

实验四：逆波兰表示法

实验环境	PC 机+Win 2003+VS2017	严禁抄袭 仅供参考 Blog:zhangshier.vip
<p>一. 实验项目要求</p> <p>将非后缀式用来表示的算术表达式转换为用逆波兰式来表示的算术表达式，并计算用逆波兰式来表示的算术表达式的值。</p> <p>输出的格式如下：</p> <p>(1)逆波兰式的生成及计算程序，编制人：姓名，学号，班级</p> <p>(2)输入一以#结束的中缀表达式(包括+—*/ () 数字#)：在此位置输入符号串如(28+68)*2#</p> <p>(3)逆波兰式为：28&68+2*</p> <p>(4)逆波兰式 28&68+2*计算结果为 192</p> <p>备注：</p> <p>(1)在生成的逆波兰式中如果两个数相连则用&分隔，如 28 和 68，中间用&分隔；</p> <p>(2)在此位置输入符号串为用户自行输入的符号串。</p> <p>注意：</p> <p>1.表达式中允许使用运算符 (+-*/)、分割符 (括号)、数字，结束符#；</p> <p>2.如果遇到错误的表达式，应输出错误提示信息 (该信息越详细越好)；</p>		
<p>二. 理论分析或算法分析 (含实验项目要求的分析、数学或逻辑推导等)</p> <p>问题分析：逆波兰式是一种表示表达式的方法，这种表示法把操作数写在前面，把运算符写在后面，又称为后缀式。</p> <p>算法</p> <p>① 如果表达式 E 是变量或常数，则 E 的后缀表示即 E 本身。</p> <p>② 如果 E 为 $E_1 \text{ op } E_2$ 形式，则它的后缀表示为 $E_1' E_2' \text{ op}$；其中 op 是二元运算符，而 E_1'、E_2' 分别是 E_1 和 E_2 的后缀表示，若 op 为一元运算符，则视 E_1 和 E_1' 为空。</p> <p>③ 如果 E 为 (E_1) 形式，则 E_1 的后缀表示即为 E 的后缀表示。</p> <p>算法流程图</p> <pre> graph TD Start([Start]) --> Scan[从左到右扫描中缀表达式] Scan --> IsOp{运算符} IsOp -- Y --> Output[输出] IsOp -- N --> IsOpL{左括号 (} IsOpL -- Y --> PushL[入栈] IsOpL -- N --> IsOpR{右括号)} IsOpR -- Y --> IsStackEmpty{栈为空} IsStackEmpty -- Y --> PushOp[入栈] IsStackEmpty -- N --> ComparePrecedence{当前运算符与栈顶运算符比较优先级} ComparePrecedence -- 当前大 --> PushOp ComparePrecedence -- 当前小 --> PopStack[退栈推出] ComparePrecedence -- 相等 --> PopStack IsOpR -- N --> IsStackEmpty IsOpR -- Y --> IsStackEmpty IsStackEmpty -- Y --> PopStack IsStackEmpty -- N --> IsStackEmpty IsStackEmpty -- Y --> Error[Error] IsStackEmpty -- N --> End([End]) </pre>		

三. 实现方法（含实现思路、程序流程图、实验电路图和源程序列表等）
调试程序，在控制台生成并计算 $(28+68)*2\#$

四. 实验结果分析 (含执行结果验证、输出显示信息、图形、调试过程中所遇的问题及处理方法等, 如果有引用的参考文献, 安排在本节最后列出)



```
Microsoft Visual Studio 调试控制台
逆波兰式的生成及计算程序, 编制人: 张帆, 192062116, 1920542班
输入一以#结束的中缀表达式(包括 + - * / ( ) 数字 # ): (28+68)*2#
逆波兰式为: 28&68&+2&*
逆波兰式28&68&+2&*计算结果:192
top = 0
D:\太工\大三第二学期\Compilation.principle\Experiment\x64\Debug\Experiment.exe (进程 21140) 已退出,
返回代码为: 0。 编制人: 张帆, 192062116, 1920542班
若要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。
```