2379. Minimum Recolors to Get K Consecutive Black Blocks

You are given a **0-indexed** string `blocks` of length `n`, where `blocks[i]` is either `'W'` or `'B'`, representing the color of the `ith` block. The characters `'W'` and `'B'` denote the colors white and black, respectively.

You are also given an integer `k`, which is the desired number of **consecutive** black blocks.

In one operation, you can **recolor** a white block such that it becomes a black block.

Return *the **minimum** number of operations needed such that there is at least **one** occurrence of* `k` *consecutive black blocks.*

Example 1:

```
Input: blocks = "WBBWWBBWBW", k = 7
Output: 3
Explanation:
One way to achieve 7 consecutive black blocks is to recolor the 0th, 3rd, and 4th
blocks
so that blocks = "BBBBBBBWBW".
It can be shown that there is no way to achieve 7 consecutive black blocks in less
than 3 operations.
Therefore, we return 3.
```

Example 2:

```
Input: blocks = "WBWBBBW", k = 2
Output: 0
Explanation:
No changes need to be made, since 2 consecutive black blocks already exist.
Therefore, we return 0.
```

Constraints:

- `n == blocks.length`
- `1 <= n <= 100`
- `blocks[i]` is either `'W'` or `'B'`.
- `1 <= k <= n`

Overview

We are given a string `blocks`, where each character represents a block that is either black ('B') or white ('W') and the ability to apply an operation to change a white block black an unlimited number of times. Our goal is to find the **minimum number of recoloring operations** needed to create a segment of `k` consecutive black blocks.

2379. Minimum Recolors to Get K Consecutive Black Blocks

```
/*
    Naive approach: try all windows
    Time complexity: $O\left(n^{2}\right)$
    Space complexity: O(1)
*/
class Solution {
    public:
        int minimumRecolors(string blocks, int k) {
            int n=blocks.size();
            int ans=INT_MAX;
            // For each window starting from i
            for(int i=0;i<=n-k;++i){
                int cnt_white = 0; // Initialize the number of White blocks
                // Take a window of size k ( (k+i-1)-i+1 )
                for(int j=i;j<k+i;++j){
                    cnt_white+=(blocks[j]=='W'); // Update the number of White blocks
                }

                // Minimize the answer
                ans=std::min(ans,cnt_white);
            }

            return ans;
        }
};
```

2379. Minimum Recolors to Get K Consecutive Black Blocks

```
/*
    Static Sliding window
    Time complexity: O(k+n)=O(2n)=O(n)
    Space complexity: O(1)
*/
class Solution {
    public:
        int minimumRecolors(std::string blocks, int k) {
            int n=blocks.size();

            // Create a window of size k
            // This window will always contain the count of White blocks
            int win=0;
            for(int i=0;i<k;++i) win+=(blocks[i]=='W');

            int ans=INT_MAX;

            // Slide window to the left
            for(int i=0;i<=n-k;++i){
                // Minimize the answer
                ans=std::min(ans,win);

                // Update the White blocks count in the current window
                win-=(blocks[i]=='W');
                win+=(blocks[i+k]=='W');
            }

            return ans;
        }
};
```