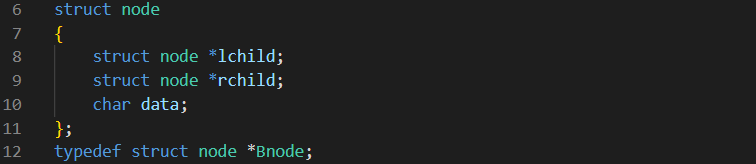
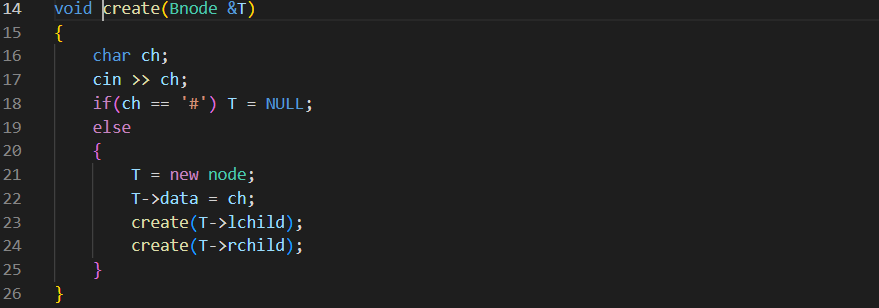
**作业3 树形结构及其应用**

