#Code Challenge, April 26th 2017

Every week, we feature the type of brain-teasing question that [might be asked](https://discuss.codecademy.com/t/about-code-challenges/82161) in a full-stack developer’s job interview, similar to the [FizzBuzz Challenge](https://discuss.codecademy.com/t/challenge-the-classic-fizzbuzz-challenge/82165/1) often used at places like Facebook and Google.

For this week’s challenge, inspired by a question reported to have been asked in developer job interviews at Microsoft:

Write a function in your favorite programming language that will accept any two [strings 42](https://en.wikipedia.org/wiki/String_(computer_science)) as parameters and return “1” if they are [anagrams 362](https://en.wikipedia.org/wiki/Anagram) and “0” if they are not.

For simplicity’s sake, focus only on strings composed from uppercase letters (in hex ASCII - from 41 (A) to 5A (Z)) and space characters.

####Scroll down and **reply to this thread with your code** to participate! Don’t just submit your code, remember to explain your solution, too! Please link to your code on a site like [repl.it 45](https://repl.it/) to make assessing your entry easier (see below).

##Extra Credit  
####Intermediate Difficulty  
If you’d like to go a step further:

Make sure that your answer has no more than O(n + m) [time complexity 65](https://en.wikipedia.org/wiki/Time_complexity), where n is the length of the first string and m of the second.

Make your submission as efficient as possible! You can presume that the hypothetical interviewer in this case is looking for the most direct approach that does not involve len(n)! permutations.

Learn more about Big O notation and time complexity [here 41](https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/) and [here 18](http://stackoverflow.com/questions/487258/what-is-a-plain-english-explanation-of-big-o-notation). There’s also an old (and thus unsupported) [Codecademy course on Big O 28](https://www.codecademy.com/courses/big-o/0/1), if you prefer.

##The Winners

For the “best entries,” [see here](https://discuss.codecademy.com/t/challenge-anagram-detector/83127/58)!

Though it is now too late for any more submissions to be considered as a “best entry,” we encourage you to submit further responses anyway! It is [our hope 3](https://discuss.codecademy.com/t/codecademys-code-challenge-philosophy/83043) not only that you’ll benefit from the experience in testing your skills or preparing for interviews, and your posting your answer here will build up a library of content that will help others to learn too.

###How to participate

* Use any language taught on Codecademy ([Python 22](https://www.codecademy.com/learn/python), [Java 17](https://www.codecademy.com/learn/learn-java), [Ruby 20](https://www.codecademy.com/learn/ruby), [JavaScript 15](https://www.codecademy.com/learn/javascript), [PHP 3](https://www.codecademy.com/learn/php), etc.)
* Please include a link to your code running on service like [repl.it 45](https://repl.it/), [CodePen 11](https://codepen.io/), [JSFiddle 4](https://jsfiddle.net), or our very own [CodeBits 19](https://www.codecademy.com/codebits/) so that your fellow students can learn from your submission easier and faster - if we have hundreds of submissions, we may not have time to assess entries that do not have links in this manner
* If you want to post your code directly, please make sure that it is [correctly formatted 15](https://discuss.codecademy.com/t/how-do-i-format-code-in-my-posts/51139)!

Happy coding!

The fine print:

* Remember, the point of code challenges like these is to test and stretch yourself with an *unusual* problem, so don’t be dissuaded if you find it difficult or don’t know where to start! Start with Googling, but see if you can find out how to think about the problem and what tools you need to solve it, but don’t just go looking for the solution itself. This way, it’ll be a better learning exercise for you - developers can’t always find and copy “the right answer” online, which is why questions like these are [**used in developer job interviews**](https://discuss.codecademy.com/t/about-code-challenges/82161)! Interviewers want to be able to *see* how you *think through* problems and not just whether or not you can solve them.
* If you are interested in “winning” the code challenge, please don’t use any unusual repos or anything that will make it difficult for us or your fellow users to assess your answer quickly.
* Do you have a code challenge to share with other users? Issue it! Make a new topic with [Challenge] in the title to open a challenge, maybe we’ll even feature it in our next newsletter!