Would Be... Could Be... BST!

Problem Statement

This challenge is sponsored by IEEE Xplore.

You are given an ambiguous description of one or more binary trees. Each key value is a non-empty string of lowercase letters, and each key occurs in only one node in the collection. You must decide if the description that is given must have been derived from a single binary search tree, could have been derived from a single binary search tree, or could not have been derived from a binary search tree. Please refer to http://en.wikipedia.org/wiki/Binary_search_tree for background information on binary search trees.

Note: The default time limits for all languages have been doubled for this program.

Input Format

The input is made up of multiple test cases. Each test case begins with an integer N, with 1 <= N <= 50,000. Next come N lines each describing an edge in the collection of binary trees:

[key1] [key2]

where [key1] and [key2] are strings corresponding the key values of two nodes, *node1* and *node2*, respectively, where *node1* is either the left child of *node2* OR *node2* is the right child of *node1*.

The last line of input following all of the test cases consists of the number 0 on a line by itself. The total number of edges in all test cases in a single input file will be less than or equal to 500,000.

Output Format

For each test case, you will output a single string followed by the newline character:

- If the description given in the test case must correspond to a binary search tree, you should output “BST!”.
- If the description given in the test case corresponds to a binary search tree in some instantiations but not in others, you should output “BST?”.
- If the description given in the test case cannot describe a binary search tree, you should output “!BST”.

Sample Input

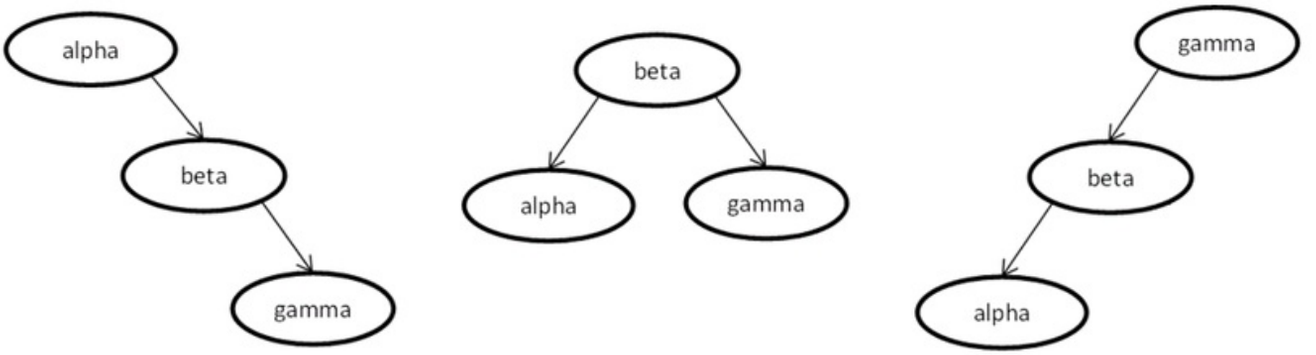
```
2
alpha beta
beta gamma
2
alpha delta
alpha gamma
2
alpha beta
delta gamma
1
beta alpha
0
```

Sample Output

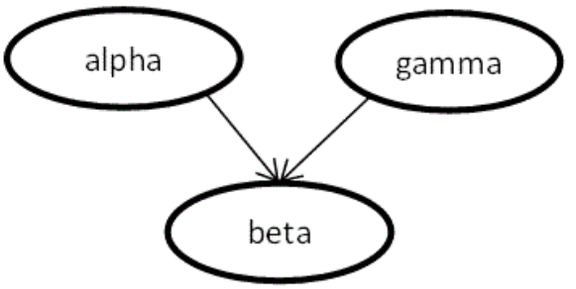
```
BST!
BST?
!BST
!BST
```

Explanation

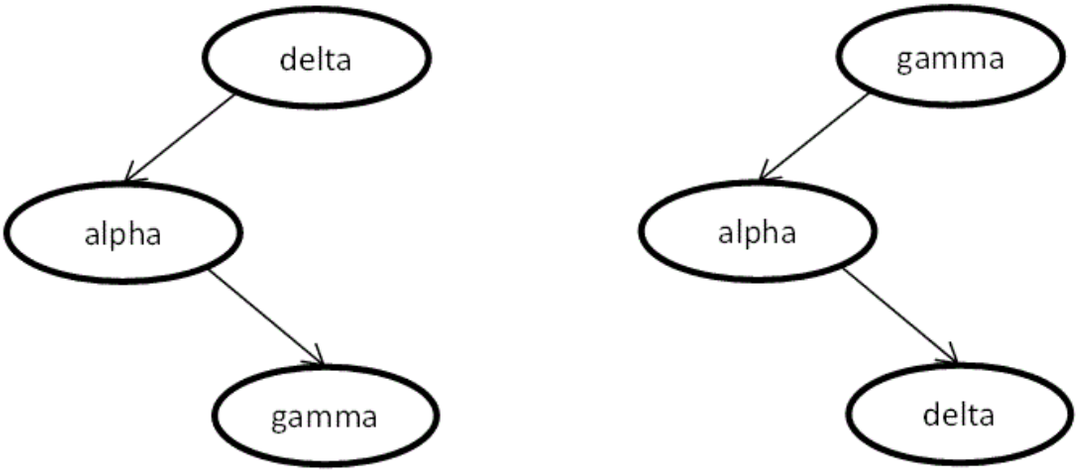
The first test case could have been derived from three different configurations, which are shown below. Note that all of these configurations are binary search trees.



NOTE: Since each test case is a description of one or more binary trees, the description for the first test case could not have come from a configuration like the one below which is not a binary tree:



The second test case describes one of the configurations below. The tree on the right is a binary search tree because it fulfills the binary search tree property, (i.e. the key in each node of the tree is greater than all the keys stored in the left sub-tree, and smaller than all the keys in the right sub-tree). However, the configuration on the left is not a binary search tree because the key value stored at the root of the tree is not smaller than all the values stored in the left sub-tree (i.e. value "gamma" of the left subtree is greater than "delta").



The third test case describes a forest of two trees, which cannot be a BST.

The last test case cannot be a BST.