5201. Watering Plants

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

You want to water `n` plants in your garden with a watering can. The plants are arranged in a row and are labeled from `0` to `n - 1` from left to right where the i^{th} plant is located at `x = i`. There is a river at `x = -1` that you can refill your watering can at.

Each plant needs a specific amount of water. You will water the plants in the following way:

- Water the plants in order from left to right.
- After watering the current plant, if you do not have enough water to **completely** water the next plant, return to the river to fully refill the watering can.
- You **cannot** refill the watering can early.

You are initially at the river (i.e., `x = -1`). It takes **one step** to move **one unit** on the x-axis.

Given a **0-indexed** integer array `plants` of `n` integers, where `plants[i]` is the amount of water the i^{th} plant needs, and an integer `capacity` representing the watering can capacity, return *the **number of steps** needed to water all the plants*.

Example 1:

```
Input: plants = [2,2,3,3], capacity = 5
Output: 14
Explanation: Start at the river with a full watering can:
- Walk to plant 0 (1 step) and water it. Watering can has 3 units of water.
- Walk to plant 1 (1 step) and water it. Watering can has 1 unit of water.
- Since you cannot completely water plant 2, walk back to the river to refill (2 steps).
- Walk to plant 2 (3 steps) and water it. Watering can has 2 units of water.
- Since you cannot completely water plant 3, walk back to the river to refill (3 steps).
- Walk to plant 3 (4 steps) and water it.
Steps needed = 1 + 1 + 2 + 3 + 3 + 4 = 14.
```

Example 2:

```
Input: plants = [1,1,1,4,2,3], capacity = 4
Output: 30
Explanation: Start at the river with a full watering can:
- Water plants 0, 1, and 2 (3 steps). Return to river (3 steps).
- Water plant 3 (4 steps). Return to river (4 steps).
- Water plant 4 (5 steps). Return to river (5 steps).
- Water plant 5 (6 steps).
Steps needed = 3 + 3 + 4 + 4 + 5 + 5 + 6 = 30.
```

Example 3:

```
Input: plants = [7,7,7,7,7,7,7], capacity = 8
Output: 49
Explanation: You have to refill before watering each plant.
Steps needed = 1 + 1 + 2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 + 7 = 49.
```

Constraints:

- `n == plants.length`
- `1 <= n <= 1000`
- `1 <= plants[i] <= 10^6`
- `max(plants[i]) <= capacity <= 10^9`

JavaScript

```javascript
const wateringPlants = (a, capacity) => {
    let n = a.length, res = 0, cur = capacity;
    for (let i = 0; i < n; i++) {
        let add;
        if (cur - a[i] >= 0) {
            cur -= a[i];
            add = 1;
        } else {
            cur = capacity;
            add = (i + 1) * 2 - 1;
            cur -= a[i];
        }
        res += add;
        // pr(a[i], "res", res, "cur", cur, "add", add);
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/590267481/) | More Details > (/submissions/detail/590267481/)

Share your acceptance!