**1. 二叉树的动态（或者静态）二叉链表存储结构（左右链表示）的建立**

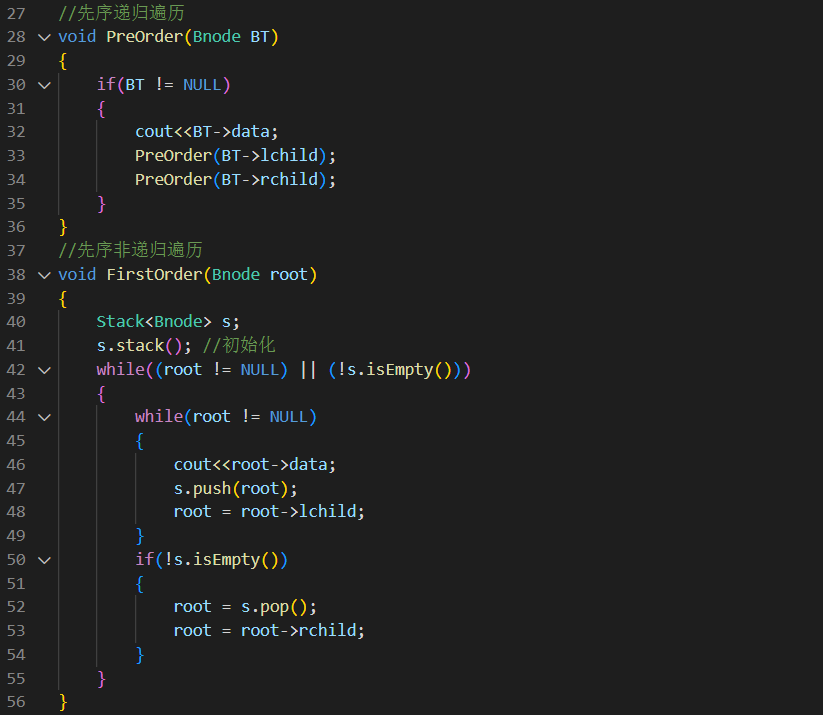
定义链表结构如下：

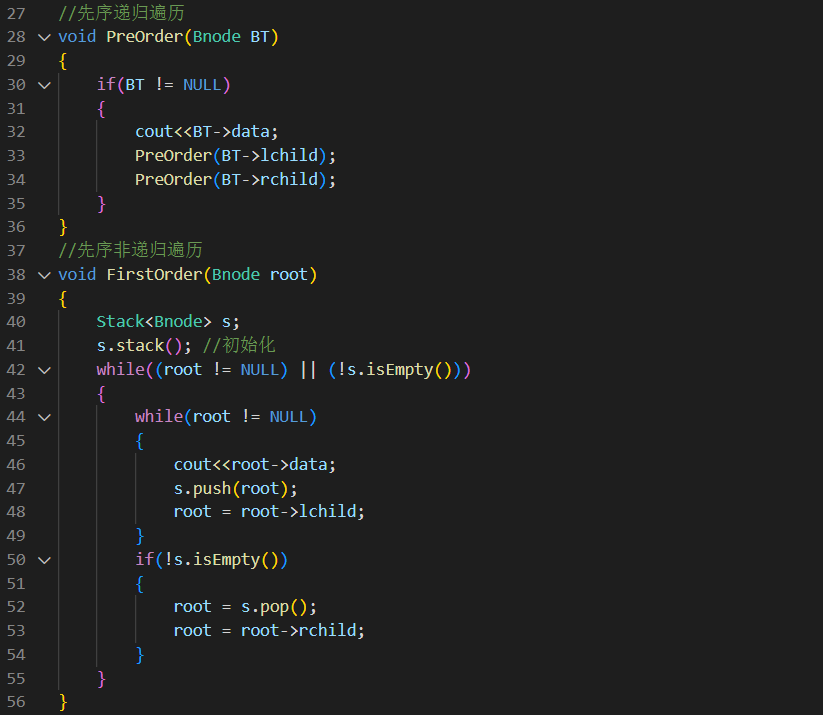
结构体node中包含指向左右儿子的指针lchild与rchild以及数据域data。

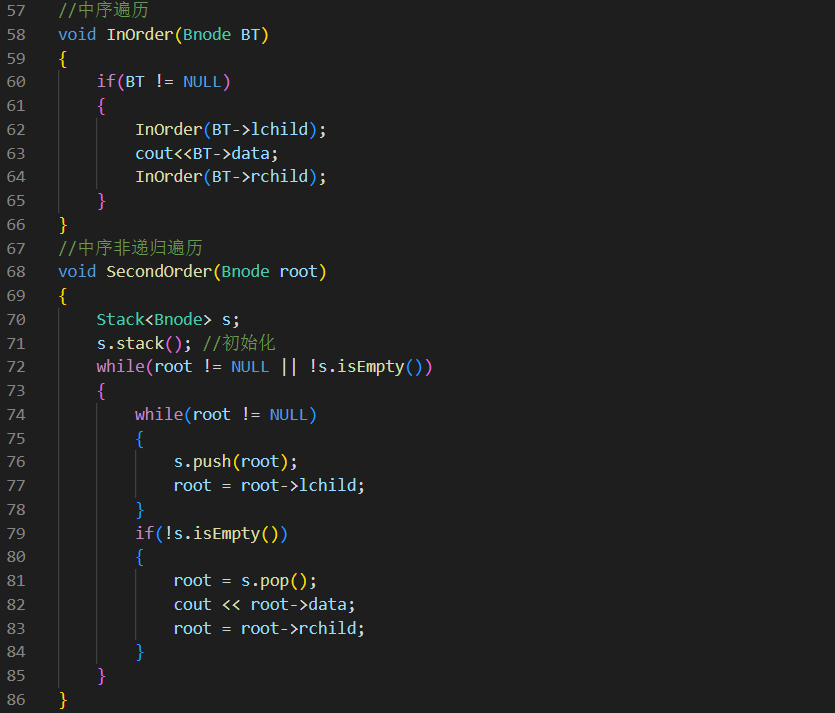
二叉树结构的建立按照先序序列创建，以‘#’代表空（例如AB##CD###），建立过程采用递归的思想，代码如下：

