数据结构 第二次测试.007行动

新一轮007行动开始了，要想成为一名合格的特工怎能不AC。特工要开始行动了，第一个任务是嫌疑人追踪。

**1.追踪**

**Description**

特工拿到了一个二叉树地图，嫌疑人已经逃离现场，根据时间估算嫌疑人只能前往距离根节点长度为n的位置。现在，你能帮他找到嫌疑人可能逃跑的路径吗？

特工拿到的地图是一颗二叉树，估计的距离n是一个整数，打印出二叉树中结点值的和为输入整数n的所有路径。从树的根节点开始往下一直到叶结点所经过的结点形成一条路径。二叉树结点的定义如下：

struct BinaryTreeNode{

int m\_nValue;

BinaryTreeNode\* m\_pLeft;

BinaryTreeNode\* m\_pRight;

}

Input

10

/ \

5 12

/ \

4 7

**查找22**

Output:[10,5,7][10,12]

**Input**

每组测试案例有两行。

第一行是二叉树的层序描述，空值用null表示

第二行为查找的目标值

**Output**

二叉树中结点值的和为输入整数的所有路径，不同路径用[]分隔。不存在任何路径时输出-1。

**Sample Input**

10 5 12 4 7 null null

22

**Sample Output**

[10,5,7][10,12]

**2.密码**

**Description**

007行动又开始了，特工遇到的一个难题就是破解密码。密码分两部分，前半部分是在多行字符串中找出第一个只出现一次的字符串，后半部分是根据前半部分找出的字符串在“神秘字符串”中判断是否存在。

**Input**

输入数据有q+2行。

每组测试数据，第一行有一个整数q，表示字符串的个数，接下来的q行里，每行是一个字符串（长度低于100）。末尾行是“神秘字符串”

**Output**

前半部分输出所有字符串中第一个只出现一次的字符串，如不存在，输出-1。若前半部分密码不存在，后半部分直接输出-1；若前半部分密码存在，则判断在“神秘字符串”是否存在，若是则返回第一次匹配的下标，不是输出-1（下标从0开始）。

**Sample Input**

6

aba

bc

aba

ef

d

ef

aaabc

**Sample Output**

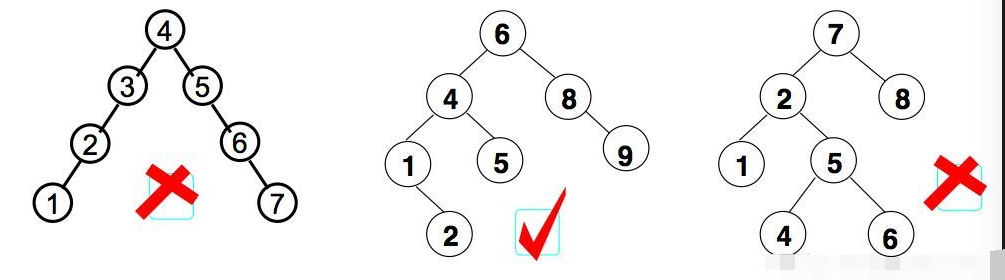
bc3

**3. 特工组织**

**Description**

作为一个特工BOSS，想要对全体特工进行有效管理，简单的办法是调整特工组织架构使得尽量扁平，又得符合特工组织一个领导最多两个直接下属的传统，组织架构自然而然变成了一颗二叉树。树的每一层对应职位，大boss在第一层，第二级职位的boss在第二层，执行特工在叶子节点。

1.对于执行特工，如果跟其他执行特工组队并且职位相差大于一就会不开心，请你判断当前的特工组织架构是否会导致某些执行特工不开心？



（不开心） （开心） （不开心）

2.传达给任意两个执行特工消息最好的办法就是通知他们的公共的特工，对于给定的两个执行特工，请你找出需要通知的特工。例如：中间图中，2和5需要通知4，5和9需要通知6。

**Input**

每组测试案例有两行。

第一行按先序遍历顺序输入二叉树的各个结点值（特工代号，各不相同），#表示空节点，各个值之间空格区分。

第二行有两个执行特工的代号，请你求出需要通知的特工代号

**Output**

输出有两行：

第一行，各个执行特工都开心的组织架构输出YES 否则输出 NO

第二行，对于开心的组织架构，输出要通知的特工代号；对于不开心的组织架构直接输出-1

**Sample Input**

6 4 1 # 2 # # 5 # # 8 # 9 # #

2 9

**Sample Output**

YES

6