**二叉树的遍历及应用**

**一、实验目的**

1.理解二叉树的类型定义与性质。

2.掌握二叉树的二叉链表存储结构的表示和实现方法。

3.掌握二叉树遍历操作的算法实现。

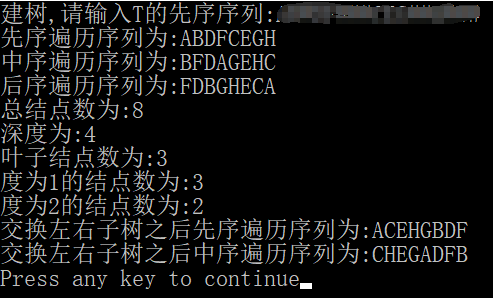
**二、实验内容**

1.利用二叉树的先序遍历序列建立二叉树。

2.实现二叉树的先序、中序和后序三种遍历操作。

3.求结点总数、深度、叶子结点数、度为1和度为2的结点数。

4.交换二叉树每个结点的左孩子和右孩子



**三、程序框架**

#include<stdio.h>

//定义二叉链表结点

typedef struct BiNode{

char data;

struct BiNode \*lchild,\*rchild; //左右孩子指针

}BiNode,\*BiTree;

//先序遍历的顺序建立二叉树

void CreateTree(BiTree &T){

char ch;

scanf("%c",&ch);

if(ch=='#')

T=NULL; //递归结束，建空树

else{

}

}

//先序遍历二叉树T,输出序列

void PreOrder(BiTree T)

{

}

//中序遍历二叉树T,输出序列

void InOrder(BiTree T)

{

}

//后序遍历二叉树T,输出序列

void PostOrder(BiTree T)

{

}

//求二叉树的深度

int Depth(BiTree &T)

{

}

//求二叉树的结点总数

int NodeCount(BiTree T)

{

}

//求二叉树的叶子结点数

int LeafCount(BiTree T)

{

}

//求二叉树度为1的结点数

int Du1(BiTree T)

{

}

//求二叉树度为2的结点数

int Du2(BiTree T)

{

}

//交换二叉树每个结点的左孩子和右孩子

void ChangeLR(BiTree &T)

{

}

void main()

{

BiTree T;

//自己补充

}