**2. 采用二叉树的上述二叉链表存储结构，编写程序实现二叉树的先序、中序和后序遍历的递归和非递归算法以及层序遍历算法**

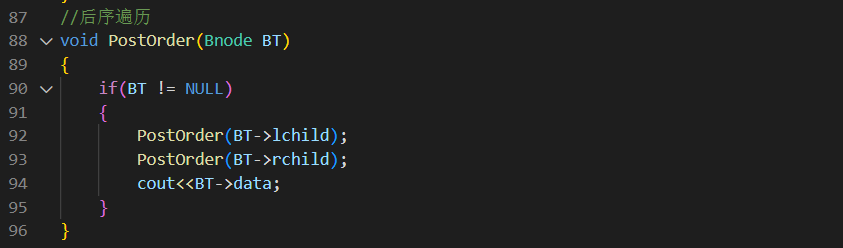
A．先序的递归与非递归遍历

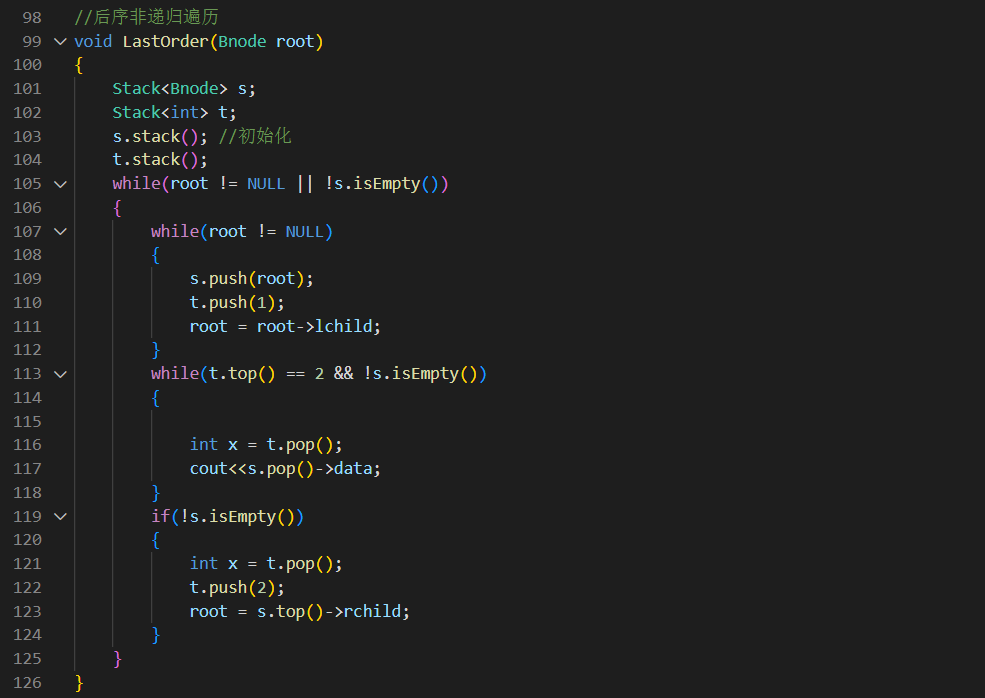
****

****

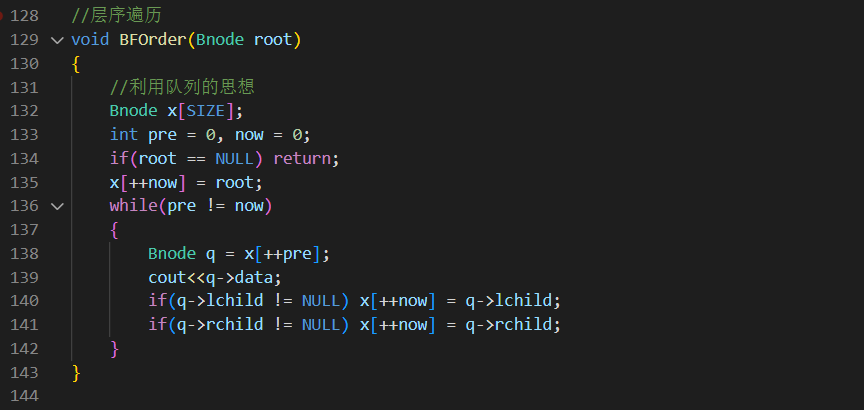
