#include <iostream>

#include <stack>

using namespace std;

// Binary Tree node

struct Node {

    int data;

    struct Node \*left, \*right;

};

void zizagtraversal(struct Node\* root)

{

    // if null then return

    if (!root)

        return;

    // declare two stacks

    stack<struct Node\*> currentlevel;

    stack<struct Node\*> nextlevel;

    // push the root

    currentlevel.push(root);

    // check if stack is empty

    bool lefttoright = true;

    while (!currentlevel.empty()) {

        // pop out of stack

        struct Node\* temp = currentlevel.top();

        currentlevel.pop();

        // if not null

        if (temp) {

            // print the data in it

            cout << temp->data << " ";

            // store data according to current

            // order.

            if (lefttoright) {

                if (temp->left)

                    nextlevel.push(temp->left);

                if (temp->right)

                    nextlevel.push(temp->right);

            }

            else {

                if (temp->right)

                    nextlevel.push(temp->right);

                if (temp->left)

                    nextlevel.push(temp->left);

            }

        }

        if (currentlevel.empty()) {

            lefttoright = !lefttoright;

            swap(currentlevel, nextlevel);

        }

    }

}

struct Node\* newNode(int data)

{

    struct Node\* node = new struct Node;

    node->data = data;

    node->left = node->right = NULL;

    return (node);

}

int main()

{

    // create tree

    struct Node\* root = newNode(1);

    root->left = newNode(2);

    root->right = newNode(3);

    root->left->left = newNode(7);

    root->left->right = newNode(6);

    root->right->left = newNode(5);

    root->right->right = newNode(4);

    cout << "ZigZag Order traversal of binary tree is \n";

    zizagtraversal(root);

    return 0;

}