Problem Statement

You are given a sequence of N tiles, each of which is either Red or Green.

You are also provided with an integer K, which indicates the number of consecutive Red tiles you need.

In each operation, you can paint one tile Red, using one box of paint per tile.

Your task is to determine the minimum number of paint boxes required to achieve at least K consecutive Red tiles in the given sequence.

Input Format

The first line contains two space separated integers, N represent the total number of tiles and K reppresent the number of consecutive Red tiles required.

The second line contains a string of length N where each character is either 'R' or 'G'.

Output Format

Print a single integer representing the minimum number of paint boxes needed.

Constraints

1 <= N <= 10^4

1 <= K <= 10^4

Sample Testcase 0

Testcase Input

10 7 GRRGGRRGRG

Testcase Output

3

Explanation

One way to achieve 7 consecutive Red tiles is to recolor the 0th, 3rd, and 4th tiles so that it becomes " RRRRRRRGRG. " It can be shown that there is no way to achieve 7 consecutive black blocks in less than 3 operations. Therefore, we get 3.

Sample Testcase 1

Testcase Input

5 3 RRRGR

Testcase Output

0

Explanation

The input sequence "RRRGR" already contains a segment of 3 consecutive Red tiles. Since the required number of consecutive Red tiles is 3 and this condition is already met, no additional repainting is needed. Therefore, the minimum number of paint boxes required is 0.

ANSWER:

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

class Main {

    public static void main(String[] args) {

        /\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/

       Scanner sc = new Scanner(System.in);

        // Read input values

        int n = sc.nextInt(); // Length of the string

        int k = sc.nextInt(); // Number of consecutive 'R' tiles required

        sc.nextLine(); // Consume newline character

        String str = sc.nextLine(); // The string of tiles

        // Variables to track the minimum number of operations

        int minOperations = Integer.MAX\_VALUE;

        // Loop through possible starting points for a window of size k

        for (int i = 0; i <= n - k; i++) {

            // Count number of 'G's in the current window of size k

            int countG = 0;

            for (int j = i; j < i + k; j++) {

                if (str.charAt(j) == 'G') {

                    countG++;

                }

            }

            // Update minimum operations needed

            minOperations = Math.min(minOperations, countG);

        }

        // Print the result

        System.out.println(minOperations);

    }

}