B．中序递归与非递归遍历

C．后序递归与非递归遍历

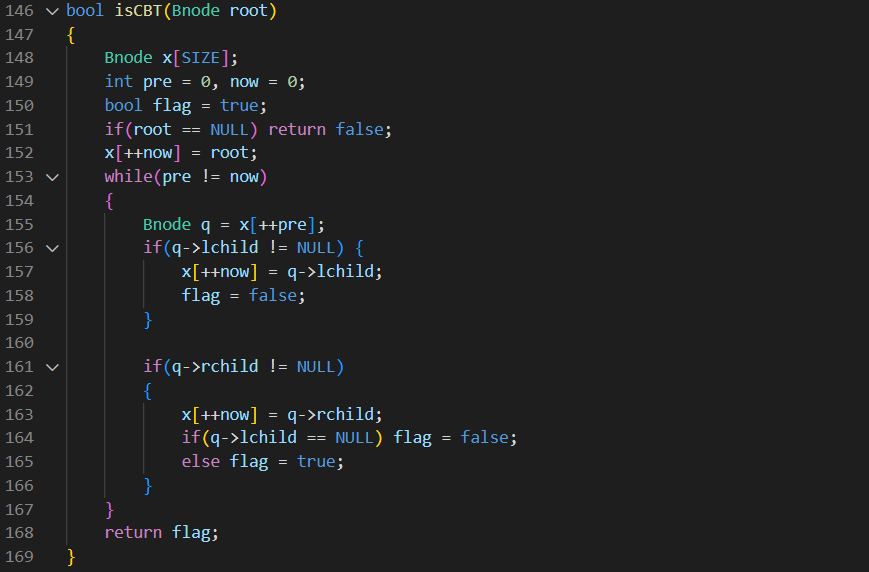
****

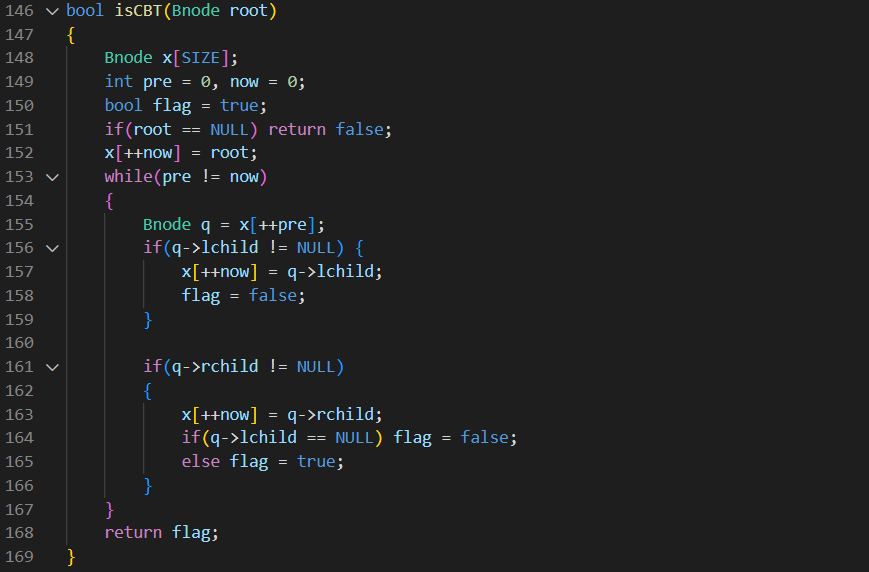
****

D．层序遍历（基于队列思想）



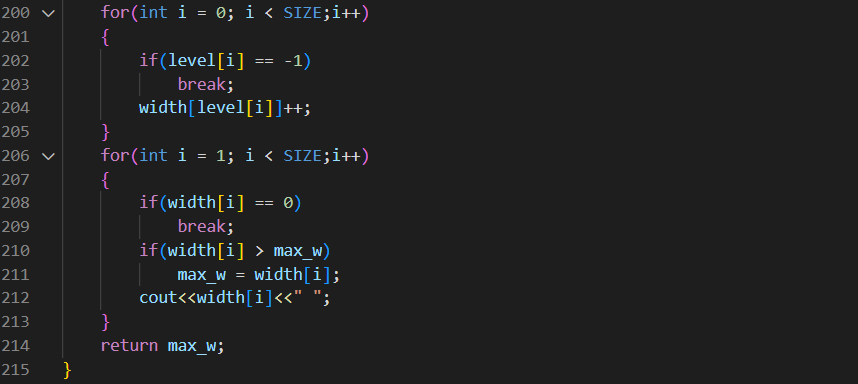
