#include<iostream>

using namespace std;

struct Node

{

int data;

Node\* left;

Node\* right;

Node(int d){

data=d;

left=right=NULL;

}

};

bool search(Node\* root,int x){

while(root!=NULL){

if(root->data==x){

return true;

}

else if(root->data>x){

root=root->left;

}

else{

root=root->right;

}

}

}

Node\* insert(Node\* root,int x){

Node\* temp=new Node(x);

Node\* curr=root;

Node\* parent=NULL;

while(curr!=NULL){

parent=curr;

if(curr->data>x){

curr=curr->left;

}

else if(curr->data<x){

curr=curr->right;

}

else{

return root;

}

}

if(parent==NULL)

return temp;

else if(parent->data>x){

parent->left=temp;

}

else

parent->right=temp;

return root;

}

void inorder(Node\* root){

if(root!=NULL){

inorder(root->left);

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

inorder(root->right);

}

}

int height(Node\* root){

if(root==NULL){

return 0;

}

else{

return(1+max(height(root->left),height(root->right)));

}

}

int findmin(Node\* root){

if(root==NULL){

return 0;

}

else{

root=root->left;

findmin(root->left);

return (root->data);

}

}

Node\* mirror(Node\* root){

if(root!=NULL){

Node\* temp;

mirror(root->left);

mirror(root->right);

temp=root->left;

root->left=root->right;

root->right=temp;

//return root;

}

}

int main(){

int choice,ch;

Node\* root=new Node(10);

root->left=new Node(5);

root->right=new Node(15);

root->right->left=new Node(12);

root->right->right=new Node(18);

int x=20;

if(search(root,x)){

cout<<"Element found"<<"\n";

}

else{

cout<<"Element not found"<<"\n";

}

cout<<endl;

inorder(root);

cout<<endl;

inorder(insert(root,x));

cout<<endl;

cout<<height(root)<<"\n";

cout<<endl;

cout<<findmin(root);

cout<<endl;

root=mirror(root);

inorder(root);

}