

## License Key Formatting

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 alphanumerical characters and dashes. The dashes split the alphanumerical 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 alphanumerical characters (a-z and/or A-Z and/or 0-9) and dashes(-).
3. String  $S$  is non-empty.

## Solution 1

```
public String licenseKeyFormatting(String s, int k) {  
    StringBuilder sb = new StringBuilder();  
    for (int i = s.length() - 1; i >= 0; i--)  
        if (s.charAt(i) != '-')  
            sb.append(sb.length() % (k + 1) == k ? '-' : '').append(s.charAt(  
i));  
    return sb.reverse().toString().toUpperCase();  
}
```

written by [yuxiangmusic](#) original link [here](#)

## Solution 2

```
class Solution(object):
    def licenseKeyFormatting(self, S, K):
        """
        :type S: str
        :type K: int
        :rtype: str
        """
        S = S.upper().replace('-', '')
        size = len(S)
        s1 = K if size%K==0 else size%K
        res = S[:s1]
        while s1<size:
            res += '-' + S[s1:s1+K]
            s1 += K
        return res
```

written by [yujiao03@gmail.com](mailto:yujiao03@gmail.com) original link [here](#)

## Solution 3

```
public class Solution {
    public String licenseKeyFormatting(String S, int K) {
        String[] strs = S.split("-");
        StringBuilder rst = new StringBuilder();
        int gap = 'A' - 'a';

        for (String str : strs) {
            rst.append(str);
        }

        int len = rst.length();

        for (int i = 0; i < len; i++) {
            char c = rst.charAt(i);
            if (c >= 'a' && c <= 'z') {
                rst.setCharAt(i, (char)(c+gap));
            }
        }

        for (int i = len-K; i > 0; i -= K) {
            rst.insert(i, '-');
        }
        return rst.toString();
    }
}
```

written by [zjay](#) original link [here](#)

From [LeetCoder](#).