# 题目

给你一个字符串s和一个字符串数组words，请你判断s是否为words的 前缀字符串。

字符串s要成为words的前缀字符串，需要满足：s可以由words中的前k（k为正数）个字符串按顺序相连得到，且k不超过words.length。

如果s是words的前缀字符串，返回true；否则，返回false。

示例 1：

输入：s = "iloveleetcode", words = ["i","love","leetcode","apples"]

输出：true

解释：

s 可以由 "i"、"love" 和 "leetcode" 相连得到。

示例 2：

输入：s = "iloveleetcode", words = ["apples","i","love","leetcode"]

输出：false

解释：

数组的前缀相连无法得到 s 。

提示：

1 <= words.length <= 100

1 <= words[i].length <= 20

1 <= s.length <= 1000

words[i] 和 s 仅由小写英文字母组成

# 分析

## 方法一：模拟

思路：

使用一个字符串prefix来逐步构建前缀字符串。它遍历words数组中的每个字符串，将其逐个添加到prefix中。在每次添加后，它检查prefix是否等于s。如果相等，说明找到了前缀字符串，返回true。如果prefix的长度已经超过了s的长度，说明无法再形成前缀字符串，可以提前退出循环并返回false。如果循环结束后仍未找到前缀字符串，则返回false。

请注意，这个实现假设输入的字符串和数组都是有效的，并且符合题目要求的格式。在实际应用中，可能还需要添加一些额外的输入验证和错误处理逻辑。

代码：

class Solution {

public:

bool isPrefixString(string s, vector<string>& words) {

string prefix = "";

for (const string& word : words) {

prefix += word;

if (prefix == s) {

return true; // 找到前缀字符串，返回true

}

if (prefix.length() > s.length()) {

break; // 当前前缀长度已经超过s，无法再形成前缀，提前退出循环

}

}

return false; // 循环结束仍未找到前缀字符串，返回false

}

};