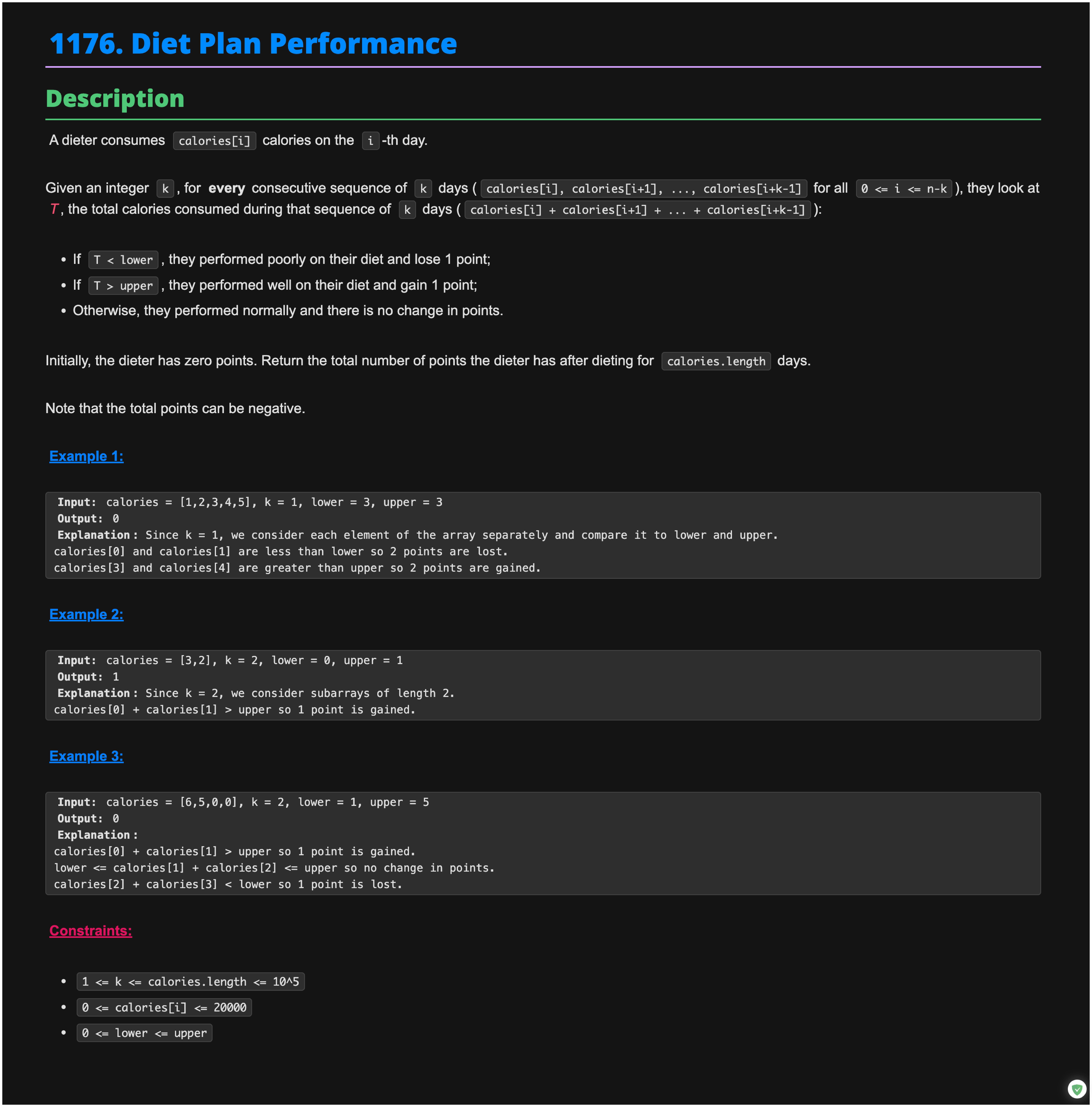

1176. Diet Plan Performance

Description

A dieter consumes `calories[i]` calories on the `i` -th day.

Given an integer `k` , for **every** consecutive sequence of `k` days (`calories[i], calories[i+1], ..., calories[i+k-1]` for all `0 <= i <= n-k`), they look at *T*, the total calories consumed during that sequence of `k` days (`calories[i] + calories[i+1] + ... + calories[i+k-1]`):

- If `T < lower` , they performed poorly on their diet and lose 1 point;
- If `T > upper` , they performed well on their diet and gain 1 point;
- Otherwise, they performed normally and there is no change in points.

Initially, the dieter has zero points. Return the total number of points the dieter has after dieting for `calories.length` days.

Note that the total points can be negative.

Example 1:

```
 Input: calories = [1,2,3,4,5], k = 1, lower = 3, upper = 3
 Output: 0
 Explanation: Since k = 1, we consider each element of the array separately and compare it to lower and upper.
calories[0] and calories[1] are less than lower so 2 points are lost.
calories[3] and calories[4] are greater than upper so 2 points are gained.
```

Example 2:

```
 Input: calories = [3,2], k = 2, lower = 0, upper = 1
 Output: 1
 Explanation: Since k = 2, we consider subarrays of length 2.
calories[0] + calories[1] > upper so 1 point is gained.
```

Example 3:

```
 Input: calories = [6,5,0,0], k = 2, lower = 1, upper = 5
 Output: 0
 Explanation:
calories[0] + calories[1] > upper so 1 point is gained.
lower <= calories[1] + calories[2] <= upper so no change in points.
calories[2] + calories[3] < lower so 1 point is lost.
```

Constraints:

- `1 <= k <= calories.length <= 10^5`
- `0 <= calories[i] <= 20000`
- `0 <= lower <= upper`