**[2020-03-09] Challenge #383 [Easy] Necklace matching**



Challenge

Imagine a necklace with lettered beads that can slide along the string. [Here's an example image.](https://www.craftkitsandsupplies.com/images/Beads/Alpha_Beads/Wood_Alphabet_Beads_26217.jpg) In this example, you could take the N off NICOLE and slide it around to the other end to make ICOLEN. Do it again to get COLENI, and so on. For the purpose of today's challenge, we'll say that the strings "nicole", "icolen", and "coleni" describe the same necklace.

Generally, two strings describe the same necklace if you can remove some number of letters from the beginning of one, attach them to the end in their original ordering, and get the other string. Reordering the letters in some other way does not, in general, produce a string that describes the same necklace.

Write a function that returns whether two strings describe the same necklace.

Examples

same\_necklace("nicole", "icolen") => true

same\_necklace("nicole", "lenico") => true

same\_necklace("nicole", "coneli") => false

same\_necklace("aabaaaaabaab", "aabaabaabaaa") => true

same\_necklace("abc", "cba") => false

same\_necklace("xxyyy", "xxxyy") => false

same\_necklace("xyxxz", "xxyxz") => false

same\_necklace("x", "x") => true

same\_necklace("x", "xx") => false

same\_necklace("x", "") => false

same\_necklace("", "") => true

Optional Bonus 1

If you have a string of N letters and you move each letter one at a time from the start to the end, you'll eventually get back to the string you started with, after N steps. Sometimes, you'll see the same string you started with before N steps. For instance, if you start with "abcabcabc", you'll see the same string ("abcabcabc") 3 times over the course of moving a letter 9 times.

Write a function that returns the number of times you encounter the same starting string if you move each letter in the string from the start to the end, one at a time.

repeats("abc") => 1

repeats("abcabcabc") => 3

repeats("abcabcabcx") => 1

repeats("aaaaaa") => 6

repeats("a") => 1

repeats("") => 1

Optional Bonus 2

There is exactly one set of four words in [the enable1 word list](https://raw.githubusercontent.com/dolph/dictionary/master/enable1.txt) that all describe the same necklace. Find the four words.

**SOLUTION (Pseudo-Code):**

* Use a data structure that has removeFirst, addLast for string of ‘n’ size.
  + Run the 2 operations
  + Run a comparison between new string and old string
  + Return true if comparison is the same.
* Repeat above for n-1 times.
* Return false.