# 北京邮电大学软件学院

# 2019-2020学年第1学期实验报告

**课程名称： 算法与数据结构**

**实验名称： 实验二**

**实验完成人：**

**姓名：**\_\_王衔飞\_\_\_**学号：**\_\_\_\_\_\_\_\_**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_贾红娓 \_李璐路\_\_\_\_\_\_\_\_

**日 期： 2019 年 10 月 10 日**

1. **实验目的**

仅仅认识到栈和队列是两种特殊的线性表是远远不够的，本次实验的目的在于使学生深入了解栈和队列的特征，以便在实际问题背景下灵活运用它们；同时还将巩固这两种结构的构造方法，接触较复杂问题的递归算法设计。

1. **实验内容**

**必做内容**

1. **数制转换问题**

**[问题描述]**

将十进制数N和其它d进制数的转换是计算机实现计算的基本问题，其解决方案很多，

其中最简单方法基于下列原理：即除d取余法。例如：（1348）10=（2504）8

|  |  |  |
| --- | --- | --- |
| **N** | **N div 8** | **N mod 8** |
| 1348 | 168 | 4 |
| 168 | 21 | 0 |
| 21 | 2 | 5 |
| 2 | 0 | 2 |

从表中我们可以看出，最先产生的余数4是转换结果的最低位，这正好符合栈的特性即后进先出的特性。所以可以用顺序栈来模拟这个过程。

**[基本要求]**

对于键盘输入的任意一个非负的十进制整数，打印输出与其等值的八进制数。由于上述

的计算过程是从低位到高位顺序产生的八进制数的各个数位，而打印输出，一般来说应从高位到低位进行，恰好和计算过程相反。因此可以先将计算过程中得到的八进制数的各位进栈，待相对应的八进制数的各位均产生以后，再使其按顺序出栈，并打印输出。即得到了与输入的十进制数相对应的八进制数。

**[测试数据]**

　　由学生依据软件工程的测试技术自己确定。注意测试边界数据。

1. **括号匹配的检验**

**[问题描述]**

假设表达式中允许有两种括号：圆括号和方括号，其嵌套的顺序随意，即（（）[ ]）或

[（[ ] [ ]）]等为正确格式，[（ ]）或（（（]均为不正确的格式。检验括号是否匹配的方法可用“***期待的紧迫程度***”这个概念来描述。例如：考虑下列的括号序列：

　　[　(　[　]　[　]　)　]

　　1　2　3　4　5　6　7　8

当计算机接受了第1个括号以后，它期待着与其匹配的第8个括号的出现，然而等来的却是第2个括号，此时第1个括号“[”只能暂时靠边，而迫切等待与第2个括号相匹配的第7个括号“）”的出现，类似的，因只等来了第3个括号“[”，此时，其期待的紧迫程度较第2个括号更紧迫，则第2个括号只能靠边，让位于第3个括号，显然第3个括号的期待紧迫程度高于第2个括号，而第2个括号的期待紧迫程度高于第1个括号；在接受了第4个括号之后，第3个括号的期待得到了满足，消解之后，第2个括号的期待匹配就成了最急迫的任务了，…… ，依次类推。可见这个处理过程正好和栈的特点相吻合。

**[基本要求]**

　　读入圆括号和方括号的任意序列，输出“匹配”或“此串括号匹配不合法”。

**[测试数据]**

　　输入（[ ]（）），结果“匹配”

　　输入 [（ ）]，结果“此串括号匹配不合法”

**[实现提示]**

设置一个栈，每读入一个括号，若是左括号，则作为一个新的更急迫的期待压入栈中；

若是右括号，并且与当前栈顶的左括号相匹配，则将当前栈顶的左括号退出，继续读下一个括号，如果读入的右括号与当前栈顶的左括号不匹配，则属于不合法的情况。在初始和结束时，栈应该是空的。

**选作内容**

1. **停车场管理**

**[问题描述]**

设停车场内只有一个可停放n辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车

在停车场内按车辆到达时间的先后顺序，依次由北向南排列（大门在最南端，最先到达的第一辆车停放在车场的最北端），若车场内已停满n辆汽车，则后来的汽车只能在门外的便道上等候，一旦有车开走，则排在便道上的第一辆车即可开入；当停车场内某辆车要离开时，在它之后开入的车辆必须先退出车场为它让路，待该辆车开出大门外，其它车辆再按原次序进入车场，每辆停放在车场的车在它离开停车场时必须按它停留的时间长短交纳费用。试为停车场编制按上述要求进行管理的模拟程序。

**[测试数据]**

设n=2，输入数据为：(‘A’,1,5),(‘A’,2,10),(‘D’,1,15),(‘A’,3, 20),

(‘A’,4,25),(‘A’,5,30),(‘D’,2,35),(‘D’,4,40),(‘E’,0,0)。每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码及到达或离去的时刻，其中，’A’表示到达；’D’表示离去，’E’表示输入结束。

**[基本要求]**

以栈模拟停车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟

管理。每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码及到达或离去的时刻，对每一组输入数据进行操作后的输出数据为：若是车辆到达，则输出汽车在停车场内或便道上的停车位置；若是车离去；则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费)。栈以顺序结构实现，队列以链表实现。

