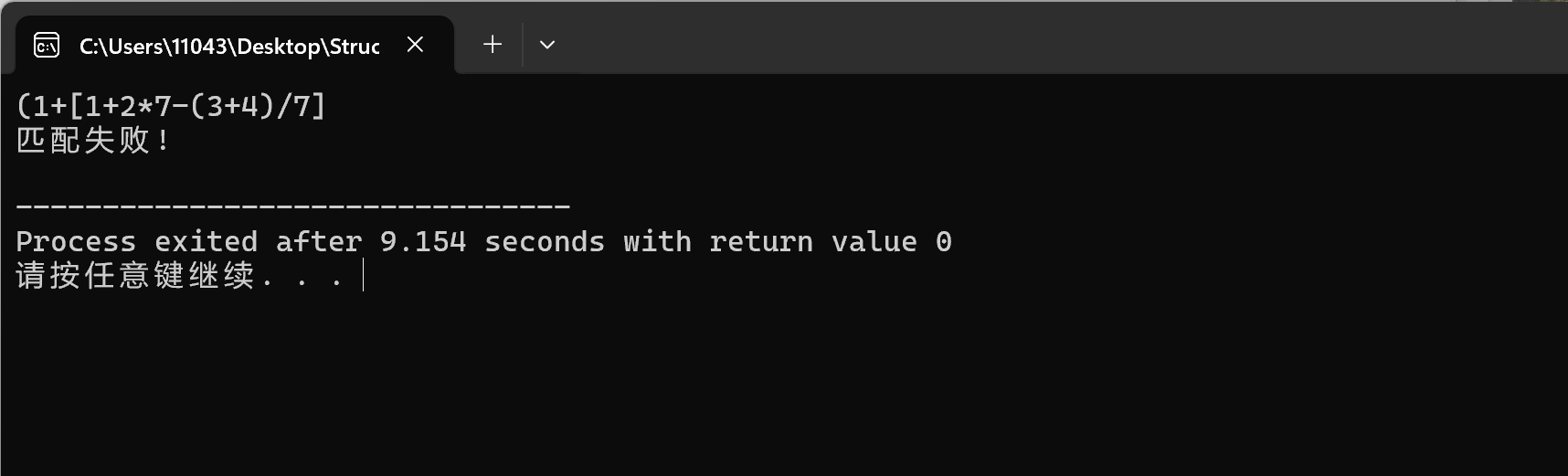
return 0;

}

顺序栈运行结果



括号配对运行结果



后缀表达式运行结果

