

递归算法

- 1. 求一个数组全部元素的和值.
- 2. 以递减的顺序输出一个递增有序顺序表中的元素值.
- 3. 在数组中查找一个元素。
- **Sum.cpp**

去掉算术表达式的括号

问题：

将一个代数的算术表达式的括号去掉。如，表达式： $a-(b-c)\#$ ，去掉括号之后的表达式为 $a-b+c$ ；又如表达式 $(a-b-c)*(c-d)\#$ ，则结果为： $a*c-a*d-b*c+b*d-c*c+c*d$ 。

为了处理的简单，我们做以下的规定：

- (1) 运算符：+、-、*、/、()且无单目运算符
- (2) 操作数：小写的英文字母
- (3) 表达式输入的结束符为：#

测试数据：

- 1.输入： $a-(b-c)\#$ 输出： $a-b+c$
- 2.输入： $a/(b/c)\#$ 输出： $a/b*c$
- 3.输入： $(a+b)*(c*d)\#$ 输出： $a*c*d+b*c*d$
- 4.输入： $(a+b)*(c+d)\#$ 输出： $a*c+a*d+b*c+b*d$
- 5.输入： $a/(b-c)\#$ 输出： $a/(b-c)$
- 6.输入： $a/(b+c)\#$ 输出： $a/(b+c)$
- 7.输入： $(a-b-c)*(c-d)\#$ 输出： $a*c-a*d-b*c+b*d-c*c+c*d$
- 8.输入： $a*(b-c)\#$ 输出： $a*b-a*c$
- 9.输入： $a*(b+c)\#$ 输出： $a*b+a*c$
- 10.输入： $a-((b-c)-f)\#$ 输出： $a-b+c+f$
11. $a/((b/c)/e)\#$ 输出： $a/b*c*e$

算法分析 `changeexp.cpp`

- **1. 思路：**
- (1) 先为输入的表达式建立起相应的二叉树；
- (2) 按需求置换二叉树中的分支或结点；
- (3) 以中序遍历序列输出二叉树即为结果

(1) 先为输入的表达式建立起相应的二叉树；
可以根据编译原理的方法来建立起相应的二叉树。

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid i$

由于文法规则 E 和 T 含有左递归，因此可以用 {} 方法消除左递归

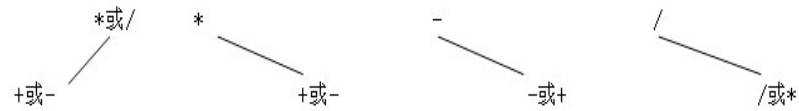
$E \rightarrow T \{ +T \mid -T \}$

$T \rightarrow F \{ *F \mid /F \}$

$F \rightarrow (E) \mid i$

因此用递归下降方法可以写出相应的递归函数。//函数中少了出错的处理部分

(2) 按需求置换二叉树中的分支或结点；



因此，我们可以后序遍历算法为基础进行适当的改进即可。

由于经过置换后的二叉树还有可能出现以上几种情况，因此，我们需要反复利用这个函数进行置换，直到这棵二叉树再也没有进行过置换为止。

(3) 以中序遍历序列输出二叉树即为结果

要注意考虑一些表达式的括号是不能消除的，如， $a/(b-c)$ 和 $a/(b+c)$ 等情况的二叉树输出