**[实现提示]**

需另设一个栈，临时停放为给要离去的汽车让路而从停车场退出来的汽车，也用顺序存

储结构实现。输入数据按到达或离去的时刻有序。栈中每个元素表示一辆汽车，包含两个数据项：汽车的牌照号码和进入停车场的时刻。

**[思考]**

　　（1） 两个栈共享空间，思考应开辟数组的空间是多少？

（2） 汽车可有不同种类，则它们的占地面积不同，收费标准也不同，如1辆客车和

1.5辆小汽车的占地面积相同，1辆十轮卡车占地面积相当于3辆小汽车的占地面积。

（3） 汽车可以直接从便道上开走，此时排在它前面的汽车要先开走让路，然后再依次

排到队尾。

（4） 停放在便道上的汽车也收费，收费标准比停放在停车场的车低，请思考如何修改

结构以满足这种要求。

1. **实验环境**

OS：macOS 10.15 IDE：Clion 2019.2 Compiler：GCC8.1

1. **实验结果**
2. 进制转换

测试了几个数据以及边界数据0，测试结果正确，完成试验任务。

2. 括号匹配检验：

测试了不同的表达式，其中包括能够匹配的、不能匹配的、空字符串以及非法表达式，都能进行正确的判断及处理。测试数据见附录。完成试验任务。

3. 停车场

按照题目中的测试数据进行测试，目测结果正确，完成试验任务。

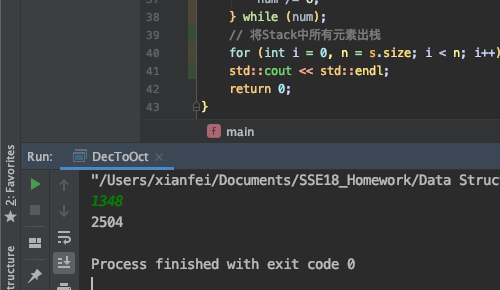
1. **附录**
2. 进制转换（十进制转八进制）

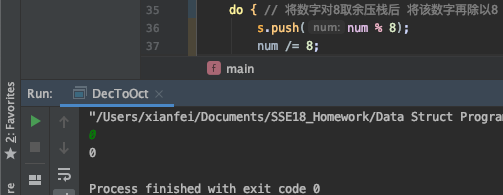
结构：一个堆栈

算法：将数字对8取余压栈后，再将该数字除以8，直到该数字等于零，之后弹出堆栈中所有元素即为要求的八进制数。

测试数据：题目中的1348，边界数据0

截图：





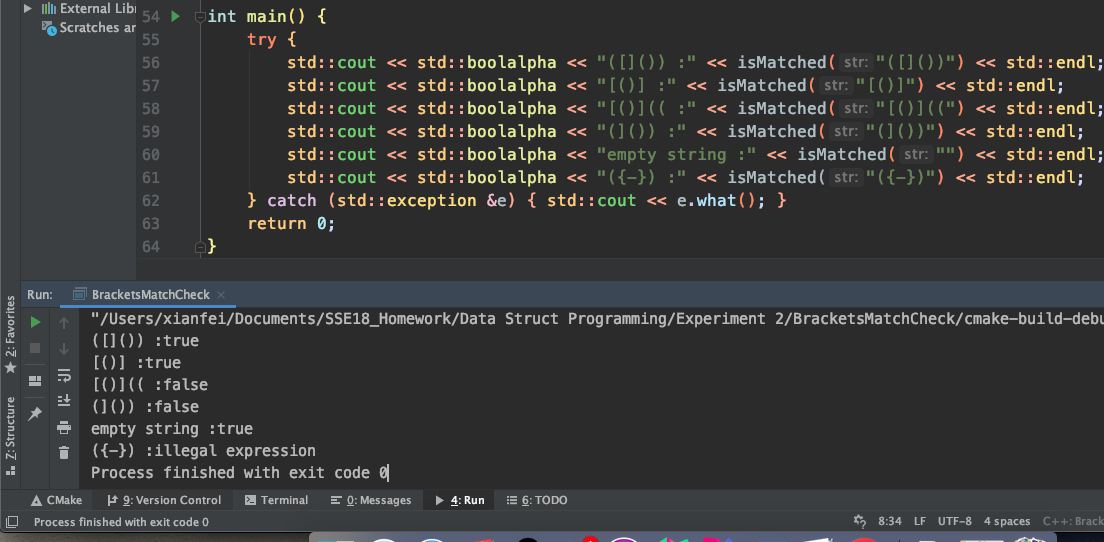
心得：此程序也可以用于其他的进制转换，进制转换使用堆栈十分方便

1. 括号匹配：

结构：一个堆栈

算法：遇到左括号压栈，遇到右括号出栈，判断出栈的左括号与遇到的右括号是否匹配，如果不匹配则返回false。

测试数据：包含能够匹配的、不能匹配的、空字符串以及非法表达式，测试结果如截图所示。



心得：括号匹配使用堆栈十分方便，遇到不属于题目的符号则抛出异常。

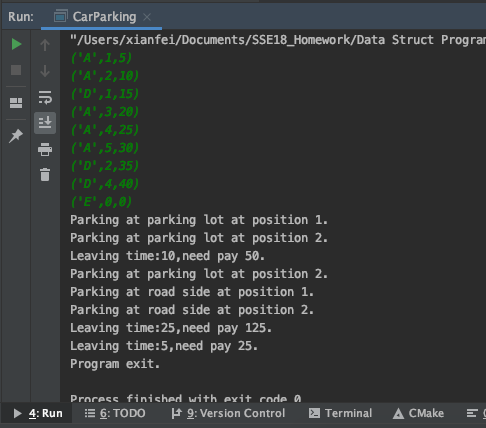
1. 停车场

结构：两个栈，一个栈为停车场，另一个栈为要出停车场车辆临时存放的栈。一个队列，为路边停车的地方。

算法：分别为进入和离开，进入先判断栈的大小是否小于n，如果不小于则放到路边，并记录车号和时间。离开时在栈中查找，查找时先把其他车辆搬到临时栈中，等到停车栈空时再把它们搬回来。并计算停车时间。

测试：使用题目中测试数据进行测试，假设单位时间停车费为5元。

截图：



心得：体会到了堆栈和队列在实际生活中的应用。