#include <bits/stdc++.h>

using namespace std;

// Class of the Binary Tree node

struct Node

{

int data;

Node \*left, \*right;

Node(int x)

{

data = x;

left = right = NULL;

}

};

vector<int> lst;

// Function to find the nodes

// having single child

void printNodesOneChild(Node\* root)

{

// Base case

if (root == NULL)

return;

// Condition to check if the

// node is having only one child

if (root->left != NULL &&

root->right == NULL ||

root->left == NULL &&

root->right != NULL)

{

lst.push\_back(root->data);

}

// Traversing the left child

printNodesOneChild(root -> left);

// Traversing the right child

printNodesOneChild(root -> right);

}

//Driver code

int main()

{

// Constructing the binary tree

Node \*root = new Node(2);

root -> left = new Node(3);

root -> right = new Node(5);

root -> left -> left = new Node(7);

root -> right -> left = new Node(8);

root -> right -> right = new Node(6);

// Function calling

printNodesOneChild(root);

// Condition to check if there is

// no such node having single child

if (lst.size() == 0)

printf("-1");

else

{

for(int value : lst)

{

cout << (value) << endl;

}

}

}