# 题目

给你两个字符串 word1 和 word2 。请你从 word1 开始，通过交替添加字母来合并字符串。如果一个字符串比另一个字符串长，就将多出来的字母追加到合并后字符串的末尾。

返回 合并后的字符串 。

示例 1：

输入：word1 = "abc", word2 = "pqr"

输出："apbqcr"

解释：字符串合并情况如下所示：

word1： a b c

word2： p q r

合并后： a p b q c r

示例 2：

输入：word1 = "ab", word2 = "pqrs"

输出："apbqrs"

解释：注意，word2 比 word1 长，"rs" 需要追加到合并后字符串的末尾。

word1： a b

word2： p q r s

合并后： a p b q r s

示例 3：

输入：word1 = "abcd", word2 = "pq"

输出："apbqcd"

解释：注意，word1 比 word2 长，"cd" 需要追加到合并后字符串的末尾。

word1： a b c d

word2： p q

合并后： a p b q c d

提示：

1 <= word1.length, word2.length <= 100

word1 和 word2 由小写英文字母组成

# 分析

## 方法一：双指针

思路：

使用两个指针分别指向两个字符串的起始位置，然后在一个循环中交替遍历这两个字符串的字符，并将它们依次添加到结果字符串中。具体步骤如下：

1、初始化一个空字符串 result 用于存储合并后的字符串。

2、使用两个指针 i 和 j 分别指向 word1 和 word2 的起始位置。

3、在一个循环中，不断地从两个字符串中取出字符，如果 word1 和 word2 中还有字符未取完，则依次取出一个字符添加到 result 中，并将指针向后移动一位。

4、当其中一个字符串遍历完毕后，继续将另一个字符串中剩余的字符依次添加到 result 中。

5、最后返回合并后的字符串 result。

代码：

class Solution {

public:

string mergeAlternately(string word1, string word2) {

string result = "";

int i = 0, j = 0;

// 只需要一个循环来遍历两个字符串中的字符

while (i < word1.size() || j < word2.size()) {

if (i < word1.size()) {

result += word1[i++];

}

if (j < word2.size()) {

result += word2[j++];

}

}

return result;

}

};