# Problem 3168: Minimum Number of Chairs in a Waiting Room

## Problem Information

**Difficulty:** Easy
**Acceptance Rate:** 78.92%
**Paid Only:** No
**Tags:** String, Simulation

## Problem Description

You are given a string `s`. Simulate events at each second `i`:

* If `s[i] == 'E'`, a person enters the waiting room and takes one of the chairs in it. * If `s[i] == 'L'`, a person leaves the waiting room, freeing up a chair.

Return the **minimum** number of chairs needed so that a chair is available for every person who enters the waiting room given that it is initially **empty**.

**Example 1:**

**Input:** s = "EEEEEEE"

**Output:** 7

**Explanation:**

After each second, a person enters the waiting room and no person leaves it. Therefore, a minimum of 7 chairs is needed.

**Example 2:**

**Input:** s = "ELELEEL"

**Output:** 2

**Explanation:**

Let's consider that there are 2 chairs in the waiting room. The table below shows the state of the waiting room at each second.

Second | Event | People in the Waiting Room | Available Chairs ---|---|---|--- 0 | Enter | 1 | 1 1 | Leave | 0 | 2 2 | Enter | 1 | 1 3 | Leave | 0 | 2 4 | Enter | 1 | 1 5 | Enter | 2 | 0 6 | Leave | 1 | 1
**Example 3:**

**Input:** s = "ELEELEELLL"

**Output:** 3

**Explanation:**

Let's consider that there are 3 chairs in the waiting room. The table below shows the state of the waiting room at each second.

Second | Event | People in the Waiting Room | Available Chairs ---|---|---|--- 0 | Enter | 1 | 2 1 | Leave | 0 | 3 2 | Enter | 1 | 2 3 | Enter | 2 | 1 4 | Leave | 1 | 2 5 | Enter | 2 | 1 6 | Enter | 3 | 0 7 | Leave | 2 | 1 8 | Leave | 1 | 2 9 | Leave | 0 | 3

**Constraints:**

* `1 <= s.length <= 50` * `s` consists only of the letters `'E'` and `'L'`. * `s` represents a valid sequence of entries and exits.

## Code Snippets

**C++:**

```
class Solution {
public:
int minimumChairs(string s) {

}
};
```

**Java:**

```
class Solution {
public int minimumChairs(String s) {

}
}
```

**Python3:**

```
class Solution:
def minimumChairs(self, s: str) -> int:
```