

```

#include <iostream>
#include <vector>
#include <sstream>
using namespace std;

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

vector<TreeNode*> buildTrees(vector<int>& inorder, int start, int end) {
    vector<TreeNode*> trees;
    if (start > end) {
        trees.push_back(nullptr);
        return trees;
    }

    for (int i = start; i <= end; i++) {
        vector<TreeNode*> leftTrees = buildTrees(inorder, start, i - 1);
        vector<TreeNode*> rightTrees = buildTrees(inorder, i + 1, end);

        for (TreeNode* left : leftTrees) {
            for (TreeNode* right : rightTrees) {
                TreeNode* root = new TreeNode(inorder[i]);
                root->left = left;
                root->right = right;
                trees.push_back(root);
            }
        }
    }

    return trees;
}

vector<int> preorder(TreeNode* root) {
    vector<int> res;
    if (root == nullptr) {
        return res;
    }
    res.push_back(root->val);
    vector<int> left = preorder(root->left);
    vector<int> right = preorder(root->right);
    res.insert(res.end(), left.begin(), left.end());
    res.insert(res.end(), right.begin(), right.end());
    return res;
}

int main() {
    int n;
    cout << "Enter the number of test cases: ";
    cin >> n;

    cin.ignore(); // ignore the newline character left in the buffer
    after entering n

    vector<vector<int>> > inorders(n);

```

```

    for (int i = 0; i < n; i++) {
        string line;
        cout << "Enter the in-order traversal vector for test case #" <<
i + 1 << ": " << endl;
        getline(cin, line);
        istringstream iss(line);
        int val;
        while (iss >> val) {
            inorders[i].push_back(val);
        }
    }

    for (int i = 0; i < n; i++) {
        vector<TreeNode*> trees = buildTrees(inorders[i], 0,
inorders[i].size() - 1);

        cout << "Preorder traversal for tree #" << i + 1 << ": " << endl;
        for (TreeNode* root : trees) {
            vector<int> pre = preorder(root);
            for (int val : pre) {
                cout << val << " ";
            }
            cout << endl;
        }
    }

    return 0;
}

```