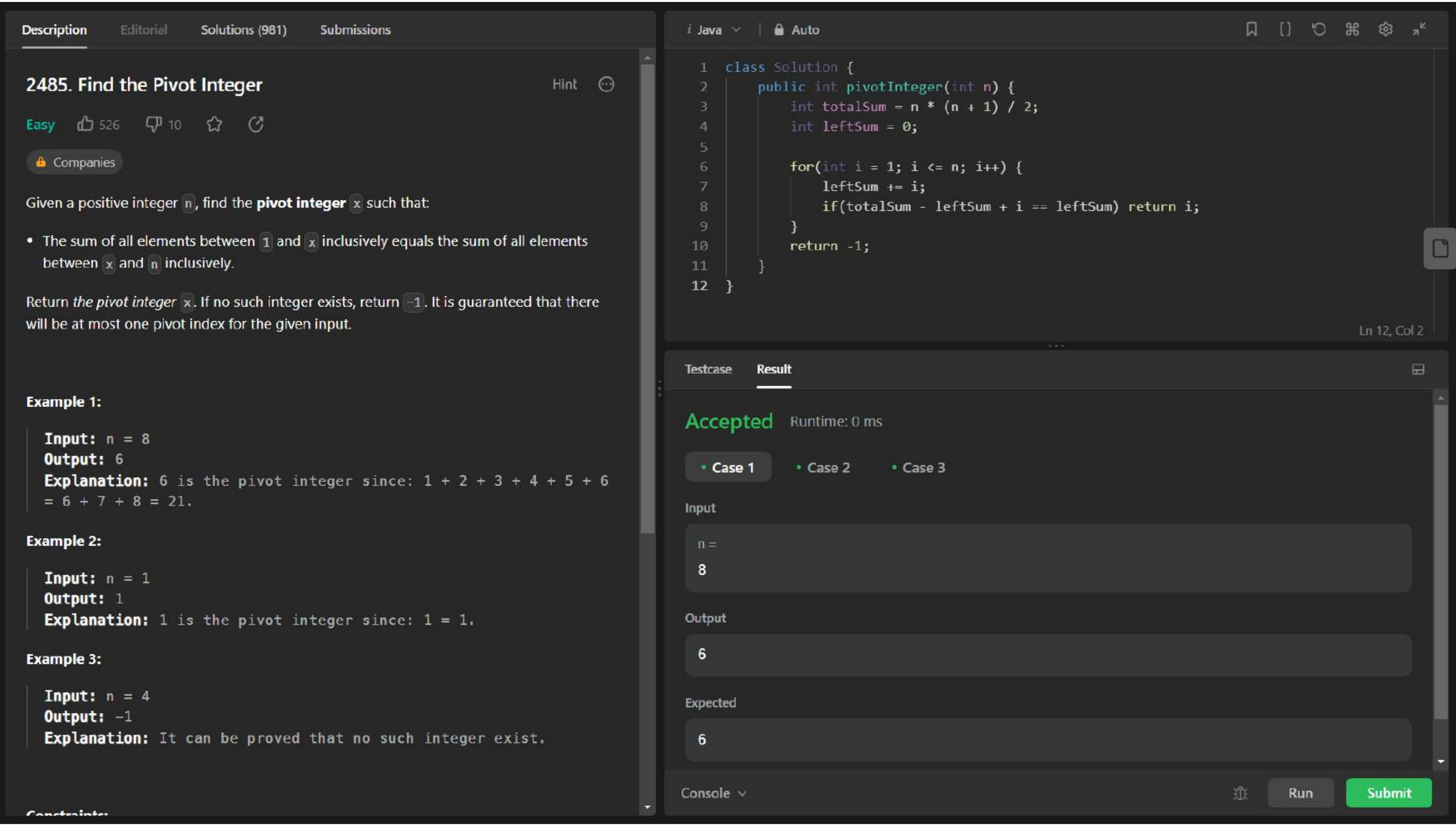


2485. Find the Pivot Integer

Given a positive integer `n`, find the **pivot integer** `x` such that:

- The sum of all elements between `1` and `x` inclusively equals the sum of all elements between `x` and `n` inclusively.

Return *the pivot integer* `x`. If no such integer exists, return `-1`. It is guaranteed that there will be at most one pivot index for the given input.

Example 1:

```
Input: n = 8
Output: 6
Explanation: 6 is the pivot integer since: 1 + 2 + 3 + 4 + 5 + 6 = 6 + 7 + 8 = 21.
```

Example 2:

```
Input: n = 1
Output: 1
Explanation: 1 is the pivot integer since: 1 = 1.
```

Example 3:

```
Input: n = 4
Output: -1
Explanation: It can be proved that no such integer exist.
```

Constraints:

```java
class Solution {
    public int pivotInteger(int n) {
        int totalSum = n * (n + 1) / 2;
        int leftSum = 0;

        for(int i = 1; i <= n; i++) {
            leftSum += i;
            if(totalSum - leftSum + i == leftSum) return i;
        }
        return -1;
    }
}
```

