Program that will return true if the binary tree is a mirror image of itself, and false if it is not.

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* left;

Node\* right;

Node(int data) {

this->data = data;

this->left = NULL;

this->right = NULL;

}

};

bool is\_mirror(Node\* root1, Node\* root2) {

if (root1 == NULL && root2 == NULL) {

return true;

}

if (root1 != NULL && root2 != NULL) {

if (root1->data == root2->data) {

return (is\_mirror(root1->left, root2->right) &&

is\_mirror(root1->right, root2->left));

}

}

return false;

}

bool is\_symmetric(Node\* root) {

return is\_mirror(root, root);

}

int main() {

Node\* root = new Node(1);

root->left = new Node(2);

root->right = new Node(2);

root->left->left = new Node(3);

root->left->right = new Node(4);

root->right->left = new Node(4);

root->right->right = new Node(3);

cout << is\_symmetric(root) << endl;

return 0;

}