

# 构建二叉树

## 输入描述

输入分为多行。第一行为一个整数，表示根节点存储的数据；其余行每行描述一个节点：一个整数和一个仅由 L 和 R 组成的字符串（长度大于等于 1），整数表示该节点存储的数据，字符串表示该节点在二叉树中的位置。

输入保证节点的父节点已经存在。

## 输出描述

不同于之前使用 `std::cout` 输出的方式，本题中调用提供的 DLL 中的 `print_tree` 函数输出结果。`print_tree` 函数的签名为：

```
namespace cpp {  
    void print_tree(const TreeNode *root);  
}
```

## 示例

### 示例 1

输入：1

输出：[1]

[1]

解释：输出的第一行为二叉树的先序遍历序列，第二行为二叉树的中序遍历序列

### 示例 2

输入：1

2 L

3 R

4 LL

5 LR

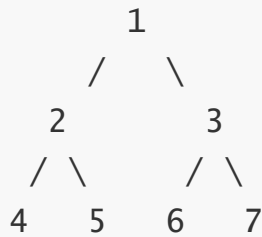
6 RL

7 RR

输出: [1,2,4,5,3,6,7]

[4,2,5,1,6,3,7]

解释: 组成的二叉树如下图所示



以 5 LR 为例, 表示数据为 5 的节点位于根的左子树的右子树

## 提示

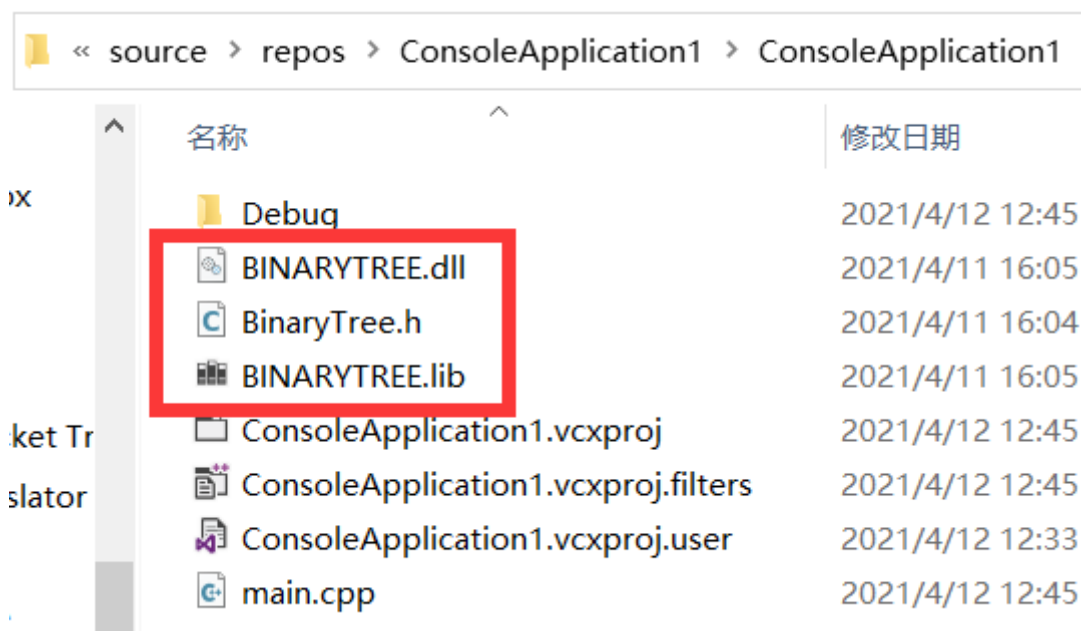
### 树节点的数据结构

```
struct TreeNode {
    int data;
    TreeNode *left, *right;

