#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;

}

class HbPair{

public:

int height;

bool balance;

};

HbPair isHeightBalanced(node \*root){

HbPair p;

if(root==NULL){

p.height=0;

p.balance=true;

return p;

}

HbPair left=isHeightBalanced(root->left);

HbPair right=isHeightBalanced(root->right);

p.height=max(left.height,right.height)+1;

if(abs(left.height-right.height)<=1 && left.balance && right.balance){

p.balance=true;

}

else{

p.balance=false;

}

return p;

}

int main() {

node \*root=buildTree();

if(isHeightBalanced(root).balance){

cout<<"Balanced";

}

else{

cout<<"Not balanced";

}

}

Input-

8 10 1 -1 -1 6 9 -1 -1 7 -1 -1 3 -1 14 13 -1 -1 -1

Output-

Not balanced