You are given a string text of words that are placed among some number of spaces. Each word consists of one or more lowercase English letters and are separated by at least one space. It's guaranteed that text **contains at least one word**.

Rearrange the spaces so that there is an **equal** number of spaces between every pair of adjacent words and that number is **maximized**. If you cannot redistribute all the spaces equally, place the **extra spaces at the end**, meaning the returned string should be the same length as text.

Return *the string after rearranging the spaces*.

**Example 1:**

**Input:** text = " this is a sentence "

**Output:** "this is a sentence"

**Explanation:** There are a total of 9 spaces and 4 words. We can evenly divide the 9 spaces between the words: 9 / (4-1) = 3 spaces.

**Example 2:**

**Input:** text = " practice makes perfect"

**Output:** "practice makes perfect "

**Explanation:** There are a total of 7 spaces and 3 words. 7 / (3-1) = 3 spaces plus 1 extra space. We place this extra space at the end of the string.

**Example 3:**

**Input:** text = "hello world"

**Output:** "hello world"

**Example 4:**

**Input:** text = " walks udp package into bar a"

**Output:** "walks udp package into bar a "

**Example 5:**

**Input:** text = "a"

**Output:** "a"

**Constraints:**

* 1 <= text.length <= 100
* text consists of lowercase English letters and ' '.
* text contains at least one word.