Tree path

#include <iostream>

using namespace std;

class Node{

public:

int data;

Node \*left;

Node \*right;

Node(){}

Node(int data){

this->data = data;

left = right = NULL;

}

};

class Tree{

public:

Node \*root;

Tree(){

root = NULL;

}

void insert(int n){

root = insert(root, n);

}

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

if( root == NULL)

return new Node(n);

if( n > root->data)

root->right = insert(root->right, n);

else

root->left = insert(root->left, n);

return root;

}

void printpath(int n){

string path = printpath(root, n, "Root");

if( path.length() > 0){

cout<<path<<endl;

}else{

cout<<"Not Found"<<endl;

}

}

string printpath(Node\* root,int n, string path){

if( root != NULL){

if( root->data == n){

return path;

}

string rpath = "";

rpath = printpath( root->right, n, path+" R");

if( rpath.length() > 0)

return rpath;

rpath = printpath( root->left, n, path+" L");

if( rpath.length() > 0)

return rpath;

}

return "";

}

};

int main(){

int n;

Tree tree;

do{

cin>>n;

if( n == -1)

break;

tree.insert(n);

}while( true );

cin>>n;

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

int x;

cin>>x;

tree.printpath(x);

}

}