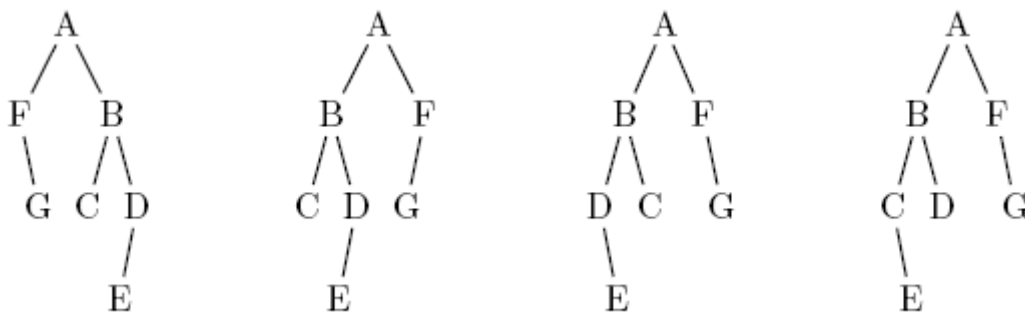


3538 Newton's Apple

Two binary trees (called **A** and **B**) are *equivalent* if and only if one of the following two conditions holds:

1. Both trees are empty. Or,
2. The root nodes of both trees are equal, and either:
 - (a) the left subtree of **A** is *equivalent* to the left subtree of **B**, and the right subtree of **A** is equivalent to the right subtree of **B**. Or,
 - (b) the left subtree of **A** is *equivalent* to the right subtree of **B**, and the right subtree of **A** is equivalent to the left subtree of **B**.

For example, the three trees on the left of the following figure are all equivalent to each other but none is equivalent to the right-most tree.



Write a program that determines if two given binary trees are equivalent or not.

Input

Your program will be tested on a number of test cases. The first line of the input contains an integer D which represents the number of test cases. Each test case is specified on two lines. The first line specifies the first tree, with the second tree specified on the second line. Each tree is specified using a left-to-right *postfix* notation where each empty subtree is explicitly specified using the keyword 'nil'. All data in the tree are upper-case letters. The end of the line is specified using the keyword 'end'.

For example, the tree on the left of the figure is specified as:

```
nil nil nil G F nil nil C nil nil E nil D B A end
```

Output

For each test case, print on a separate line, the word 'true' if the two trees are equivalent. Otherwise print 'false'.

The two test cases in the following sample I/O represent the four trees drawn on the previous page (from left to right)

Sample Input

```
2
nil nil nil G F nil nil C nil nil E nil D B A end
nil nil C nil nil E nil D B nil nil G nil F A end
nil nil nil E D nil nil C B nil nil nil G F A end
nil nil nil E C nil nil D B nil nil nil G F A end
```

Sample Output

```
true
false
```