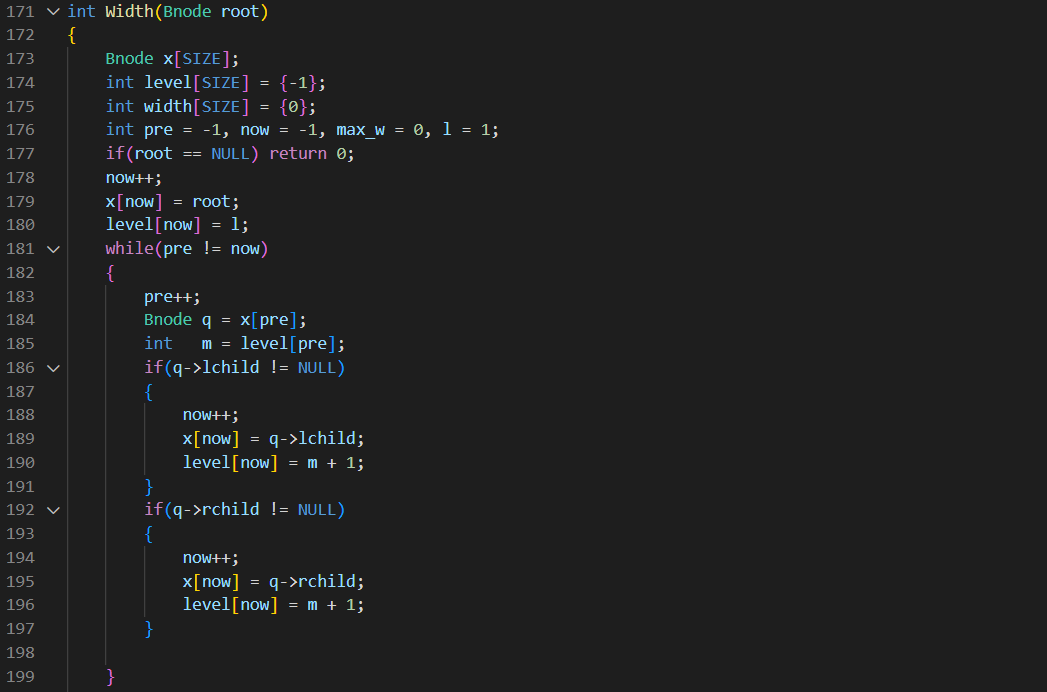
**3. 判断任意一棵二叉树是否为完全二叉树**

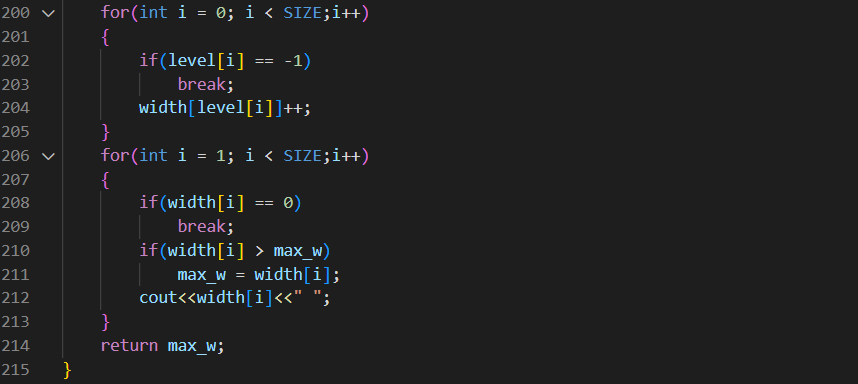
基于层序遍历，引入bool型变量flag，初始值为true，遍历过程中当左儿子不为空时，flag变为false，当右子树存在且左子树存在时，flag又重新变为true，据此判断此二叉树是否为完全二叉树。

