

密钥格式化

有一个密钥字符串 S ，只包含字母，数字以及 '-'（破折号）。其中， N 个 '-' 将字符串分成了 $N+1$ 组。

给你一个数字 K ，请你重新格式化字符串，使每个分组恰好包含 K 个字符。特别地，第一个分组包含的字符个数必须小于等于 K ，但至少要包含 1 个字符。两个分组之间需要用 '-'（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

```
char * licenseKeyFormatting(char * S, int K)
{
    int len = strlen(S); // 字符串总长度
    int countPZH = 0;
    for (int i = 0; i < len; i++) {
        if (S[i] == '-') {
            countPZH++; // 统计破折号个数
        }
    }
    int countVal = len - countPZH; // 除破折号剩余字符个数
    if (countVal == 0) {
        return ""; // 特殊情况，如果剩余字符为空，则无法分割，返回空字符串
    }
    char* temp = (char*)malloc(countVal + 1);
    memset(temp, 0, countVal + 1); // 初始化数组，存储除破折号剩余字符
    int i = 0;
```

```

int j = 0;
while (S[i] != 0) {
    if (S[i] != '-') {
        if (S[i] >= 'a' && S[i] <= 'z') {
            S[i] -= ('a' - 'A'); //按照题意：需要将小写字母转换为大写字母
        }
        temp[j] = S[i]; //除小写字母外，其余大写字母或数组可直接复制给 temp
        j++;
    }
    i++;
}
//以下两个计算十分关键
int groupNum = countVal / K + (countVal % K != 0); //计算分组数 需要考虑
//是否能够整除分组
int firstGNum = countVal % K + K * !(countVal % K); //计算第一组的字符
//按照题意，第一组字符数为分组余数
int retSize = countVal + groupNum - 1; //初始化最终输出结果数组
char* ret = (char*)malloc(retSize + 1);
memset(ret, 0, retSize + 1);
char* p = ret;
char* q = temp;
if (firstGNum) {
    strncpy(p, q, firstGNum); //率先拷贝第一个分组
    p += firstGNum; //拷贝完成，需要注意，两个指针的移动，否则无法完成后续拷
    q += firstGNum;
    *p = '-'; //拷贝完第一组，末尾添加“-”
    p++;
}
for (int i = 0; i < groupNum - 1; i++) {
    strncat(p, q, K);
    p += K;
    q += K;
    *p = '-'; //拷贝完每一组，末尾添加“-”
    p++;
}
*(p - 1) = 0; //拷贝完，末尾添加“\0”
return ret;
}

```