

## 2483. Minimum Penalty for a Shop

Solved 

Medium

 Topics

 Companies

 Hint

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

If the shop closes at the  $j^{\text{th}}$  hour ( $0 \leq j \leq n$ ), 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  $j^{\text{th}}$  hour, it means the shop is closed at the hour `j`.

### Example 1:

**Input:** customers = "YNYN"

**Output:** 2

**Explanation:**

- Closing the shop at the 0<sup>th</sup> hour incurs in  $1+1+0+1 = 3$  penalty.
  - Closing the shop at the 1<sup>st</sup> hour incurs in  $0+1+0+1 = 2$  penalty.
  - Closing the shop at the 2<sup>nd</sup> hour incurs in  $0+0+0+1 = 1$  penalty.
  - Closing the shop at the 3<sup>rd</sup> hour incurs in  $0+0+1+1 = 2$  penalty.
  - Closing the shop at the 4<sup>th</sup> hour incurs in  $0+0+1+0 = 1$  penalty.
- Closing the shop at 2<sup>nd</sup> or 4<sup>th</sup> hour gives a minimum penalty. Since 2 is earlier, the optimal closing time is 2.

### Example 2:

**Input:** customers = "NNNNN"

**Output:** 0

**Explanation:** It is best to close the shop at the 0<sup>th</sup> hour as no customers arrive.

### Example 3:

**Input:** customers = "YYYY"

**Output:** 4

**Explanation:** It is best to close the shop at the 4<sup>th</sup> hour as customers arrive at each hour.

### Constraints:

- $1 \leq \text{customers.length} \leq 10^5$
- `customers` consists only of characters 'Y' and 'N'.

## Python:

class Solution:

def bestClosingTime(self, customers: str) -> int:

max\_score = score = best\_hour = 0

#peak = running score = max\_score at hour

for idx, char in enumerate(customers):

score += 1 if char == 'Y' else -1

if score > max\_score: # new peak?

```
        max_score, best_hour = score, idx + 1
```

```
    return best_hour
```

## JavaScript:

```
/**
 * @param {string} customers
 * @return {number}
 */
var bestClosingTime = function(customers) {
    let max_score = 0, score = 0, best_hour = -1;
    for(let i = 0; i < customers.length; ++i) {
        score += (customers[i] === 'Y') ? 1 : -1;
        if(score > max_score) {
            max_score = score;
            best_hour = i;
        }
    }
    return best_hour + 1;
};
```

## Java:

```
class Solution {
    public int bestClosingTime(String customers) {
        int maxScore = 0;
        int score = 0;
        int bestHour = 0;

        for (int i = 0; i < customers.length(); i++) {
            score += (customers.charAt(i) == 'Y') ? 1 : -1;

            if (score > maxScore) {
                maxScore = score;
                bestHour = i + 1;
            }
        }
        return bestHour;
    }
}
```