

lab2

18340146 计算机科学与技术 宋渝杰

算法描述：

中缀转表达式树我采用“递归建树”的方式，具体步骤如下：

1. 对中缀字符串，先找到其优先级最低的运算符（先寻找不在括号内的最右端加减符号，如果没有的话就寻找不在括号内的最右端乘除符号）
2. 如果找到该运算符，该运算符为根节点符号，运算符左边的字符串为左子树，右边的字符串为右子树；如果找不到，则该节点为叶节点，结束这部分递归
3. 对左右字符串进行递归建树


之后是树的输出：可以用层次遍历的方式，使用一个队列，先将根节点入队，之后在队列不为空的情况下，如果队头节点不为空节点：入队队头节点的左右子节点（注意：空节点也入队），之后队头节点出队并输出；如果队头节点为空节点，则输出 "x"（意为空节点 null）并出队。

输出树的结构：某一层的第 $2n-1$ 、第 $2n$ 个节点为上一层第 n 个不为 "x" 的节点的左右子节点（ n 从 1 开始计数）。这也是上文为什么要入队空节点以及输出 "x" 的意义

验证正确性：表达式树有一个特性：后序遍历结果等于后缀表达式，因此将树进行后序遍历，并与 lab1 的后缀表达式相对比即可验证准确性


测试结果：

标准样例：

 C:\Users\Song\Desktop\S & W\大三下\编译原理\lab2.exe

```
Input:
3+(4+5/(2-1))
Print Tree By Level:
+
3  +
x  x  4  /
x  x  5  -
x  x  2  1
x  x  x  x
Print Tree By Pos:
3 4 5 2 1 - / + +
Input:
21+42-30/(5+5)*(4-2)
Print Tree By Level:
-
+  *
21 42 / -
x  x  x  x  30 + 4 2
x  x  5  5  x  x  x  x
x  x  x  x
Print Tree By Pos:
21 42 + 30 5 5 + / 4 2 - * -
```

额外样例（包括小数、正负号等）：

 C:\Users\Song\Desktop\S & W\大三下\编译原理\lab2.exe

```
Input:
1314+25.5*12
Print Tree By Level:
+
1314 *
x  x  25.5 12
x  x  x  x
Print Tree By Pos:
1314 25.5 12 * +
Input:
-2*(+3)
Print Tree By Level:
*
-2 3
x  x  x  x
Print Tree By Pos:
-2 3 *
```