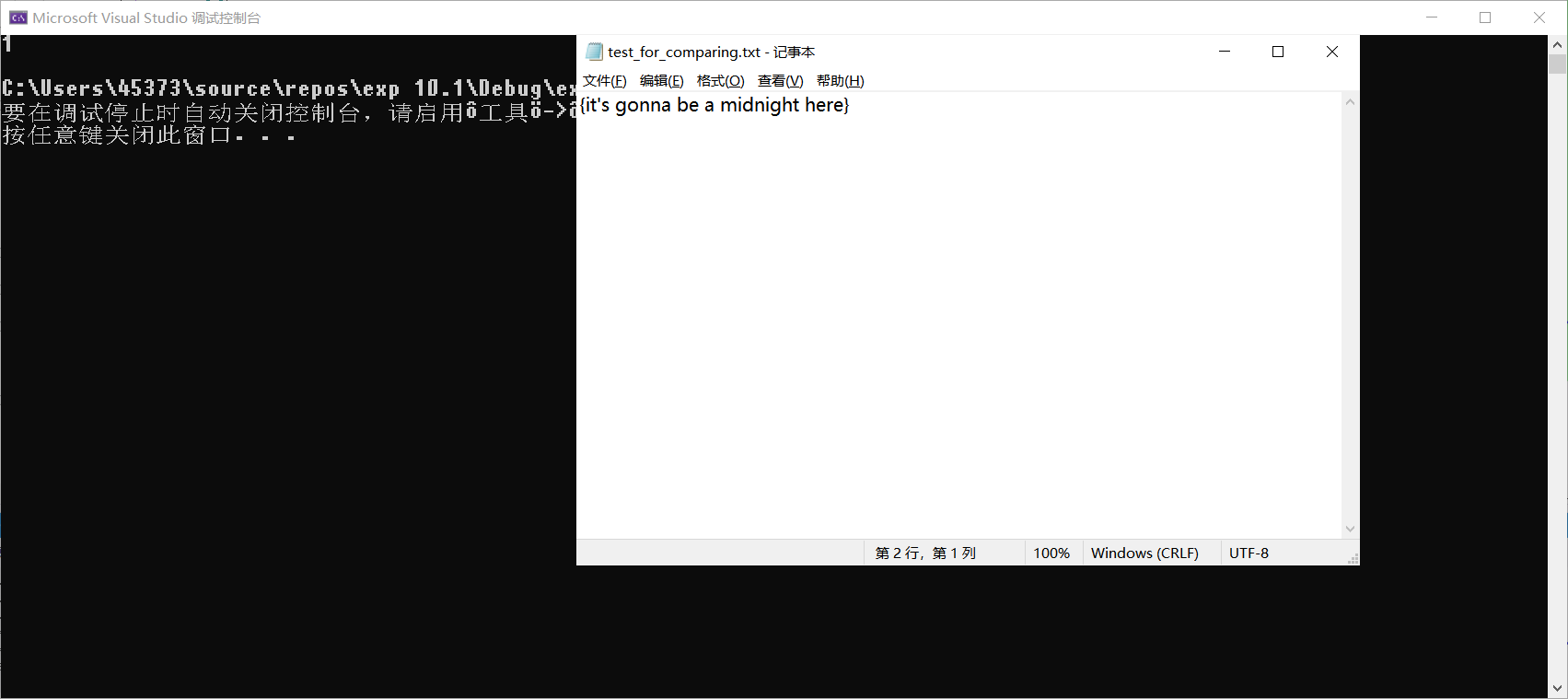
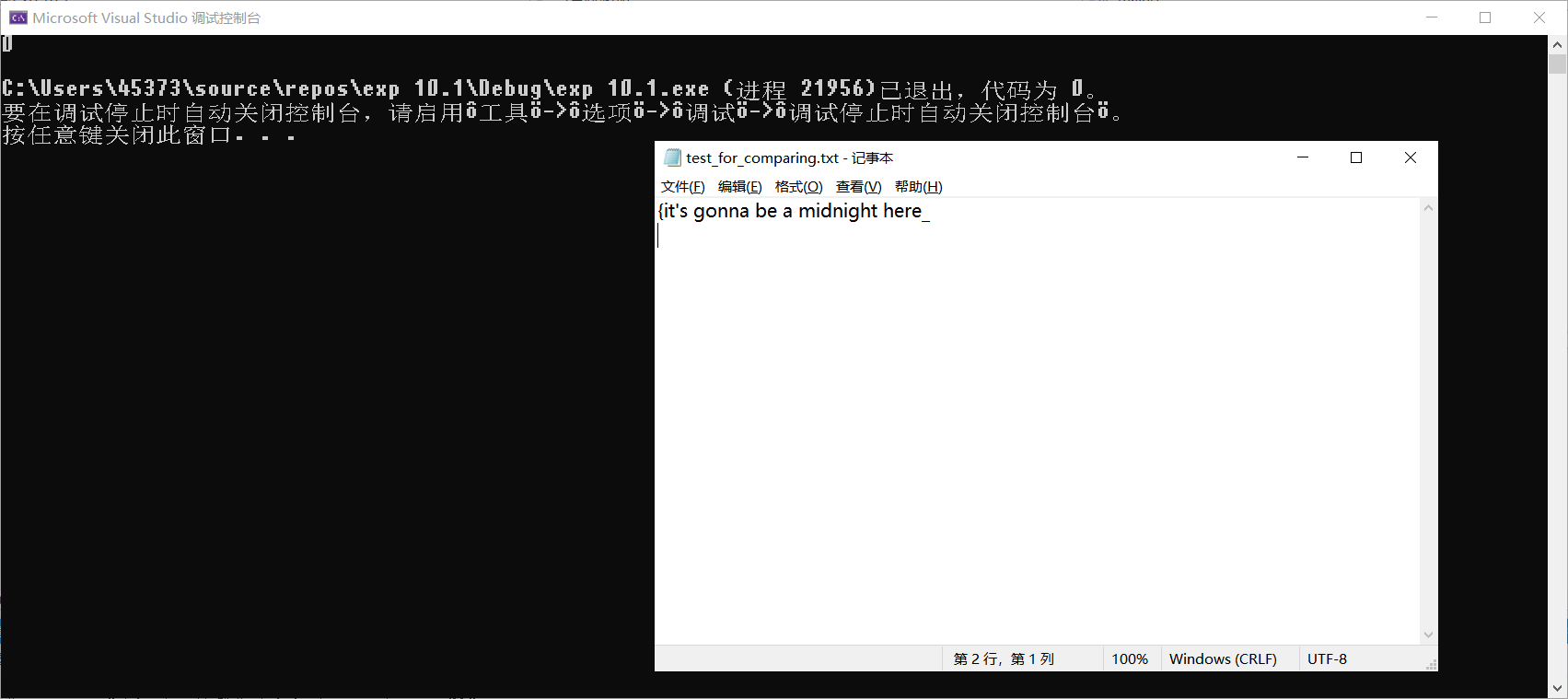
# 实验十 面向对象程序设计实践

