Aim-Given the root of a binary tree, return *all root-to-leaf paths in****any order***.

#include <iostream>

#include <vector>

#include <string>

struct TreeNode {

int val;

TreeNode\* left;

TreeNode\* right;

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

};

void findPaths(TreeNode\* root, std::string path, std::vector<std::string>& paths) {

if (!root) return;

path += std::to\_string(root->val);

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

paths.push\_back(path);

return;

}

path += "->";

if (root->left) findPaths(root->left, path, paths);

if (root->right) findPaths(root->right, path, paths);

}

std::vector<std::string> binaryTreePaths(TreeNode\* root) {

std::vector<std::string> paths;

findPaths(root, "", paths);

return paths;

}

int main() {

TreeNode\* root = new TreeNode(1);

root->left = new TreeNode(2);

root->right = new TreeNode(3);

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

std::vector<std::string> paths = binaryTreePaths(root);

std::cout << "Root-to-leaf paths:\n";

for (const std::string& path : paths) {

std::cout << path << "\n";

}

// Clean up memory

delete root->left->right;

delete root->left;

delete root->right;

delete root;

return 0;

}