实验二、树的应用

**重要提醒**：

**样例里有null的，在代码实现时用-1来代替**

1.编写程序判断树是否同构？其中同构是指给定两棵树T1和T2。如果T1可以通过若干次左右孩子互换就变成T2，则称两棵树是“同构”的，输出True，否则输出False。

输入：第一行N，表示树的结点数；

第二行开始：结点数据   左孩子编号   右孩子编号，如果无孩子结点记为“-1”。

输入样例：

8

A 1 2

B 3 4

C 5 -1

D -1 -1

E 6 -1

G 7 -1

F -1 -1

H -1 -1

8

G -1 4

B 7 6

F -1 -1

A 5 1

H -1 -1

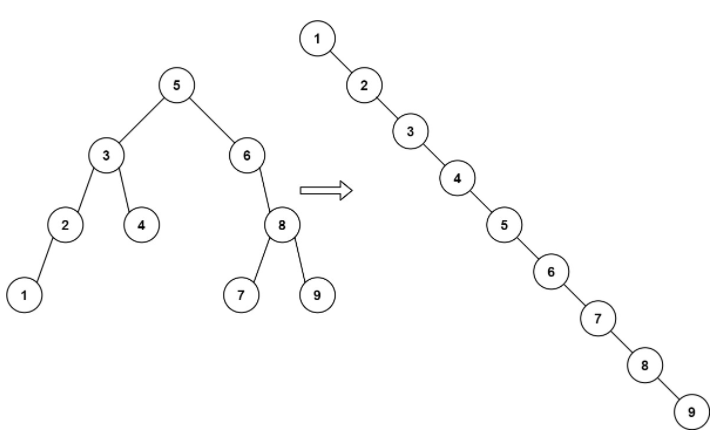
C 0 -1

D -1 -1

E 2 -1

输出样例：True

2、给定一棵二叉搜索树，请 按中序遍历将其重新排列为一棵递增顺序搜索树，使树中最左边的节点成为树的根节点，并且每个节点没有左子节点，只有一个右子节点。例如，将左下图的二叉搜索树转换为右下图的树。



输入样例：5 3 6 2 4 null 8 1 null null null 7 9

输出样例：1 null 2 null 3 null 4 null 5 null 6 null 7 null 8 null 9

3.给定一个二叉树的根节点 root ，判断其是否是一个有效的二叉搜索树。

输入样例1：

输入：2 1 3

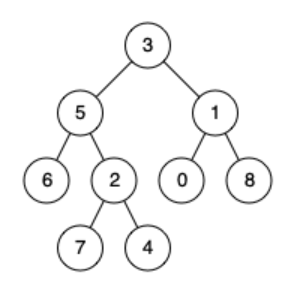
输出：True

输入样例2：

输入：5 1 4 null null 3 6

输出：False

4. 给定一个二叉树， 编写算法计算二叉树中任意两个结点的公共祖先。其中，输入第一行为二叉树序列，第二行和第三行分别为两个节点编号；输出：两个节点的公共祖先。例如:



输入样例1：

输入：

3 5 1 6 2 0 8 null null 7 4

5

1

输出: 3

输入样例2：

输入:

