3077. Maximum Strength of K Disjoint Subarrays

Description

You are given a **0-indexed** array of integers `nums` of length `n`, and a **positive** **odd** integer `k`.

The strength of `x` subarrays is defined as `strength = sum[1] * x - sum[2] * (x - 1) + sum[3] * (x - 2) - sum[4] * (x - 3) + ... + sum[x] * 1` where `sum[i]` is the sum of the elements in the `i th` subarray. Formally, strength is sum of `(-1) i+1 * sum[i] * (x - i + 1)` over all `i` 's such that `1 <= i <= x`.

You need to select `k` **disjoint** **subarrays** from `nums`, such that their **strength** is **maximum**.

Return *the* ***maximum*** *possible* ***strength*** *that can be obtained*.

Note that the selected subarrays **don't** need to cover the entire array.

Example 1:

```
Input: nums = [1,2,3,-1,2], k = 3
Output: 22
Explanation: The best possible way to select 3 subarrays is: nums[0..2], nums[3..3], and nums[4..4]. The strength is (1 + 2 + 3) * 3 - (-1) * 2 + 2 * 1 = 22.
```

Example 2:

```
Input: nums = [12,-2,-2,-2,-2], k = 5
Output: 64
Explanation: The only possible way to select 5 disjoint subarrays is: nums[0..0], nums[1..1], nums[2..2], nums[3..3], and nums[4..4]. The strength is 12 * 5 - (-2) * 4 + (-2) * 3 - (-2) * 2 + (-2) * 1 = 64.
```

Example 3:

```
Input: nums = [-1,-2,-3], k = 1
Output: -1
Explanation: The best possible way to select 1 subarray is: nums[0..0]. The strength is -1.
```

Constraints:

- `1 <= n <= 10 4`
- `-10 9 <= nums[i] <= 10 9`
- `1 <= k <= n`
- `1 <= n * k <= 10 6`
- `k` is odd.