## 程序设计基础课程设计选做题目

## 1 基于堆栈的表达式求值

表达式求值是程序设计语言编译中的一个基本问题。它的实现就是对“栈”的典型应用。本文针对表达式求值使用的是最简单直观的算法“算符优先法”。

算术四则运算的运算规则是：先乘除，后加减；从左到右计算；先算括号内，再算括号外。

表达式组成：任何一个表达式都有操作数、运算符和界定符组成。操作数即可以是常量，也可以是被说明为变量或常量的标识符。运算符可以分为算术运算，关系运算和逻辑运算符。界定符有左右括号和结束符等。为了设计方便可以只考虑使用算术运算。

运算符优先级：对于两个相继出现的操作符有三种关系：大于、等于和小于。由此可以列出“+-\*/”之间的优先级。如下表：

+ - \* / ( ) #

+ > > < < < > >

- > > < < < > >

\* > > > > < > >

/ > > > > < > >

( < < < < < =

) > > > > > >

# < < < < < =

解释：加减乘除优先性都低于“（”但是高于“）”，由运算从左到右可知，当θ1=θ2 ，令θ1>θ2

为了算法简洁，在表达式的左边和右边虚设一个“#”，这一对“#”表示一个表达式求值完成。

“（”=“）”当一对括号相遇时表示括号内已运算完成。

“）”和“（”、“#”和“(”、“(”和“#”无法相继出现如果出现则表达式出现语法错误。

为实现优先算法，可以使用两个工作栈，一个是OPTR，用于寄存运算符，一个是OPND，用于寄存运算数和运算结果。

算法基本思路：

首先置操作数栈为空栈，表达式起始符为“#”为栈底元素。

依次读入表达式中的每个字符，若是操作数则进OPND栈，若是运算符则和OPTR栈的栈顶运算符比较优先权作相应操作，直至整个表达式求值完毕（OPTR栈顶元素和当前读入的字符均为“#”）。

## 2 基于队列的银行排队系统

利用队列的保存顺序特性，模拟售票口服务解决实际问题  
问题背景：  
一般来说，当我们去火车站买票时，服务窗口越多，队走的越快。但是，火车站的领导希望顾客满意，但又不希望雇佣过多的员工。那么应该开多少个窗口才是最合理的，能满足火车站领导的需求呢？（人数为100）  
假设模拟的服务窗口必须满足以下条件：  
（1）只排一队，并且先到的人先得到服务（这是一个队列）  
（2）平均每隔n秒就会来一位顾客  
（3）如果有空闲的窗口，在顾客抵达之时就会马上处理  
（4）从顾客来到窗口到处理完顾客请求，这个平均需要m秒  
请问当给定n和m的值时，模拟窗口服务。

## 3 基于哈希表的单词存储、查询

针对整个单词设计哈希函数，设计一下解决冲突的策略，能存储100个单词。

## 4 基于双向链表的汉字备选表

该功能主要用于输入法中的提示，提示的字是按照其的出现频率先后顺序排列的。

（1）首先要建立一个同音词表，里面存储着发音，和对应的汉字

如：

che 车 扯 撤 彻 ……

she 设 射 蛇 社 ……

……

（2）每次在搜索时，用户的输入会动态改变链表中节点的顺序，下次再输入时显示的顺序会发生变化