# 实验五 逆波兰式的产生及计算

## 实验目的

将简单算术表达式转换为用逆波兰式，并计算用逆波兰式的值，本实验只需要支持正数即可。

## 实验内容

简单表达式转换成逆波兰式的原则如下：设置一个输出队列，设置一个符号栈，为了处理方便，可以在输入简单表达式的最后加一个#，然后遍历表达式中的每个单元

1.是数字，则放入队列中，回到第一步，不是则下一步。

2是左括号，则放入符号栈中，回到第一步，不是则下一步。

3是右括号，则不断弹出符号栈顶元素，放入队列，直到弹出左括号，左括号不加入队列，回到第一步，不是则下一步。

4是操作符(+-\*/())，如符号栈顶为空或栈顶为左括号或栈顶操作符优先级比当前操作符优先级低，则直接将当前操作符压入符号栈，否则将栈顶元素出栈，添加到队列，直到不满足条件，回到第一步。

遍历结束后，将符号栈中剩余的符号都逐个弹出到输出队列中。此时输出队列中就是所要的逆波兰式。逆波兰式是不需要保存括号的。

逆波兰式计算的原则如下：设置一个输出栈，从左到右遍历逆波兰式

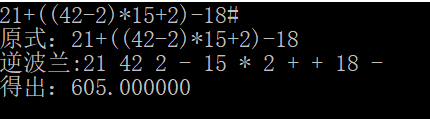
1. 如果是数字，则压入输出栈
2. 如果是(+-\*/)，则从输出栈弹出两个操作数，和操作符计算后，将结果压入输出栈

遍历完成后输出栈中的数值就是计算的数值。

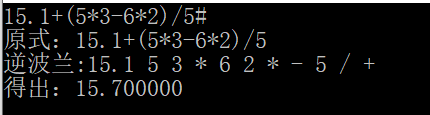
计算值用double类型存储和输出。

测试案例：

输入表达式21+((42-2)\*15+2)-18#进行计算，结果如下：



输入表达式15.1+(5\*3-6\*2)/5#进行计算，结果如下：



## 实验要求

* 1. ● 使用C/C++完成编程；
  2. ● 使用实验所提供的模板撰写实验报告，不要增删，要求内容详实，针对本实验的具体设计、思路描述，关键代码片段及其注释，及实验输入输出屏幕截图；
  3. ● 在截止日期前将代码、实验报告提交

周二晚上：<http://xzc.cn/rR0PF9Rf5O>

周三下午：<http://xzc.cn/o7cWX3z384>

周三晚上：<http://xzc.cn/fWufRKx516>

周四下午：http://xzc.cn/T0gz1GkKgM