# 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

### 实验描述：

编写一个程序，利用堆栈

来检查一个C/C++语言程序中的花括号、方括号和圆括号是否配对，若能够完全配对返回1，

否则返回0

### 实验代码：

#### 1stack.cpp

/\*

@Author: 王贤义

 \* @FilePath: \code\C+++C\c++\10\1stack.cpp

@Description: 堆栈实现及括号匹配检查

\*/

#include <iostream>

#include <vector>

using namespace std;

const int MaxSize = 100; // 定义堆栈最大容量

typedef char ElemType;

class Stack

{

private:

    ElemType stack[MaxSize]; // 堆栈数组

    int top;                 // 堆栈指针，指向栈顶元素

public:

    Stack()

    { // 构造函数，初始化堆栈指针

        top = -1;

    }

    void InitStack()

    { // 初始化堆栈

        top = -1;

    }

    bool IsEmpty()

    { // 判断堆栈是否为空

        return top == -1;

    }

    bool IsFull()

    { // 判断堆栈是否已满

        return top == MaxSize - 1;

    }

    void push(ElemType item)

    { // 元素进栈

        if (IsFull())

        {

            cout << "Stack is full" << endl;

            return;

        }

        top++;

        stack[top] = item;

    }

    ElemType pop()

    { // 弹出栈顶元素并返回其值

        if (IsEmpty())

        {

            cout << "Stack is empty" << endl;

            return -1;

        }

        ElemType item = stack[top];

        top--;

        return item;

    }

    ElemType peek()

    { // 返回栈顶元素的值，但不移动栈顶指针

        if (IsEmpty())

        {

            cout << "Stack is empty" << endl;

            return -1;

        }

        return stack[top];

    }

    void ClearStack()

    { // 清空堆栈

        top = -1;

    }

};

/\*\*

@description: 检查括号匹配

@param {string} exp 待检查的表达式

@return {bool}

\*/

bool checkBalance(string exp)

{

    Stack s;

    for (int i = 0; i < exp.size(); i++)

    {

        char c = exp[i];

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

        {

            s.push(c);

        }

        else if (c == ')' || c == '}' || c == ']')

        {

            if (s.IsEmpty())

            {

                return false;

            }

            char top = s.pop();

            if ((c == ')' && top != '(') || (c == '}' && top != '{') || (c == ']' && top != '['))

            {

                return false;

            }

        }

    }

    return s.IsEmpty();

}

int main()

{

    string exp;

    while (true)

    {

        cout << "Please input an expression (input 'exit' to exit): ";

        cin >> exp;

        if (exp == "exit")

        {

            break;

        }

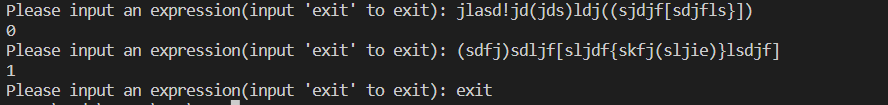
        cout << checkBalance(exp) << endl;

    }

    return 0;

}

### 实验结果：



### 思考与扩展：

无

## 题目2 成绩管理系统

### 实验描述：

现有学生成绩信息，内容如下

姓名 学号 语文数学 英语

张明明 01 67 78 82

李成友 02 78 91 88

张辉灿 03 68 82 56

王露 04 56 45 77

陈东明 05 67 38 47

…. .. .. .. …

请用C/C++编写一系统，实现学生信息管理，软件的入口界面应包括如下几个方面

的

1、功能要求：

（1）信息维护：

要求：学生信息数据要以文件的形式保存，能实现学生信息数据的维护。此模块包括子模块有：增加学生信息、删除学生信息、修改学生信息

（2）信息查询：

要求：查询时可实现按姓名查询、按学号查询

（3）成绩统计：

要求：A 输入任意的一个课程名（如数学）和一个分数段（如60--70），统计出在此分数段的学生情况。

（4）排序：能对用户指定的任意课程名，按成绩升序或降序排列学生数据并显示排序结果（使用表格的形式显示排序后的输出结果）

### 实验代码：

### 实验结果：

### 思考与扩展：

## 题目3 通信录管理系统

### 实验描述：

用 C/C++设计出模拟手机通信录管理系统，实现对手机中的通信录进行管理。

### 实验代码：

### 实验结果：

### 思考与扩展：

## 题目4 学生管理系统

### 实验描述：

使用下面的数据，用 C/C++设计一个简单的学籍管理系统，实现出最基本的功能。

学生基本信息文件(A.TXT)及其内容：A.TXT 文件不需要编程录入数据，可用文本编辑工具

直接生成

学号 姓名 性别 宿舍号码电话号码

01 张成成男 501 87732111

02 李成华女 101 87723112

03 王成凤女 101 87723112

04 张明明男 502 87734333

05 陈东 男 501 87732111

06 李果 男 502 87734333

07 张园园女 102 87756122

… …. .. … ………..

学生成绩基本信息文件(B.TXT)及其内容：

学号 课程编号课程名称 学分 平时成绩实验成绩 卷面成绩 综合成绩

01 A01 大学物理 3 66 78 82

02 B03 高等数学 4 78 -1 90

01 B03 高等数学 4 45 -1 88

02 C01 VF 3 65 76 66

… …. ………. .. .. …

### 实验代码：

### 实验结果：

### 思考与扩展：