

<https://course.acciojob.com/idle?question=33eb177f-4ed4-4fc4-ac6b-38e8939e3297>

● EASY

● Max Score: 30 Points

●

Watering Plants

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.

Input Format

The first line of the input contains the number n (total number of plants) and capacity k of the container.

The next n integers denotes the water required by each plant.

Output Format

Print the number of steps required to water all the plants

Example 1

Input

```
4 5
2 2 3 3
```

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

```
6 4
1 1 1 4 2 3
```

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.

Constraints

```
n == plants.length
1 <= n <= 1000
1 <= plants[i] <= 106
max(plants[i]) <= capacity <= 109
```

Topic Tags

- **Math**

My code

// in java

```
import java.util.*;
```

```
import java.io.*;
```

```
class Solution{
```

```
    public static int waterPlants(int n, int k, int a[]){
```

```
        // Write code here
```

```
        int t=k;
```

```
        int ans=0;
```

```
        int c=0;//ditane b/w plant to riv
```

```
        for(int i=0;i<n;i++)
```

```
        {
```

```
            if(t<=a[i])
```

```
            {
```

```
                ans=ans+c+c;
```

```

        t=k;
    }
    c++;
    ans++;
    t=t-a[i];
}
return ans;
}
}

public class Main {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt(), k = input.nextInt();
        int a[] = new int[n];
        for(int i = 0; i < n; i++){
            a[i] = input.nextInt();
        }
        System.out.println(Solution.waterPlants(n, k, a));
    }
}

```