

## 482. License Key Formatting

[My Submissions \(/contest/leetcode-weekly-contest-14/problems/license-key-formatting/submissions/\)](/contest/leetcode-weekly-contest-14/problems/license-key-formatting/submissions/)[Back To Contest \(/contest/leetcode-weekly-contest-14/\)](/contest/leetcode-weekly-contest-14/)User Accepted: **471**   User Tried: **485**   Total Accepted: **484**   Total Submissions: **1067**   Difficulty: **Medium**

Now you are given a string S, which represents a software license key which we would like to format. The string S is composed of alphanumeric characters and dashes. The dashes split the alphanumeric characters within the string into groups. (i.e. if there are M dashes, the string is split into M+1 groups). The dashes in the given string are possibly misplaced.

We want each group of characters to be of length K (except for possibly the first group, which could be shorter, but still must contain at least one character). To satisfy this requirement, we will reinsert dashes. Additionally, all the lower case letters in the string must be converted to upper case.

So, you are given a non-empty string S, representing a license key to format, and an integer K. And you need to return the license key formatted according to the description above.

### Example 1:

**Input:** S = "2-4A0r7-4k", K = 4**Output:** "24A0-R74K"**Explanation:** The string S has been split into two parts, each part has 4 characters.

### Example 2:

**Input:** S = "2-4A0r7-4k", K = 3**Output:** "24-A0R-74K"**Explanation:** The string S has been split into three parts, each part has 3 characters except the first part as it could be shorter as said above.

### Note:

1. The length of string S will not exceed 12,000, and K is a positive integer.
2. String S consists only of alphanumeric characters (a-z and/or A-Z and/or 0-9) and dashes(-).
3. String S is non-empty.

[Discuss \(https://discuss.leetcode.com/category/619\)](https://discuss.leetcode.com/category/619)

Python



```
1 class Solution(object):
2     def licenseKeyFormatting(self, S, K):
3         """
4         :type S: str
5         :type K: int
6         :rtype: str
7         """
8         import re
9         S=re.sub('-', '', S)
10        S=S.upper()
11        for i in range(len(S)-K,0,-K):
12            S=S[:i]+'-'+S[i:]
13        return S
```

Custom Testcase ☐ Shortcut: Ctrl + enter

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)