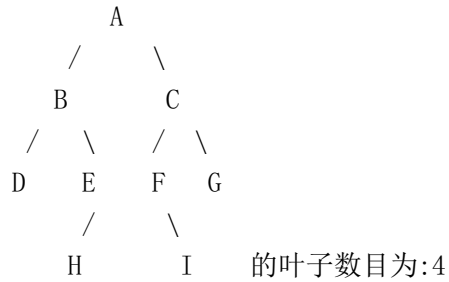
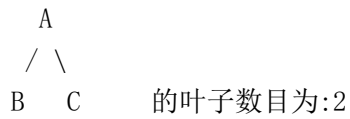


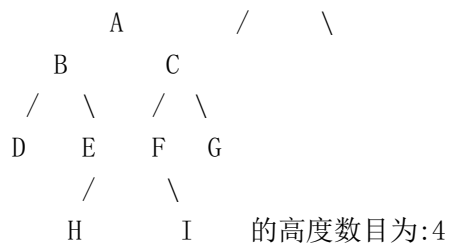
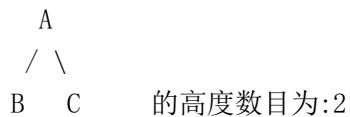
## HW6

任务 1: 编写递归函数计算二叉树的叶子节点数目, 如



提示: 叶子的数目 = 左子树的叶子的数目 + 右子树的叶子的数目

任务 2: 编写递归函数计算二叉树的高度



提示: 树高 = max(左子树高度, 右子树高度) + 1

任务 3: 利用

-----

1) 二叉树创建函数(输入带有空指针标记'\$'的先序遍历序列, 输出不带头节点的二叉树) // 通过字符串(用户键盘输入)创建二叉树的函数

```
void MakeCharTree(TreeNode **bt) {
    char ch;
    ch = getchar();
    if (ch == '$') *bt = NULL;
    else {
        *bt = (TreeNode*)malloc(sizeof(TreeNode));
        (*bt)->data = ch;
    }
}
```

```

        MakeCharTree(&((*bt)->leftChild));
        MakeCharTree(&((*bt)->rightChild));
    } // else ends
} // function ends

```

- 2) 递归版本的后序遍历函数 RecPostOrder 3) 栈数据结构及其操作的概念 实现，当用户输入:++5\$\$\$2\$\$+6\$\$\$时，输出表达式  $(5+3)*2+(6+3)$  的计算结果 提示 1: 用 1) 创建二叉树，然后把 2) 的后序遍历函数进行改写，同时观察递归现象和
- 3) 栈概念的关系，最终完成计算。提示 2: 任务 3 的计算过程没有用到栈的代码，只是用了概念，因为递归后序遍历隐含了栈，所以不需要再创建栈的结构。

**HW6作业截止时间：5月1号**