**湖 州 学 院**

**学生实验报告**

课程名称： 《 数据结构 》

实验项目： 实验三 树和二叉树

**湖州学院教务处制**

**实验三 树和二叉树**

**一、 实验目的**

1．掌握树和二叉树的结构特征，以及各种存储结构的特点及适用范围。

2．掌握用指针类型描述、访问和处理二叉树的运算。

**二、实验内容**

1．输入先序遍历字符序列，建立二叉链表。

2．按先序、中序和后序遍历二叉树，输出所有的结点（递归算法）。

3．求二叉树的结点个数。

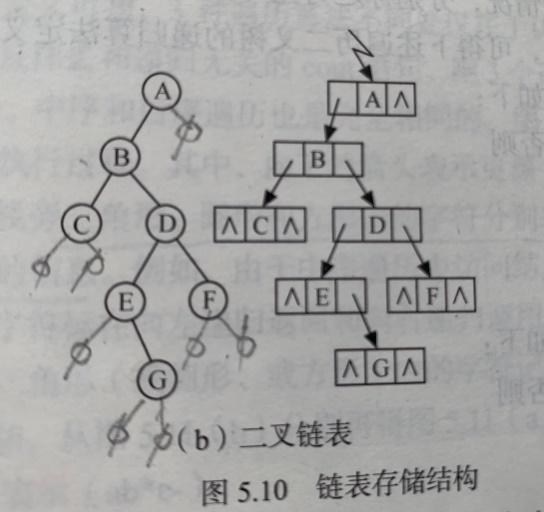
4．输出二叉树的叶子结点。

5．求二叉树的高度。

6．在主函数中设计一个简单的菜单，分别调试上述算法。

**说明：**

**输入的先序遍历字符序列为：ABC##DE#G##F###(期中’#’代表空树)。建立好的二叉树及二叉链表表示形式如下图所示：**

****

**三、源程序清单及实例的运行结果（源程序中应该附有必要的注释）。**

1.源程序清单：

#include<iostream>

using namespace std;

#define OK 1

typedef struct bitnode //二叉树的二叉链表存储表示

{

char data;

struct bitnode\* lchild, \* rchild;

}bitnode, \* bitree;

void creatbitree(bitree& t) //先序创建二叉树

{

char ch;

cin >> ch;

if (ch == '#')

t = NULL;

else

{

t = new bitnode;

t->data = ch;

creatbitree(t->lchild);

creatbitree(t->rchild);

}

}

void preorder(bitree t) //先序遍历二叉树

{

if (t)

{

cout << t->data;

preorder(t->lchild);

preorder(t->rchild);

}

}

void inorder(bitree t) //中序遍历二叉树

{

if (t)

{

inorder(t->lchild);

cout << t->data;

inorder(t->rchild);

}

}

void postorder(bitree t) //后序遍历二叉树

{

if (t)

{

postorder(t->lchild);

postorder(t->rchild);

cout << t->data;

}

}

int nodecount(bitree t) //求二叉树的结点个数

{

if (t == NULL)

return 0;

else

return nodecount(t->lchild) + nodecount(t->rchild) + 1;

}

void leafcount(bitree t) //求二叉树的叶子结点

{

if (t) {

if (!t->lchild && !t->rchild)

cout << t->data << " ";

leafcount(t->lchild);

leafcount(t->rchild);

}

}

int depth(bitree t) //求二叉树的高度

{

if (t == NULL)

return 0;

else

{

int l = depth(t->lchild);

int r = depth(t->rchild);

if (l > r)

return l + 1;

else

return r + 1;

}

}

int main()

{

bitree t;

cout << "请输入二叉树节点值：" ;

creatbitree(t);

cout << "------------------" << endl;

cout << "按序号输入执行程序" << endl;

cout << "先序遍历二叉树请输入1" << endl;

cout << "中序遍历二叉树请输入2" << endl;

cout << "后序遍历二叉树请输入3" << endl;

cout << "求二叉树节点数请输入4" << endl;

cout << "输出叶子节点请输入5" << endl;

cout << "求二叉树的高度请输入6" << endl;

cout << "退出程序请按0" << endl;

cout << "------------------" << endl;

int n;

while (OK)

{

cout << "请输入要执行的功能" << endl;

cin >> n;

switch (n)

{

case 1:

cout << "先序遍历" << endl;

preorder (t); //先序遍历

cout << endl;

break;

case 2:

cout << "中序遍历" << endl;

inorder(t); //中序遍历

cout << endl;

break;

case 3:

cout << "后序遍历" << endl;

postorder(t); //后序遍历

cout << endl;

break;

case 4:

cout << "二叉树节点个数为：" << nodecount(t) << endl; //二叉树结点个数

break;

case 5:

cout << "二叉树叶子节点为：";

leafcount(t);

cout << endl;

break;

case 6:

cout << "二叉树高度为：" << depth(t) << endl; //输出二叉树高度

break;

case 0:

exit(0);

}

}

return 0;

}

2.程序的运行结果：所有功能的运行结果都要截屏并打印出来



**四、实验小结**

1.说明遇到的问题及解决方法；

问题1

文本

描述已自动生成

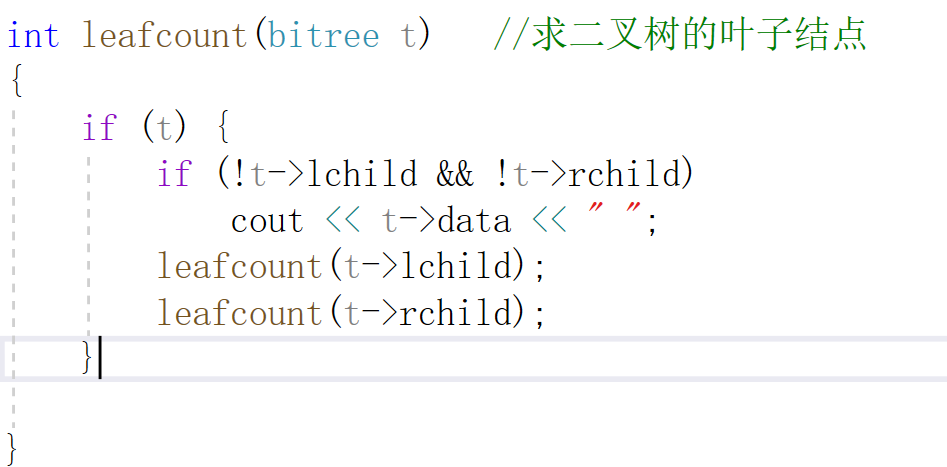
图形用户界面, 文本

描述已自动生成

题目要求的是求二叉树叶子节点，错误求成叶子节点个数

解决办法：重新编写程序

问题2



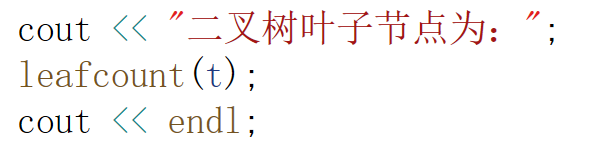


改为：

发现仍然出错



解决办法：将



改为:

2.其它要说明的问题。