实验九 二叉树的建立及遍历应用

[ 发布人： [易法令](https://www.scholat.com/flyi)   截止时间：2021-11-18 23:59:59  ]

一、【实验目的】  
1、掌握二叉树的建立方法

2、掌握二叉树遍历的基本方法（前序、中序、后序）

3、掌握递归二叉树遍历算法的应用

二、【实验内容】

1.构造一棵二叉树,树的形态如下图(亦见附件)所示，打印出先序遍历、中序遍历、后序遍历的遍历序列。

            A

       B          F

   C          G

D     E

2.选择一种遍历方式计算该树中叶子结点的个数，并打印出叶子结点。

3.编写一个层序遍历算法，利用队列结构按层次（同一层自左至右）输出二叉树中所有的结点。

三、【实验源代码】

 #ifndef Treet2\_H

#define Treet2\_H

#include <malloc.h>

#include <stdio.h>

#define ElemType char

typedef struct node {

ElemType date;

struct node \*left;

struct node \*right;

} Node;

typedef struct {

Node \*root;

} Tree;

class flg{

public:

int flagg=0;

Node \*temp2;

void change();

void change1(Node \*temp);

};

void flg::change(){

flagg=-flagg;

}

void flg::change1(Node \*temp){

Node \*temp2=temp;

}

void insert(Tree \*tree, ElemType value,flg flg2) {

Node \*node = (Node \*)malloc(sizeof(Node));

node->date = value;

node->left = NULL;

node->right = NULL;

if (tree->root == NULL) {

tree->root = node;

}

else {

Node \*temp = tree->root;

while (temp != NULL) {

if (temp->left == NULL) {

temp->left = node;

return;

}

else if (temp->right == NULL) {

temp->right = node;

if(flg2.flagg==1){

temp = flg2.temp2;

flg2.change();

}

else{

return;

}

}

else {

temp = temp->left;

flg2.change();

flg2.change1(temp->right);

// temp2 = temp->right;

}

}

}

}

void inorder(Node \*node) {

if (node != NULL) {

inorder(node->left);

printf("node->date%c", node->date);

if(node->left!=NULL)printf(" left=%c ",node->left->date);

if(node->right!=NULL)printf(" right=%c ",node->right->date);

printf("\n");

inorder(node->right);

printf("\n");

}

}

void preorder(Node \*node){

if(node !=NULL){

printf("node->date%c", node->date);

if(node->left!=NULL)printf(" left=%c ",node->left->date);

if(node->right!=NULL)printf(" right=%c ",node->right->date);

printf("\n");

inorder(node->left);

inorder(node->right);

}

}

void postorder(Node \*node){

if(node !=NULL){

inorder(node->left);

inorder(node->right);

printf("node->date%c", node->date);

if(node->left!=NULL)printf(" left=%c ",node->left->date);

if(node->right!=NULL)printf(" right=%c ",node->right->date);

printf("\n");

}

}

#endif

#include <stdio.h>

#include "treet2.hpp"

int main(void) {

Tree mytree;

mytree.root = NULL;

int n;

scanf("%d", &n);

printf("n=%d\n",n);

flg flg1;

for (int i = 0; i < n; i++) {

char temp;

scanf("%c", &temp);

printf("tmep=%c ", temp);

insert(&mytree, temp,flg1);

}

printf("\ninorder:===========================\n");

inorder(mytree.root);

printf("\npreorder:==========================\n");

preorder(mytree.root);

