**二叉树的建立及遍历**

**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.应用二叉树的遍历操作来实现判断两棵二叉树是否相等的操作。

4.查找某个结点的双亲

**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)

{

}

//判断两颗树是否相等，相等返回1，不等返回0

int IsEqual(BiTree T1,BiTree T2)

{

}

//寻找孩子结点child的双亲结点，返回双亲结点的位置

BiNode\* FindParent(BiTree T,char child)

{

}

void main()

{

BiTree T1,T2;

//自己补充

}

T1

T2



