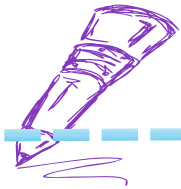
表达式计算问题



前缀表达式—波兰式（Polish Notation, PN）

- + * 31 -5 22 70

中缀表达式

- 31*（5-22）+70

后缀表达式—逆波兰式（Reverse Polish Notation, RPN）

- 31 5 22 - * 70 +

表达式计算问题—后缀表达式计算

- 算法过程：
- 顺序扫描后缀表达式，
  - 如果是操作数，则压入栈中；
  - 如果是操作符，则从栈中弹出两个操作数进行计算，把结果再压入栈
- 扫描结束时，栈顶元素就是表达式的值

步骤	待处理的后缀表达式（头----------尾）	栈（顶----------底）	进行的计算
1	31 5 22 - * 70 +		
2	5 22 - * 70 +	31	
3	22 - * 70 +	5 31	
4	- * 70 +	22 5 31	
5	* 70 +	- 17 31	5-22 = -17
6	70 +	-527	31*（-17）= -527
7	+	70 -527	
8		-457	-527+70 = -457
9		-457	计算结束

步骤	待处理的后缀表达式 （头----------尾）	栈 （顶----------底）	进行的计算
1	6 2 / 3 - 4 2 * +		
2			
3			
4			
5			
6			
7			
8			
9			

中缀表达式转换为后缀表达式

中缀表达式

- 31*（5-22）+70

后缀表达式

- 31 5 22 - * 70 +

	+	-	*	/	(	)	#
+	>	>	<	<	<	>	>
-	>	>	<	<	<	>	>
*	>	>	>	>	<	>	>
/	>	>	>	>	<	>	>
(	<	<	<	<	<	=	>
)	>	>	>	>	=	>	>
#	<	<	<	<	<	<	=

操作符的优先级

- **乘除高于加减，（）的优先级最低**
- **同一优先级的，按照先后规定优先级**
- **（）里面的是同一层次需要一起计算**

中缀表达式转换为后缀表达式

- 算法过程如下：
- 从左至右依次扫描中缀表达式
  - 如果是操作数，则直接输出
  - 如果是“(”，则入栈中
  - 如果是“)” 则弹出栈顶元素并放入后缀表达式中，反复执行直到栈顶元素为“(”时为止，表明这一层括号内的操作处理完毕
  - 如果是操作符，则将该操作符和操作符栈顶元素比较
    - 如果**大于**栈顶元素的优先级时，则将它压入栈中
    - 如果**小于**栈顶元素的优先级时，则取出栈顶元素放入后缀表达式，并弹出该栈顶元素，反复执行直到栈顶元素的优先级小于当前操作符的优先级
- 重复上述步骤直到遇到中缀表达式的结束‘#’，弹出栈中的所有元素并放入后缀表达式中，算法结束

步骤	待处理的中缀表达式 （头----------尾 ）	栈 （顶-------底）	已经输出的部分后缀表达式 （头----------尾）
1	31*（5-22）+70		
2	*（5-22）+70		31
3	（5-22）+70	*	31
4	5-22）+70	（*	31
5	-22）+70	（*	31 5
6	22）+70	-（*	31 5
7	）+70	-（*	31 5 22
8	+70	*	31 5 22 -
9	70	+	31 5 22 - *
10		+	31 5 22 - * 70
11			31 5 22 - * 70 +

步骤	待处理的中缀表达式 （头----------尾 ）	栈 （顶-------底）	已经输出的部分后缀表达式 （头----------尾）
1	a*（b + c）*d		
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			