1.  
  
#include <iostream>

using namespace std;

struct node{

int num;

node\* offspring[3];

};

void offspringsNULL(node \*nod)

{

for (int i = 0; i<3; i++)

nod->offspring[i] = NULL;

}

void createSimpleTree(node \*root)

{

root->offspring[0] = new node;

offspringsNULL(root->offspring[0]);

root->offspring[0] -> num = 2;

root->offspring[0] -> offspring[0] = new node;

offspringsNULL(root->offspring[0] -> offspring[0]);

root->offspring[0] -> offspring[0] ->num = 5;

root->offspring[1] = new node;

offspringsNULL(root->offspring[1]);

root->offspring[1]->num = 3;

root->offspring[1]->offspring[0] = new node;

offspringsNULL(root->offspring[1]->offspring[0]);

root->offspring[1]->offspring[0]->num = 6;

root->offspring[1]->offspring[1] = new node;

offspringsNULL(root->offspring[1]->offspring[1]);

root->offspring[1]->offspring[1]->num = 7;

root->offspring[2] = new node;

root->offspring[2]->num = 4;

root->offspring[2]->offspring[0] = new node;

offspringsNULL(root->offspring[2]->offspring[0]);

root->offspring[2]->offspring[0]->num = 8;

root->offspring[2]->offspring[0]->offspring[0] = new node;

offspringsNULL(root->offspring[2]->offspring[0]->offspring[0]);

root->offspring[2]->offspring[0]->offspring[0]->num = 9;

root->offspring[2]->offspring[0]->offspring[1] = new node;

offspringsNULL(root->offspring[2]->offspring[0]->offspring[1]);

root->offspring[2]->offspring[0]->offspring[1]->num = 10;

root->offspring[2]->offspring[0]->offspring[2] = new node;

offspringsNULL(root->offspring[2]->offspring[0]->offspring[2]);

root->offspring[2]->offspring[0]->offspring[2]->num = 11;

}

int show (node\* nod, int level, int count)

{

if (!nod)

return 0;

;

// cout<<"levl ="<<level<< endl;

for (int i = 0; i < 3; i++){

show (nod->offspring[i], level++, count);

if (count < level) count=level;}

return count;

}

int main(){

node\* root;

root = new node;

root->num = 1;

root->offspring[0] = root->offspring[1] = root->offspring[2] = NULL;

createSimpleTree(root);

;

cout << show (root, 1, 1) << endl;

return 0;

}

2.#include <iostream>

using namespace std;

struct node{

int num;

node\* left;

node\* right;

};

void addNode(node\* nod, int data){

if (data < nod->num)

if (nod->left)

addNode(nod->left, data);

else

{

node\* newNode = new node;

newNode->num = data;

newNode->left = newNode->right = NULL;

nod->left = newNode;

}

else

if (nod->right)

addNode(nod->right, data);

else

{

node\* newNode = new node;

newNode->num = data;

newNode->left = newNode->right = NULL;

nod->right = newNode;

}

}

void showByLevels(node\* nod, bool sibling){

if (!nod)

return;

cout << nod->num;

if (nod->left) {

cout << "( ";

showByLevels(nod->left, true);

}

if (nod->right) {

cout << "( ";

showByLevels(nod->right, true);

}

cout << ")";

}

int main(){

int arr[14] = {9,14,5,7,13,16,3,12,20,4,11,6,15,16};

node\* root = new node;

root->left = root->right = NULL;

root->num = 10;

for (int i = 0; i < 14; i++)

addNode(root, arr[i]);

cout<< endl<<"showByLevels "<<endl;

showByLevels(root, false);

return 0;

}