#include <iostream>

#include <vector>

class TreeNode {

public:

int value;

TreeNode\* left;

TreeNode\* right;

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

};

void preOrderTraversal(TreeNode\* root, std::vector<int>& result) {

if (root == nullptr) {

return;

}

// Visit the root

result.push\_back(root->value);

// Visit the left subtree

preOrderTraversal(root->left, result);

// Visit the right subtree

preOrderTraversal(root->right, result);

}

int main() {

// Creating a sample binary tree

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);

std::vector<int> result;

preOrderTraversal(root, result);

// Printing the pre-order traversal result

for (int val : result) {

std::cout << val << " ";

}

std::cout << std::endl;

return 0;

}