3 5 1 6 2 0 8 null null 7 4

5

4

输出: 5

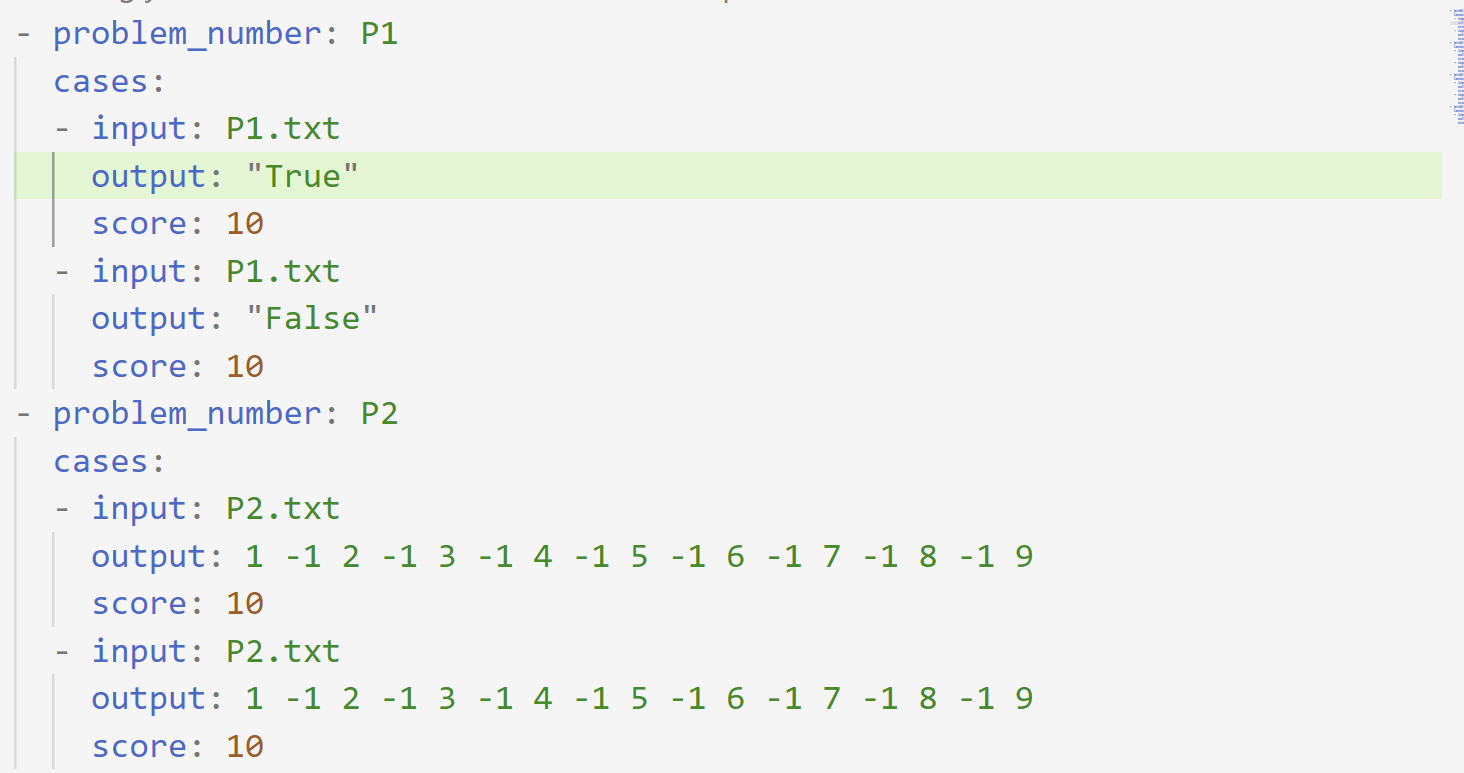
**注：以下内容来自助教**

源代码文件命名格式：学号-姓名-Px-Y/N.cpp，其中Y/N表示的是否使用模板。

第一次实验提供了模板给大家使用，但截止2022-04-26提交上来的同学，使用模板的人数很少。考虑到有些同学可能想使用模板，但是害怕脚本无法正确检测。

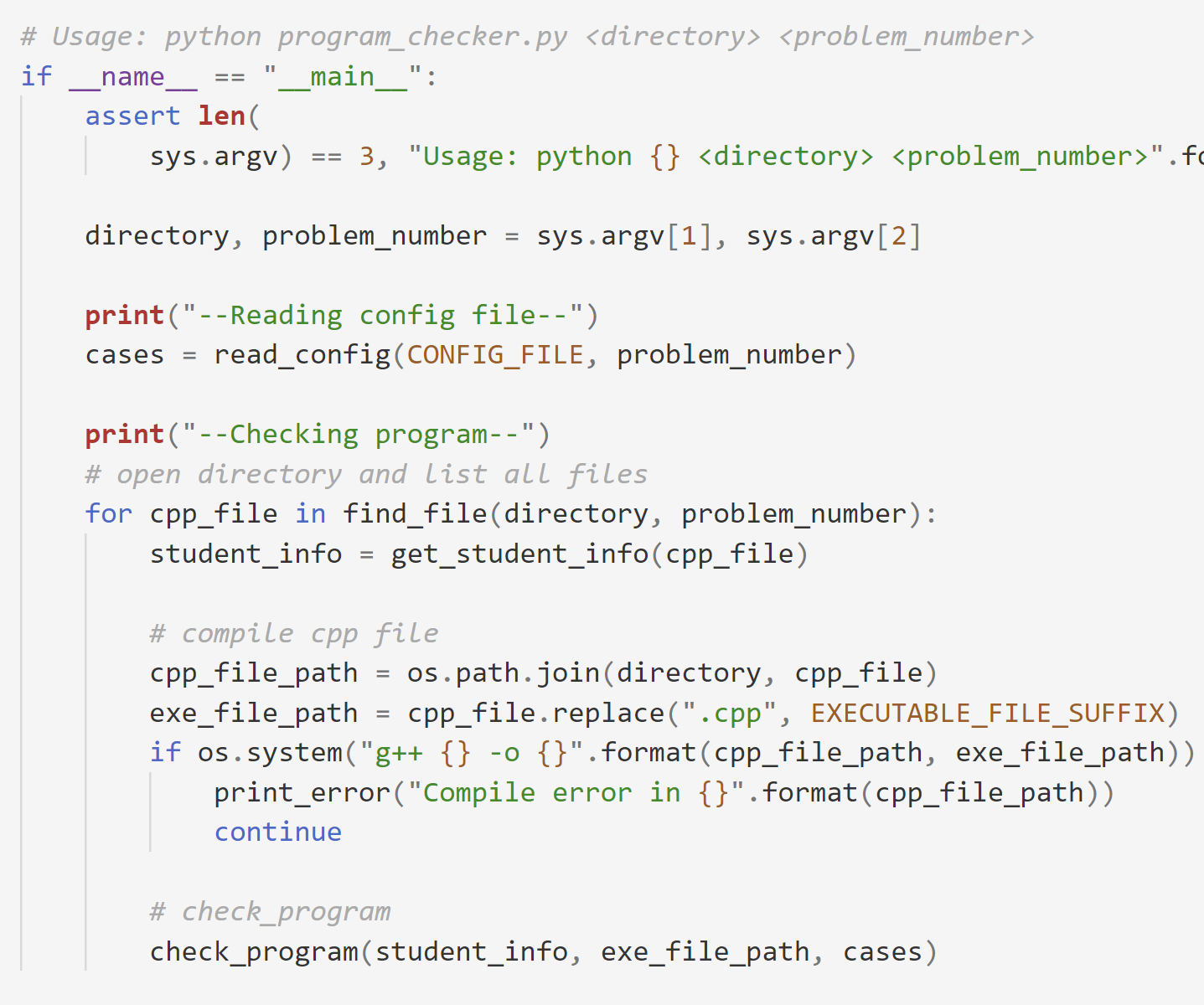
**所以这次我们把脚本提供出来！！！下面是有关测试脚本的介绍：**

**配置文件 config.yaml**



配置文件包含题目编号和测试样例，其中测试样例可以多个。**在使用过程中，只要更改input和相应的output就行，score仅作为参考。**