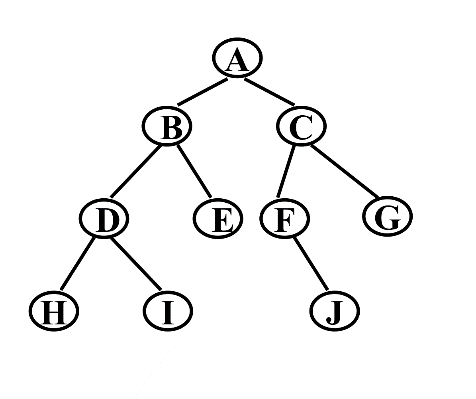


**4. 计算任意一棵二叉树的宽度**

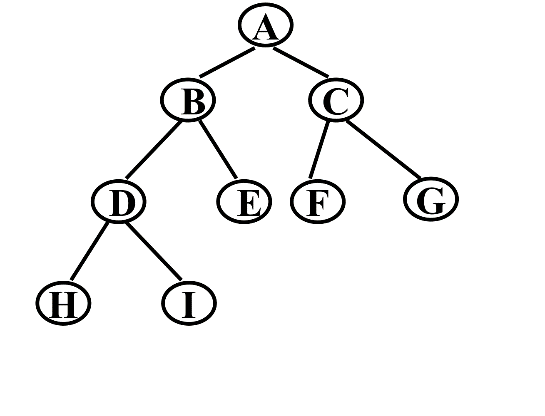
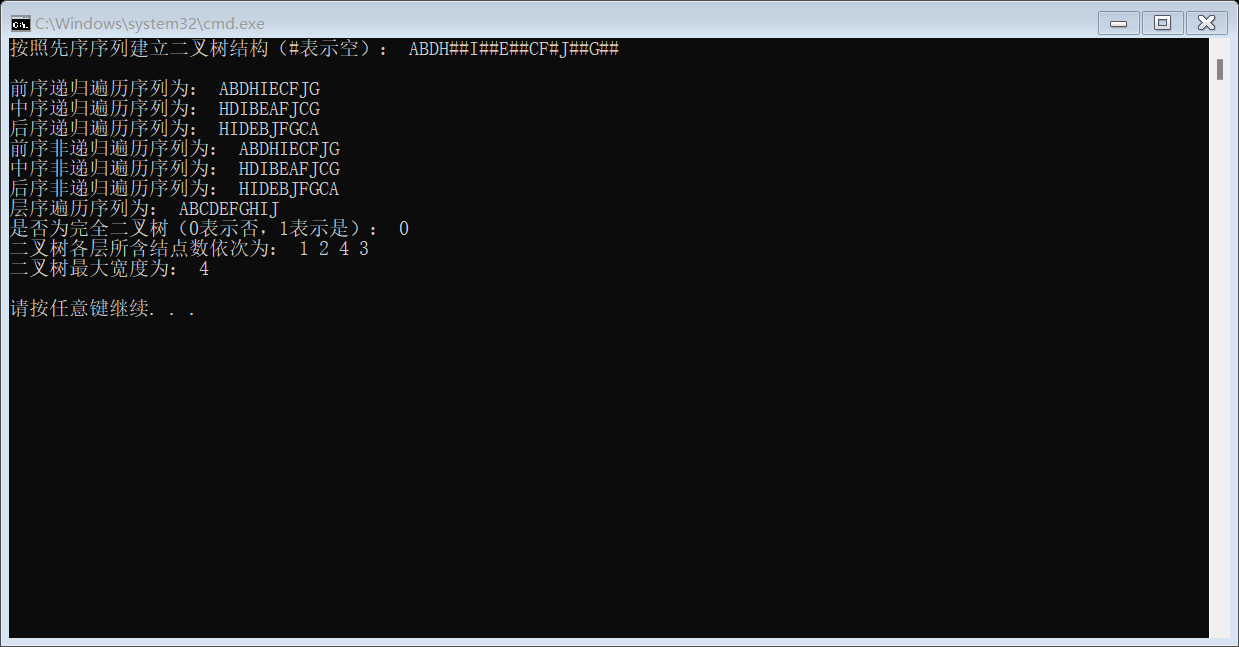
在层序遍历的过程中，对加入队列的每个结点加入当前所在层数的参数level，遍历结束后，对参数level的不同取值进行统计，即为各层的结点数，取最大值即为此二叉树的宽度。



