代码：

#include <iostream>

using namespace std;

class Bitnode {

private:

char data;

Bitnode\* right;

Bitnode\* left;

public:

Bitnode(char data1) {

data = data1;

right = nullptr;

left = nullptr;

}

Bitnode\* getleft() { return left; }

Bitnode\* getright() { return right; }

char getdata() { return data; }

void cleft(Bitnode\* left1) { left=left1; }

void cright(Bitnode\* right1) { right=right1; }

void cdata(char data1) { data=data1; }

friend class Bitree;

};

class Bitree {

private:

Bitnode\* root;

public:

void Deletetree (Bitnode\* &T) {

if (T) {

Deletetree(T->left);

Deletetree(T->right);

delete T;

}

}

~Bitree() {

Deletetree(root);

}

void CreateBitree(Bitnode\*& T) {

char ch;

cin >> ch;

if (ch == '@') {

T= nullptr;

}else {

T = new Bitnode(ch);

CreateBitree(T->left);

CreateBitree(T->right);

}

}

void Preorder(Bitnode \*& T) {

if (T) {

cout << T->data;

Preorder(T->left);

Preorder(T->right);

}

}

void Inorder(Bitnode\*& T) {

if (T) {

Inorder(T->left);

cout << T->data;

Inorder(T->right);

}

}

void Postorder(Bitnode\* &T) {

if (T) {

Postorder(T->left);

Postorder(T->right);

cout << T->data;

}

}

int Count1(Bitnode\*& T) {

int n=0;

if (T) {

Count1(T->left);

Count1(T->right);

if (T->left && !T->right || !T->left && T->right)n++;

}

return n;

}

void CreateBitree() {

CreateBitree(root);

}

void Preorder() {

Preorder(root);

}

void Inorder() {

Inorder(root);

}

void Postorder() {

Postorder(root);

}

void Count1() {

cout << Count1(root);

}

};

int main()

{

Bitree T;

T.CreateBitree();//前序创造

T.Preorder();//前序遍历

cout << endl;

T.Inorder();//中序遍历

cout << endl;

T.Postorder();//后序遍历

cout << endl;

T.Count1();//计算度为1的节点

}

结果：

输入实例：abc@@de@g@@f@@@



