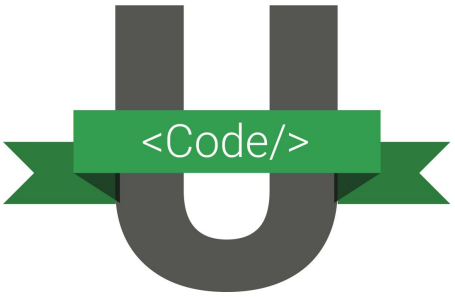

Assignment 1

If anything in the assignment is unclear, you have two options. You can ask for clarifications in the #assignments channel on Slack. It is also great if you can also make assumptions: real-world problems are always unclear, and as engineers we want to move on and make progress, even if we need to re-adjust later. If you do make assumptions, please try to identify them and document them as comments in your code.

Anagrams

Given two strings, determine if one is an anagram of the other.
Two words are anagrams of each other if they are made of the same letters in a different order.

For example:

- "listen" and "silent" are anagrams
- "triangle" and "integral" are anagrams
- "apple" and "pabble" are NOT anagrams

Optional Challenges

If you are done with the main assignment, you can pick up any of the optional challenges below and try to solve those as well. If you decide to take one or more optional challenges, make sure these are implemented as separate files from the original challenge. You might want to refactor some of your code so that you can reuse it across the main assignment and the optional challenges. You can submit a separate pull request for the optional challenges or include them with the main assignment.

1. Make the algorithm able to handle both case sensitive and case insensitive anagrams.
2. Make the algorithm able to handle anagrams of sentences, where the order of the letters is not tied to words and punctuation and white characters are ignored.
3. Make the algorithm able to handle anagrams of sentences, where each word in the first sentence is an anagram of the corresponding word in the second sentence.