**Longest String Chain:**

Given a list of words, each word consists of English lowercase letters.

Let's say word1 is a predecessor of word2 if and only if we can add exactly one letter anywhere in word1 to make it equal to word2. For example, "abc" is a predecessor of "abac".

A *word chain*is a sequence of words [word\_1, word\_2, ..., word\_k] with k >= 1, where word\_1 is a predecessor of word\_2, word\_2 is a predecessor of word\_3, and so on.

Return the longest possible length of a word chain with words chosen from the given list of words.

**Example 1:**

**Input:** words = ["a","b","ba","bca","bda","bdca"]

**Output:** 4

**Explanation**: One of the longest word chain is "a","ba","bda","bdca".

**Example 2:**

**Input:** words = ["xbc","pcxbcf","xb","cxbc","pcxbc"]

**Output:** 5

**Constraints:**

* 1 <= words.length <= 1000
* 1 <= words[i].length <= 16
* words[i] only consists of English lowercase letters.