```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:45 CST 2018*/

/*You are climbing a stair case. It takes n steps to reach to the top.

        Each time you can either climb 1 or 2 steps. In how many distinct ways can
        you climb to the top?

        Note: Given n will be a positive integer.


        Example 1:

        Input: 2
        Output:  2
        Explanation:  There are two ways to climb to the top.

        1. 1 step + 1 step
        2. 2 steps
        Example 2:

        Input: 3
        Output:  3
        Explanation:  There are three ways to climb to the top.

        1. 1 step + 1 step + 1 step
        2. 1 step + 2 steps
        3. 2 steps + 1 step*/


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 {

```

```java
    public int climbStairs(int n) {

        if (n == 1) {
            return 1;
        } else if (n == 2) {
            return 2;
        }

        int[] result = new int[n + 1];

        result[1] = 1;
        result[2] = 2;

        for (int i = 3; i <= n; i++) {
            result[i] = result[i - 1] + result[i - 2];
        }

        return result[n];
    }
}
```