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

/*
Say you have an array for which the ith element is the price of a given stock on day i.

        Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the stock multiple times). However, you may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

*/


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 {

        int[] input = {7, 1, 5, 3, 6, 4};

        Solution solution = new Solution();

        int result = solution.maxProfit(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 int maxProfit(int[] prices) {

        int pricesLength = prices.length;

        int result = 0;

        for (int i = 1; i < pricesLength; i++) {

            result += Math.max(prices[i] - prices[i-1], 0);
        }

        return result;
    }
}
```