**题目一：左旋转字符串**

字符串的左旋转操作是把字符串前面的若干个字符转移到字符串的尾部。请定义一个函数实现字符串左旋转操作的功能。比如，输入字符串"abcdefg"和数字2，该函数将返回左旋转两位得到的结果"cdefgab"。

示例 1：

输入: s = "abcdefg", k = 2

输出: "cdefgab"

示例 2：

输入: s = "lrloseumgh", k = 6

输出: "umghlrlose"

限制：

1 <= k < s.length <= 10000

**题目二：数组中的字符串匹配**

给你一个字符串数组 words ，数组中的每个字符串都可以看作是一个单词。请你按 任意 顺序返回 words 中是其他单词的子字符串的所有单词。

如果你可以删除 words[j] 最左侧和/或最右侧的若干字符得到 word[i] ，那么字符串 words[i] 就是 words[j] 的一个子字符串。

示例 1：

输入：words = ["mass","as","hero","superhero"]

输出：["as","hero"]

解释："as" 是 "mass" 的子字符串，"hero" 是 "superhero" 的子字符串。

["hero","as"] 也是有效的答案。

示例 2：

输入：words = ["leetcode","et","code"]

输出：["et","code"]

解释："et" 和 "code" 都是 "leetcode" 的子字符串。

示例 3：

输入：words = ["blue","green","bu"]

输出：[]

提示：

1 <= words.length <= 100

1 <= words[i].length <= 30

words[i] 仅包含小写英文字母。

题目数据 保证 每个 words[i] 都是独一无二的。

**题目三：分割平衡字符串**

在一个「平衡字符串」中，'L' 和 'R' 字符的数量是相同的。

给出一个平衡字符串 s，请你将它分割成尽可能多的平衡字符串。

返回可以通过分割得到的平衡字符串的最大数量。

示例 1：

输入：s = "RLRRLLRLRL"

输出：4

解释：s 可以分割为 "RL", "RRLL", "RL", "RL", 每个子字符串中都包含相同数量的 'L' 和 'R'。

示例 2：

输入：s = "RLLLLRRRLR"

输出：3

解释：s 可以分割为 "RL", "LLLRRR", "LR", 每个子字符串中都包含相同数量的 'L' 和 'R'。

示例 3：

输入：s = "LLLLRRRR"

输出：1

解释：s 只能保持原样 "LLLLRRRR".

提示：

1 <= s.length <= 1000

s[i] = 'L' 或 'R'

分割得到的每个字符串都必须是平衡字符串。

**题目四：重新格式化字符串**

给你一个混合了数字和字母的字符串 s，其中的字母均为小写英文字母。

请你将该字符串重新格式化，使得任意两个相邻字符的类型都不同。也就是说，字母后面应该跟着数字，而数字后面应该跟着字母。

请你返回 重新格式化后 的字符串；如果无法按要求重新格式化，则返回一个 空字符串 。

示例 1：

输入：s = "a0b1c2"

输出："0a1b2c"

解释："0a1b2c" 中任意两个相邻字符的类型都不同。 "a0b1c2", "0a1b2c", "0c2a1b" 也是满足题目要求的答案。

示例 2：

输入：s = "leetcode"

输出：""

解释："leetcode" 中只有字母，所以无法满足重新格式化的条件。

示例 3：

输入：s = "1229857369"

输出：""

解释："1229857369" 中只有数字，所以无法满足重新格式化的条件。

示例 4：

输入：s = "covid2019"

输出："c2o0v1i9d"

示例 5：

输入：s = "ab123"

输出："1a2b3"

提示：

1 <= s.length <= 500

s 仅由小写英文字母和/或数字组成。

**题目五：字符串的排列**

输入一个字符串，打印出该字符串中字符的所有排列。

你可以以任意顺序返回这个字符串数组，但里面不能有重复元素。

示例:

输入：s = "abc"

输出：["abc","acb","bac","bca","cab","cba"]

限制：

1 <= s 的长度 <= 8