**测试脚本 program\_checker.py**

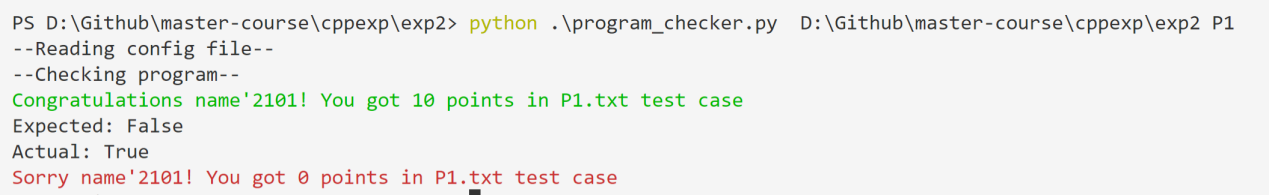


使用方法：安装python，安装pyyaml package.

然后执行 python program\_checker.py <directory> <problem\_number>

**其中directory是程序的文件夹，problem\_number是要测试的问题。**

**使用效果如下：**



**注意：**

1. 如果你选择使用模板，建议把**脚本，配置文件，输入文件，源程序放到同一个文件夹下**。如果没有的话，你可能会碰到各种文件路径问题，fix这个bug的最简单方法就是把程序里的文件路径改成绝对路径。
2. 我们的测试环境是window，如果是其它环境，需要修改可执行文件的后缀

