

## 2211. Count Collisions on a Road

Solved 

Medium

 Topics

 Companies

 Hint

There are  $n$  cars on an infinitely long road. The cars are numbered from  $0$  to  $n - 1$  from left to right and each car is present at a **unique** point.

You are given a **0-indexed** string `directions` of length  $n$ . `directions[i]` can be either `'L'`, `'R'`, or `'S'` denoting whether the  $i^{\text{th}}$  car is moving towards the **left**, towards the **right**, or **staying** at its current point respectively. Each moving car has the **same speed**.

The number of collisions can be calculated as follows:

- When two cars moving in **opposite** directions collide with each other, the number of collisions increases by  $2$ .
- When a moving car collides with a stationary car, the number of collisions increases by  $1$ .

After a collision, the cars involved can no longer move and will stay at the point where they collided. Other than that, cars cannot change their state or direction of motion.

Return *the **total number of collisions** that will happen on the road.*

### Example 1:

**Input:** directions = "RLRSLL"

**Output:** 5

#### Explanation:

The collisions that will happen on the road are:

- Cars 0 and 1 will collide with each other. Since they are moving in opposite directions, the number of collisions becomes  $0 + 2 = 2$ .
- Cars 2 and 3 will collide with each other. Since car 3 is stationary, the number of collisions becomes  $2 + 1 = 3$ .
- Cars 3 and 4 will collide with each other. Since car 3 is stationary, the number of collisions becomes  $3 + 1 = 4$ .
- Cars 4 and 5 will collide with each other. After car 4 collides with car 3, it will stay at the point of collision and get hit by car 5. The number of collisions becomes  $4 + 1 = 5$ .

Thus, the total number of collisions that will happen on the road is 5.

### Example 2:

**Input:** directions = "LLRR"

**Output:** 0

#### Explanation:

No cars will collide with each other. Thus, the total number of collisions that will happen on the road is 0.

### Constraints:

- $1 \leq \text{directions.length} \leq 10^5$
- `directions[i]` is either 'L', 'R', or 'S'.

## Python:

class Solution:

```
def countCollisions(self, directions: str) -> int:
    i, j = 0, len(directions) - 1
    while i < len(directions) and directions[i] == 'L':
        i += 1
    while j >= 0 and directions[j] == 'R':
        j -= 1
    return sum(directions[k] != 'S' for k in range(i, j + 1))
```

## JavaScript:

/\*\*

\* @param {string} directions

```

* @return {number}
*/
var countCollisions = function(directions) {
    let i = 0, j = directions.length - 1;
    while (i < directions.length && directions[i] === 'L') i++;
    while (j >= 0 && directions[j] === 'R') j--;
    let count = 0;
    for (let k = i; k <= j; k++)
        if (directions[k] !== 'S') count++;
    return count;
};

```

## Java:

```

class Solution {
    public int countCollisions(String dir) {

        int res = 0, n = dir.length(), i = 0, carsFromRight = 0;

        while (i < n && dir.charAt(i) == 'L') i++;

        for ( ; i < n; i++) {
            if (dir.charAt(i) == 'R') carsFromRight++;
            else {
                res += (dir.charAt(i) == 'S') ? carsFromRight : carsFromRight+1;
                carsFromRight = 0;
            }
        }
        return res;
    }
}

```