Zigzag Subarrays

Time Limit	Memory Limit
1 second (See Below)	512 MB

Description

Let A be an integer array of length n where A[i] is the i-th element in A (i = 1, 2, ..., n). For indices i, j with $1 \le i \le j \le n$, let A[i, j] be the subarray of A consisting of (j-i+1) elements from A[i] to A[j] -- this subarray's length is (j-i+1). For instance, if A = [2, 2, 1, 3, 2], then A[2, 3] = [2, 1] and A[3, 5] = [1, 3, 2].

If the elements of a subarray A[i, j] whose length is 2 or more (that is, $i < j$) alternate between increasing/decreasing, then the subarray is called a zigzag subarray -- more formally, one of the following conditions must be met:

- Condition 1: For all k with $i \le k < j$, if (k - i) is even then $A[k] < A[k+1]$ holds and if (k - i) is odd then $A[k] > A[k+1]$ holds.
- Condition 2: For all k with $i \le k < j$, if (k - i) is even then $A[k] > A[k+1]$ holds and if (k - i) is odd then $A[k] < A[k+1]$ holds.

For instance, if A = [2, 2, 1, 3, 2], then A[2, 3] and A[3, 5] are zigzag subarrays whereas A[1, 2] and A[1, 3] are not.

Alice wants to know how many subarrays of A are zigzag subarrays. Let's help Alice.

Input

The first line of the input will contain T, the number of test cases.

Each test case will be given over two lines. The first line will contain n, and the second line will contain n integers, the array's elements, separated by whitespace.

Output

Output each test case's answer (the number of zigzag subarrays) in each line.

Limit

- $1 \le T \le 10$
- $1 \le n \le 100,000$
- $-10^9 \le A[i] \le 10^9$ $(1 \le i \le n)$

Sample Input 1 Copy

```
4
5
2 2 1 3 2
4
2 0 2 2
5
1 2 3 2 1
7
1 2 1 2 1 2 1
```

Sample Output 1 Copy

```
6
3
5
21
```

Case 1: A[2, 3], A[2, 4], A[2, 5], A[3, 4], A[3, 5], and A[4, 5] are zigzag subarrays.

Case 2: A[1, 2], A[1, 3], and A[2, 3] are zigzag subarrays.

Case 3: No explanation.

Case 4: All subarrays of length 2 or more are zigzag subarrays.

Time Limit

- Java 8: 2 seconds
- Python 3: 1 second
- PyPy3: 1 second
- Java 8 (OpenJDK): 2 seconds
- Java 11: 2 seconds
- Kotlin (JVM): 2 seconds
- Java 15: 2 seconds