2009 Canadian Computing Competition
Day 1, Question 2
Problem B – Dinner

Input: from standard input
Output: to standard output
Source file: dinner.{c, cpp, pas}

On the way to dinner, the CCC competitors are lining up for their delicious curly fries. The N ($1 \leq N \leq 100$) competitors have lined up single-file to enter the cafeteria.

Doctor V, who runs the CCC, realized at the last minute that programmers simply hate standing in line next to programmers who use a different language. Thankfully, only two languages are allowed at the CCC: Gnold and Helpfile. Furthermore, the competitors have decided that they will only enter the cafeteria if they are in a group of at least K ($1 <= K <= 6$) competitors.

Doctor V decided to iterate the following scheme:

- He will find a group of K or more competitors who use the same language standing next to each other in line and send them to dinner.
- The remaining competitors will close the gap, potentially putting similar-language competitors together

So Doctor V recorded the sequence of competitors for you. Can all the competitors dine? If so, what is the minimum number of groups of competitors to be sent to dinner?

Note: Test cases worth 60% of the points have $K \leq 2$. Out of these, on test cases worth one third of the points (20% of the total points), $N \leq 10$.

Input Specification

The first line contains two integers N and K.

The second line contains N characters that are the sequence of competitors in line (H represents Helpfile, G represents Gnold)

Sample Input

```
7 2
GHHGHHG
```

Description of Sample Input

There are seven competitors: a Gnold programmer followed by two Helpfile programmers, followed by another Gnold programmer, followed by another two Helpfile programmers followed by a final Gnold programmer. Programmers want to goto dinner in pairs.

Output Specification

Output, on one line, the single number that is the minimum number of groups that are formed for dinner. If not all programmers can dine, output −1.

Output for Sample Input

3

Description of Output for Sample Input

First send the first pair of *H*s to dinner, leaving *GGHHG*. Then send the second pair of *H*s to dinner, leaving *GGG*; finally, send in the group of *G*s. It might be coincidental that the two pairs of Helpfile programmers entered the cafeteria successively.