

给定一个密钥字符串S，只包含字母，数字以及 '-'（破折号）。N 个 '-' 将字符串分成了 N+1 组。给定一个数字 K，重新格式化字符串，除了第一个分组以外，每个分组要包含 K 个字符，第一个分组至少包含 1 个字符。两个分组之间用 '-'（破折号）隔开，并且将所有的小写字母转换为大写字母。

给定非空字符串 S 和数字 K，按照上面描述的规则进行格式化。

示例 1:

输入: S = "5F3Z-2e-9-w", K = 4

输出: "5F3Z-2E9W"

解释: 字符串 S 被分成了两个部分，每部分 4 个字符；
注意，两个额外的破折号需要删掉。

示例 2:

输入: S = "2-5g-3-J", K = 2

输出: "2-5G-3J"

解释: 字符串 S 被分成了 3 个部分，按照前面的规则描述，第一部分的字符可以少于给定的数量，其余部分皆为 2 个字符。

提示:

1. S 的长度不超过 12,000，K 为正整数
2. S 只包含字母数字 (a-z, A-Z, 0-9) 以及破折号 '-'
3. S 非空

思路，因为只限制了头可以随意，那么从后往前进行K等分切割，头取余即可，然后需注意，若是头的“-”符号需要清理掉。

```
class Solution {
public:
    string licenseKeyFormatting(string S, int K) {
        transform(S.begin(), S.end(), S.begin(), ::toupper);
        string result="";
        int num=0;
        for(int i=S.size()-1;i>=0;i--){
```

```
    if(S[i]=='-'){
        continue;
    }else{
        result=S[i]+result;
        num++;
        if(num==K){
            result="-"+result;
            num=0;
        }
    }
}
if(result[0]=='-'){
    result =result.substr(1,result.size()-1);
}
return result;
}
};
```