B. Chess Cheater

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You like playing chess tournaments online.

In your last tournament you played n games. For the sake of this problem, each chess game is either won or lost (no draws). When you lose a game you get 0 points. When you win you get 1 or 2 points: if you have won also the previous game you get 2 points, otherwise you get 1 point. If you win the very first game of the tournament you get 1 point (since there is not a "previous game").

The outcomes of the n games are represented by a string s of length n: the i-th character of s is `W` if you have won the i-th game, while it is `L` if you have lost the i-th game.

After the tournament, you notice a bug on the website that allows you to change the outcome of **at most** k of your games (meaning that at most k times you can change some symbol `L` to `W`, or `W` to `L`). Since your only goal is to improve your chess rating, you decide to cheat and use the bug.

Compute the maximum score you can get by cheating in the optimal way.

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 20,000$) — the number of test cases. The description of the test cases follows.

The first line of each testcase contains two integers n, k ($1 \le n \le 100,000$, $0 \le k \le n$) – the number of games played and the number of outcomes that you can change.

The second line contains a string s of length n containing only the characters `W` and `L`. If you have won the i-th game then $s_i =$ `W`, if you have lost the i-th game then $s_i =$ `L`.

It is guaranteed that the sum of n over all testcases does not exceed $200,000$.

Output

For each testcase, print a single integer – the maximum score you can get by cheating in the optimal way.

Example

input Copy

```
8
5 2
WLWLL
6 5
LLLWWL
7 1
LWLWLWL
15 5
WWWLLLWWWLLLWWW
40 7
LLWLWLWWWLWLLWLWWWLWLLWLLWLLLLWLLWWWLWWL
1 0
L
1 1
L
6 1
WLLWLW
```

output Copy

```
7
11
6
26
46
0
1
6
```

→ **Problem tags**

greedy implementation sortings *1400

No tag edit access

→ **Contest materials**

- Announcement (en)
- Tutorial (en)

Note

Explanation of the first testcase. Before changing any outcome, the score is 2. Indeed, you won the first game, so you got 1 point, and you won also the third, so you got another 1 point (and not 2 because you lost the second game).

An optimal way to cheat is to change the outcomes of the second and fourth game. Doing so, you end up winning the first four games (the string of the outcomes becomes `WWWWL`). Hence, the new score is $7 = 1 + 2 + 2 + 2$: 1 point for the first game and 2 points for the second, third and fourth game.

Explanation of the second testcase. Before changing any outcome, the score is 3. Indeed, you won the fourth game, so you got 1 point, and you won also the fifth game, so you got 2 more points (since you won also the previous game).

An optimal way to cheat is to change the outcomes of the first, second, third and sixth game. Doing so, you end up winning all games (the string of the outcomes becomes `WWWWWW`). Hence, the new score is $11 = 1 + 2 + 2 + 2 + 2 + 2$: 1 point for the first game and 2 points for all the other games.
