实验心得

1 非递归的中序遍历需要借助stack s来实现，模拟递归调用 ，总的循环边界是当前节点不为空或者stack不空，在当前节点p非空时候，将p入栈s, p的左子树赋给p,保证左子树都能入栈 ，当p为空，也就是左子树最左边访问到了，这时候在栈非空的时候就取栈顶给p，输入p，出栈，这时候最底层的最左边节点访问了，将p的右子树赋给p，重复之前过程

2 在处理语句的过程中，现在字符串中搜索空格来分隔语句。经过分割之后的每个语句中以括号为标识来判断父节点。在现有的节点中搜索到现有语句的节点之后，找到以后判断节点下是否有左子树，如果有即写入，如果没有就写入右子树。

3. 按照中序遍历的方法转换到二叉树中。先要确定一个根节点，然后递归创建整个树。即，在函数输入时写入起始和结束的位置，起始和结束位置的整个数组中间的一位作为根节点，左右各为新的数组继续执行判断。

4.首先用递归计算整个树的高度。按层次输出数列，即从根节点开始，每向下一层就将level-1，当level==0时就达到该层，即可输出。依次遍历整个树找到所有该层的节点。

按层次完整的输出树就是遍历树的每一层。

因为最后才写了第四题才发现可以按层次输出。之前的输出采用的是树形输出但是结果为树形图旋转了90°，级每到每个纵列表示一层，每一个结点（即使为空）占一行。

横向树状输出暂时自己不太会。