Ex1)

For a given binary tree, write a program to output the leftmost longest path from the leaf node to the root node

#include <iostream>

#include <vector>

using namespace std;

struct TreeNode {

int val;

TreeNode\* left;

TreeNode\* right;

TreeNode(int v) : val(v), left(nullptr), right(nullptr) {}

};

void getLeftmostLongestPath(TreeNode\* root, vector<int>& path, vector<int>& result, int& maxlen) {

if (!root) return;

path.push\_back(root->val);

if (!root->left && !root->right) { // leaf node

if (path.size() > maxlen) {

result = path;

maxlen = path.size();

}

}

getLeftmostLongestPath(root->left, path, result, maxlen);

getLeftmostLongestPath(root->right, path, result, maxlen);

path.pop\_back();

}

vector<int> leftmostLongestPath(TreeNode\* root) {

vector<int> result;

vector<int> path;

int maxlen = 0;

getLeftmostLongestPath(root, path, result, maxlen);

return result;

}

int main() {

TreeNode\* root = new TreeNode(1);

root->left = new TreeNode(2);

root->right = new TreeNode(3);

root->left->left = new TreeNode(4);

root->left->right = new TreeNode(5);

root->right->left = new TreeNode(6);

root->right->right = new TreeNode(7);

root->left->left->left = new TreeNode(8);

root->left->right->right = new TreeNode(9);

vector<int> path = leftmostLongestPath(root);

for (int x : path) cout << x << " "; // output: 8 4 2 1

cout << endl;

return 0;

}