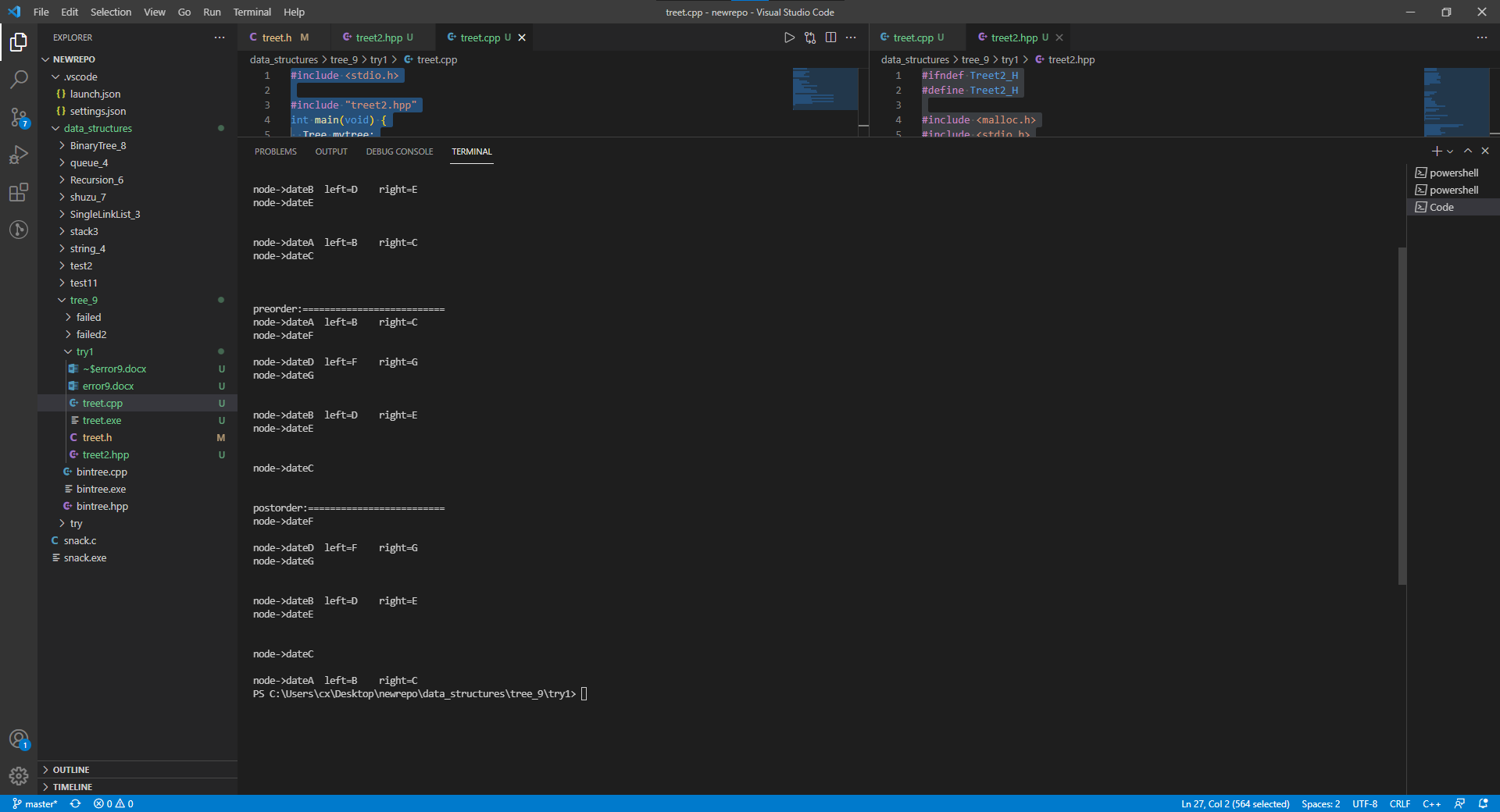
printf("\npostorder:=========================\n");

postorder(mytree.root);

;

}

四、【实验结果】



五、【实验心得】

1.类

没用好

2．C++写的跟屎一样



SCHOLAT.com 学者网

[免责声明 |](https://www.scholat.com/course/portaldisclaim.html) [关于学](https://www.scholat.com/course/portalaboutus.html)