**Sum of All Odd Length Subarrays**

**Easy**

Given an array of positive integers arr, calculate the sum of all possible odd-length subarrays.

A subarray is a contiguous subsequence of the array.

Return *the sum of all odd-length subarrays of*arr.

**Example 1:**

**Input:** arr = [1,4,2,5,3]

**Output:** 58

**Explanation:** The odd-length subarrays of arr and their sums are:

[1] = 1

[4] = 4

[2] = 2

[5] = 5

[3] = 3

[1,4,2] = 7

[4,2,5] = 11

[2,5,3] = 10

[1,4,2,5,3] = 15

If we add all these together we get 1 + 4 + 2 + 5 + 3 + 7 + 11 + 10 + 15 = 58

**Example 2:**

**Input:** arr = [1,2]

**Output:** 3

**Explanation:** There are only 2 subarrays of odd length, [1] and [2]. Their sum is 3.

**Example 3:**

**Input:** arr = [10,11,12]

**Output:** 66

**Constraints:**

* 1 <= arr.length <= 100
* 1 <= arr[i] <= 1000
* class Solution {
* public:
* int sumOddLengthSubarrays(vector<int>& arr) {
* int sum=0, n=arr.size();
* for (int i=0; i<n; i++) {
* int ev1, ev2;
* if ((i+1)%2==0) ev1=(i+1)/2;
* else ev1=(i+1)/2+1;
* int od1=i+1-ev1;
* if ((n-i)%2==0) ev2=(n-i)/2;
* else ev2=(n-i)/2+1;
* int od2=n-i-ev2;
* sum+=(arr[i]\*(ev1\*ev2)+arr[i]\*(od1\*od2));
* }
* return sum;
* }
* };