#include <iostream>

using namespace std;

class node{

public:

int data;

node \*left;

node \*right;

node(int d){

data=d;

left=NULL;

right=NULL;

}

};

node\* buildTree(){

int d;

cin>>d;

if(d==-1)

return NULL;

node \*root=new node(d);

root->left=buildTree();

root->right=buildTree();

return root;

}

//Preorder root-lrft-right

void print(node \*root){

if(root==NULL)

return;

cout<<root->data<<" ";

print(root->left);

print(root->right);

}

//Postorder left-right-root

void printPost(node \*root){

if(root==NULL)

return;

printPost(root->left);

printPost(root->right);

cout<<root->data<<" ";

}

//Inorder left-root-right

void printIn(node \*root){

if(root==NULL)

return;

printIn(root->left);

cout<<root->data<<" ";

printIn(root->right);

}

int height(node \*root){

if(root==NULL)

return 0;

int ls=height(root->left);

int rs=height(root->right);

return max(ls,rs)+1;

}

void printKthLevel(node \*root,int k){

if(root==NULL)

return;

if(k==1){

cout<<root->data<<" ";

return;

}

printKthLevel(root->left,k-1);

printKthLevel(root->right,k-1);

return;

}

void printAllLevel(node \*root){

int h=height(root);

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

printKthLevel(root,i);

cout<<endl;

}

}

int main() {

node \*root=buildTree();

print(root);

cout<<endl;

printIn(root);

cout<<endl;

printPost(root);

cout<<endl;

cout<<height(root); //print height of the tree

cout<<endl;

printKthLevel(root,3); //print all nodes at kth level

cout<<endl;

printAllLevel(root); // to print all elements level wise

}

Input-

3 4 -1 6 -1 -1 5 1 -1 -1 -1

Output-

3 4 6 5 1

4 6 3 1 5

6 4 1 5 3

3

6 1

3

4 5

6 1