

```

#include <iostream>
using namespace std;
// Node Structure
struct Node
{
    int data;
    Node *left;
    Node *right;
};
// Create Node Function
Node *createNode(int value)
{
    Node *newNode = new Node();
    newNode->data = value;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}
// PreOrder Traversal Function
void preorder(Node *root)
{
    if (root == NULL)
        return;

    cout << root->data << " ";
    preorder(root->left);
    preorder(root->right);
}
// Inorder Traversal Function
void inorder(Node *root)
{
    if (root == NULL)
        return;

    inorder(root->left);
    cout << root->data << " ";
    inorder(root->right);
}
// Postorder Traversal Function
void postorder(Node *root)
{
    if (root == NULL)

```

```

        return;

    postorder(root->left);
    postorder(root->right);
    cout << root->data << " ";
}
// Main Function
int main()
{
    Node *root = createNode(1);
    root->left = createNode(2);
    root->right = createNode(3);
    root->left->left = createNode(4);
    root->left->right = createNode(5);

    cout << "Binary Tree Traversals:" << endl;
    cout << "-----" << endl;

    cout << "Preorder (Root-L-R): ";
    preorder(root);
    cout << endl;

    cout << "Inorder (L-Root-R): ";
    inorder(root);
    cout << endl;

    cout << "Postorder (L-R-Root): ";
    postorder(root);
    cout << endl;

    return 0;
}

```