

# 实验三：树形结构及其应用

## 一. 实验目的

掌握树的存储结构和基本操作，重点巩固和体会二叉树的遍历及应用。

## 二. 实验内容

### 1、采用交互问答方式按照先序序列建立二叉树

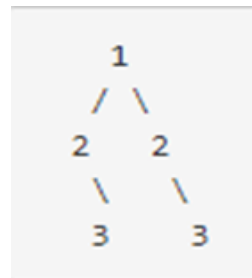
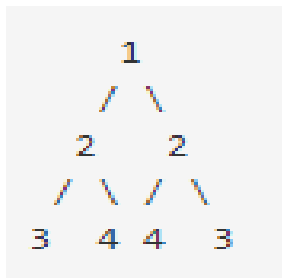
先输入根节点，每次都询问用户某个节点的左孩子是谁，右孩子是谁，其中字符' #'代表空节点。对建好的树进行前序遍历、中序遍历和后序遍历，并输出结果。

### 2、给定一棵二叉树的前序遍历和中序遍历结果

构造这棵二叉树，输出这棵二叉树的前、中、后序遍历结果。

### 3、判断实验内容 2 中构造的二叉树是不是镜像对称的

例如，下图左边的二叉树是镜像对称的，右边的二叉树则不是镜像对称的：



## 三. 实验要求

- 1、 提供模板代码 main.c，也可以自行编写代码；
- 2、 用链式存储结构实现建树；
- 3、 要求在实验课上完成实验内容的 1、2 部分，下课前 40 分钟开始检查；
- 4、 在课下完成全部的实验内容并撰写实验报告。

## 四. 参考代码

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node
{
    char ch;
    struct Node *lchild, *rchild;
} TreeNode, *BiTree; //定义树节点的结构体
void createBiTree(BiTree *p)//建立二叉树
{

}

void preOrderTraverse(BiTree p)//前序遍历
{

}

void inOrderTraverse(BiTree p)//中序遍历
{

}

void postOrderTraverse(BiTree p)//后续遍历
{

}

BiTree* BinaryTreeFromOrderings(char inorder[], char preorder[],int inorder_index,int
preorder_index, int length)
{
    //函数功能：根据前序与中序序列构造二叉树，并返回根节点
    //参数说明：inorder:中序序列字符数组，preorder:前序序列字符数组,inorder_index:记录中序
    序列索引,preorder_index:记录前序序列索引，length:序列长度

}

int isSymmetrical(BiTree root)
{
    //判断以 root 为根节点的二叉树是否是镜像对称的
}

void task1(){
    printf("start task (1) Create Tree in PreOrder\n");
    BiTree bt;
```

```

        createBiTree(&bt);
        printf("preOrderTraverse\n");
        preOrderTraverse(bt);
        printf("\n");
        printf("inOrderTraverse\n");
        inOrderTraverse(bt);
        printf("\n");
        printf("inOrderTraverse\n");
        postOrderTraverse(bt);
        printf("\n");
    }
void task2_3(){
    printf("start task (2) Input the preOrder and inOrder Sequence ,Then build the tree\n");
    char pr[100];
    char in[100];
    printf("please input the preorder sequence\n");

    printf("please input the inorder sequence\n");

    BiTree bt;
    bt=BinaryTreeFromOrderings(in, pr,0,0,strlen(pr));

    printf("preOrderTraverse\n");
    preOrderTraverse(bt);
    printf("\n");
    printf("inOrderTraverse\n");
    inOrderTraverse(bt);
    printf("\n");
    printf("inOrderTraverse\n");
    postOrderTraverse(bt);
    printf("\n");
    printf("start task (3):whether the tree is symmetrical?\n");
    if(isSymmetrical(bt)==1)
    {
        printf("It's mirror symmetrical");
    }
    else
    {
        printf("It's not mirror symmetrical");
    }
}

```

```

    }
}

int main()
{

    task1();
    task2_3();
    return 0;
}

```

## 五. 范例说明

### 1、采用交互问答方式按照先序序列建立二叉树

输入：

start task (1) Create Tree in PreOrder

1

Please enter the left child of 1

2

Please enter the left child of 2

#

Please enter the right child of 2

#

Please enter the right child of 1

3

Please enter the left child of 3

#

Please enter the right child of 3

#

输出：

Preoder: 123

Inorder: 213

Postorder: 231

### 2、由前序遍历和中序遍历结果建立二叉树，输出后序遍历结果，判断二叉树是否镜像对称

#### 测试用例 1

输入：

please input the preorder sequence

1234243

please input the inorder sequence

3241423

输出：

Preorder:1234243  
Inorder:3241423  
Postorder :3424321  
It's mirror symmetrical.

#### 测试用例 2

输入：  
please input the preorder sequence  
12323  
please input the inorder sequence  
23123

输出：  
Preorder:12323  
Inorder:23123  
Postorder :32321  
Not mirror symmetrical

#### 实验报告提交说明：

1. 电子版实验报告提交的截止时间为：2019 年 5 月 11 日晚上 10 点之前；
2. 电子版实验报告提交到该邮箱：hitsz\_ds\_2019@163.com；
3. 请把电子版实验报告及源代码打包成一个压缩包，命名格式如下：
  - a) 实验报告：“学号\_姓名\_实验 3”
  - b) 压缩包：“学号\_姓名\_实验 3”
  - c) 邮件标题：“学号\_姓名\_实验 3”
4. 纸质版实验报告请在 5 月 12 日晚上 9 点之前提交到 G701。