1、以二叉链表作存储结构，建立一棵二叉树， 输出该二叉树的中序遍历序列(递归法)，层序遍历序列，并求二叉树的深度。

考察：二叉树建树、二叉树中序遍历(递归法)、层次遍历、二叉树深度，叶子结点判断(先序遍历)

二叉链表的类型描述：

typedef char ElemType;

typedef struct BiNode

{ ElemType data;

struct BiNode \*lchild,\*rchild;

}BiNode,\*BiTree;

输入格式:

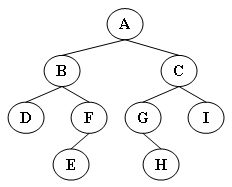
输入一个二叉树的**先序**序列，孩子为空的位置以#替代。

输出格式:

第一行输出该二叉树的**中序**遍历序列。输出的遍历序列中字符间均无间隔。第二行输出二叉树的**层序**遍历序列，第三行输出二叉树的深度。第四行输出二叉树所有叶子结点(按从左到右顺序)

具体格式参看输出样例。

对于下图中给出的二叉树：



输入样例:

ABD##FE###CG#H##I##

输出样例:

DBEFAGHCI

ABCDFGIEH

4

DEHI

1. 以二叉链表作存储结构，建立一棵二叉树， 输出该二叉树的中序遍历序列(非递归法)，求解指定点的父亲和兄弟结点，输出该二叉树删除某点后的中序遍历序列

考察：二叉树建树、二叉树中序遍历(非递归)、二叉树删除，结点父结点求解，结点兄弟结点求解

二叉链表的类型描述：

typedef char ElemType;

typedef struct BiNode

{ ElemType data;

struct BiNode \*lchild,\*rchild;

}BiNode,\*BiTree;

输入格式:

第一行输入一个二叉树的**先序**序列，孩子为空的位置以#替代

第二行输入某个已知字符P

第三行输入待删除字符Q

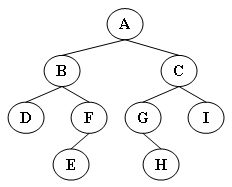
输出格式:

第一行输出该二叉树的**中序**遍历序列(非递归法)。输出的遍历序列中字符间均无间隔，第二行输入P的父结点，第三行输出P的兄弟结点，第四行输出删除Q后剩余二叉树的中序遍历序列

注：

1. **删除指删除该点及其所有子树**
2. **若结点无父结点或兄弟结点则输出提示信息**  
   具体格式参看输出样例。

对于下图中给出的二叉树：



输入样例:

ABD##FE###CG#H##I##

D

F

输出样例:

DBEFAGHCI

B

F

DBAGHCI