482. License Key Formatting

Add to List ▾

Question | Editorial Solution | My Submissions (/problems/license-key-formatting/submissions/)

Total Accepted: **4342** Total Submissions: **10094** Difficulty: **Medium** Contributors: **aizj_Forever** (/aizj_forever/)

Notes

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 sai
```

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.

Hide Company Tags | Google (/company/google/)

Have you met this question in a real interview? Yes | No

Discuss (https://discuss.leetcode.com/category/619) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```
class Solution {
public:

    int countAlphaNum(string& s) {
        int cnt = 0;
        for(auto c : s) {
            cnt += c != '-' ? 1 : 0;
        }
        return cnt;
    }

    string licenseKeyFormatting(string S, int K) {
        if(K <= 0) return "";
        int cnt = countAlphaNum(S);
        if(cnt == 0) return "";
        stringstream ss;
        int firstGroup = cnt % K;
        int j = (K - firstGroup) % K;
        for (int i = 0; i < S.length(); i++) {
            if (j == K) {
                ss << '-';
                j = 0;
```

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

```
            }
            if (S[i] == '-') continue;
            else if (S[i] >= 'a' && S[i] <= 'z') {
                ss << char(S[i] - ('a' - 'A'));
            } else {
                ss << S[i];
            }
            j++;
        }
        string res = ss.str();
        while (res.back() == '-') res.pop_back();
        return res;
    }
};
```

☐ Notes

Custom Testcase ☑

Contribute Testcase

```
"2-4A0r7-4k"
4
```

Shortcut: Command + enter

One line for one parameter. Hint

Run Code | Submit Solution

Submission Result: Accepted (/submissions/detail/90107544/)

More Details (/submissions/detail/90107544/)

Next challenges: (M) Find Permutation (/problems/find-permutation) (M) Increasing Subsequences (/problems/increasing-subsequences) (M) Target Sum (/problems/target-sum)

Share your acceptance!

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

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