```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:44 CST 2018*/

/*Given two binary trees, write a function to check if they are the same or not.

        Two binary trees are considered the same if they are structurally identical
        and the nodes have the same value.


        Example 1:

        Input:     1         1
        / \       / \
        2   3     2   3

        [1,2,3],   [1,2,3]

        Output: true
        Example 2:

        Input:     1         1
        /           \
        2             2

        [1,2],     [1,null,2]

        Output: false
        Example 3:

        Input:     1         1
        / \       / \
        2   1     1   2

        [1,2,1],   [1,1,2]

        Output: false*/


import java.util.*;
import java.lang.Math;
import java.lang.System;
import java.lang.Integer;


public class Main {

    public static void main(String[] args) throws ArithmeticException {

        String input = "ab";

        boolean answer = new Solution().repeatedSubstringPattern(input);

        System.out.println("haha");
    }

}


class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
    }
}


class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

```

```
    TreeNode(int x) {
        val = x;
    }
}


class Solution {
    public boolean isSameTree(TreeNode p, TreeNode q) {

        if (p == null && q == null) {
            return true;
        } else if (p == null && q != null || p != null && q == null) {
            return false;
        } else {
            if (p.val != q.val) {
                return false;
            } else {
                return isSameTree(p.left, q.left) && isSameTree(p.right, q.right);
            }
        }
    }
}
```