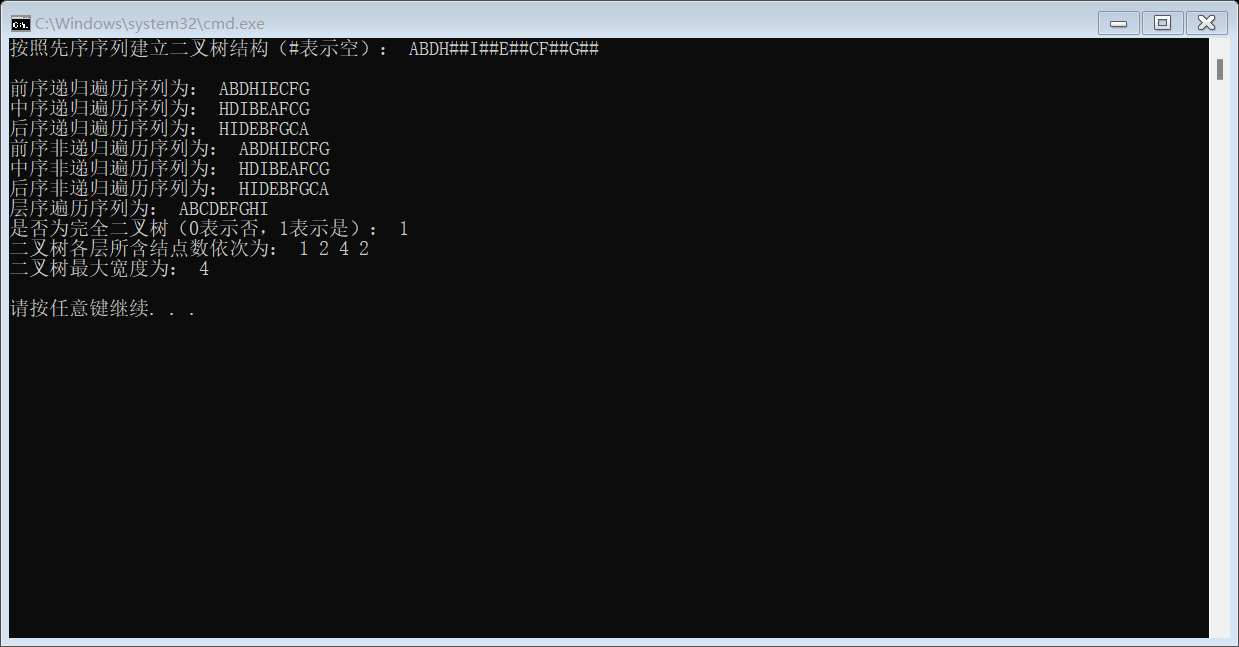


**5.测试结果**

**1.** **ABDH##I##E##CF#J##G##（如右图所示）：**

****

**2. ABDH##I##E##CF##G## （如右图所示）：**

****