Accepted Runtime: 0 ms

Input

n =

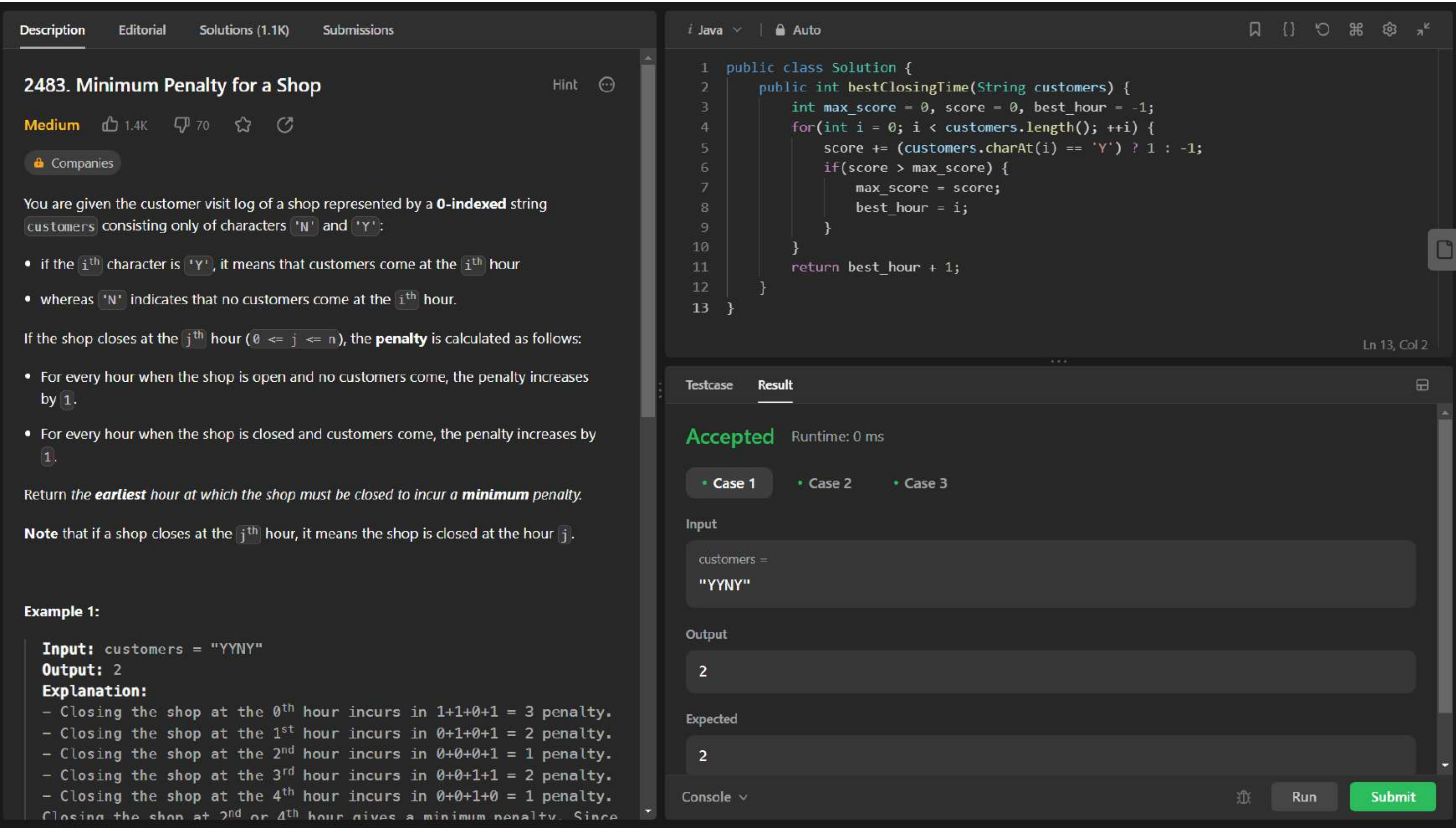
8

Output

6

Expected

6



2483. Minimum Penalty for a Shop

You are given the customer visit log of a shop represented by a **0-indexed** string `customers` consisting only of characters `'N'` and `'Y'`:

- if the i^{th} character is `'Y'`, it means that customers come at the i^{th} hour
- whereas `'N'` indicates that no customers come at the i^{th} hour.

If the shop closes at the j^{th} hour (`0 <= j <= n`), the **penalty** is calculated as follows:

- For every hour when the shop is open and no customers come, the penalty increases by `1`.
- For every hour when the shop is closed and customers come, the penalty increases by `1`.

Return *the **earliest** hour at which the shop must be closed to incur a **minimum** penalty.*

Note that if a shop closes at the j^{th} hour, it means the shop is closed at the hour `j`.

Example 1:

```
Input: customers = "YYNY"
Output: 2
Explanation:
- Closing the shop at the 0th hour incurs in 1+1+0+1 = 3 penalty.
- Closing the shop at the 1st hour incurs in 0+1+0+1 = 2 penalty.
- Closing the shop at the 2nd hour incurs in 0+0+0+1 = 1 penalty.
- Closing the shop at the 3rd hour incurs in 0+0+1+1 = 2 penalty.
- Closing the shop at the 4th hour incurs in 0+0+1+0 = 1 penalty.
Closing the shop at 2nd or 4th hour gives a minimum penalty. Since
```

```java
public class Solution {
    public int bestClosingTime(String customers) {
        int max_score = 0, score = 0, best_hour = -1;
        for(int i = 0; i < customers.length(); ++i) {
            score += (customers.charAt(i) == 'Y') ? 1 : -1;
            if(score > max_score) {
                max_score = score;
                best_hour = i;
            }
        }
        return best_hour + 1;
    }
}
```

Accepted Runtime: 0 ms

Input

customers =

"YYNY"

Output

2

Expected

2