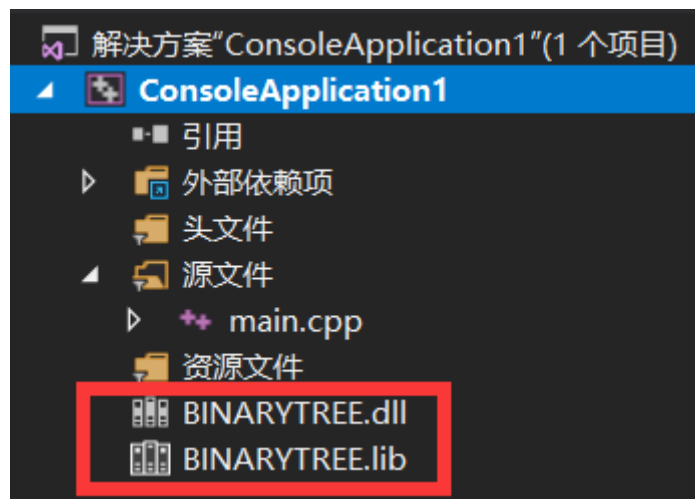
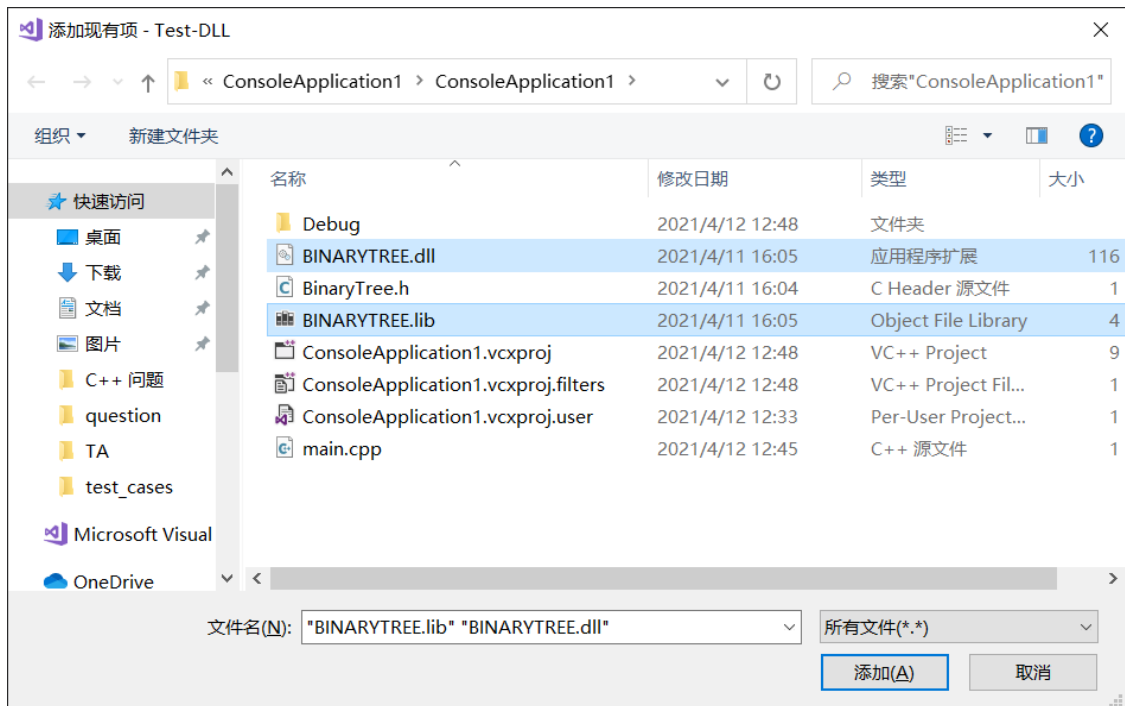
    TreeNode(void);
    TreeNode(int data);
};
```

### 在 Visual C++ 项目中引入 DLL

1. 将 question 目录下的 BINARYTREE.dll、BINARYTREE.lib 和 BinaryTree.h 三个文件复制到项目根目录下, 例如 (解决方案名和项目名均为 ConsoleApplication1) :



2. 在“解决方案资源管理器”中右键单击项目名，选择“添加 > 现有项”，在弹出的对话框中选择 `BINARYTREE.dll` 和 `BINARYTREE.lib`，点击添加



3. 在源代码中引入 `BinaryTree.h`，如下图所示

```
#include <iostream>

#include "BinaryTree.h"

using namespace std;

int main() {
    cpp::TreeNode a(42);
    cpp::print_tree(&a);
}
```

4. 现在代码中可以使用 `TreeNode` 结构和 `print_tree` 函数，如上图
5. 此外，把 `BINARYTREE.dll` 和 `BINARYTREE.lib` 文件添加到解决方案的 `Debug` 目录中（如果没有这个目录，先生成解决方案），否则插件在测试时找不到这两个文件会导致 `WA`