### 题目1 堆栈与简单语法检查

如图，当为花括号时，匹配结果为1（true）



当不匹配时输出结果为0，同理()和[]亦相同



主函数代码：

#include <iostream>

#include <fstream>

#include <string>

#include "Stack.h"

//这里elemtype假设为char型

using namespace std;

int main()

{

Stack iStack;

ifstream fin;

string str;

bool result = false;

iStack.InitStack();

fin.open("C:/Users/45373/Desktop/private file/c++实验课题目/10\_面向对象程序设计实践/test\_for\_comparing.txt");

if (fin.fail())

{

cout << "File open error!";

exit(1);

}

int i = 0;

while (getline(fin, str)) //检查{}配对情况

{

i = 0;

while ('\0' != str[i])

{

if ('{' == str[i])

{

iStack.Push('{');

}

else if ('}' == str[i] && '{' == iStack.Peek())

{

iStack.Pop();

}

i++;

}

}

/\* while (getline(fin, str)) //检查[]配对情况

{

i = 0;

while ('\0' != str[i])

{

if ('[' == str[i])

{

iStack.Push('{');

}

else if (']' == str[i] && '[' == iStack.Peek())

{

iStack.Pop();

}

i++;

}

}\*/

/\*while (getline(fin, str)) //检查()配对情况

{

i = 0;

while ('\0' != str[i])

{

if ('(' == str[i])

{

iStack.Push('(');

}

else if (')' == str[i] && '(' == iStack.Peek())

{

iStack.Pop();

}

i++;

}

}\*/

result = iStack.IsEmpty(); //配对情况是栈空，如果result为1说明配对

cout << result << endl;

return 0;

}

头文件：

#pragma once

const int MaxSize = 500;

class Stack

{

private:

char stack[MaxSize];

char\* top;

int len;

public:

Stack();

void InitStack();

void Push(char item);

char Pop();

char Peek();

bool IsEmpty();

void ClearStack();

};

头文件.cpp

#include "Stack.h"

#include <iostream>

using namespace std;

Stack::Stack()

{

InitStack();

}

void Stack::InitStack()

{

len = 0;

top = stack;

}

void Stack::Push(char item)

{

if (len < 500)

{

stack[len] = item;

top = &stack[len];

len++;

}

else

{

cout << "The stack is full." << endl;

}

}

char Stack::Pop()

{

if (!IsEmpty())

{

char temp = \*top;

top = &stack[len - 1];

len--;

return temp;

}

else

{

cout << "The stack is empty." << endl;

return '\0';

}

}

char Stack::Peek()

{

if (!IsEmpty())

{

return \*top;

}

else

{

cout << "The stack is empty." << endl;

return '\0';

}

}

bool Stack::IsEmpty()

{

return (len == 0);

}

void Stack::ClearStack()

{

len = 0;

top = stack;

}