**Armstrong Numbers**

In Practice Mode

Introduction

An [Armstrong number](https://en.wikipedia.org/wiki/Narcissistic_number) is a number that is the sum of its own digits each raised to the power of the number of digits.

For example:

* 9 is an Armstrong number, because 9 = 9^1 = 9
* 10 is *not* an Armstrong number, because 10 != 1^2 + 0^2 = 1
* 153 is an Armstrong number, because: 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153
* 154 is *not* an Armstrong number, because: 154 != 1^3 + 5^3 + 4^3 = 1 + 125 + 64 = 190

Write some code to determine whether a number is an Armstrong number.

Getting Started

Make sure you have read the "Guides" section of the [C track](https://exercism.io/my/tracks/c) on the Exercism site. This covers the basic information on setting up the development environment expected by the exercises.

Passing the Tests

Get the first test compiling, linking and passing by following the [three rules of test-driven development](http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd).

The included makefile can be used to create and run the tests using the test task.

make test

Create just the functions you need to satisfy any compiler errors and get the test to fail. Then write just enough code to get the test to pass. Once you've done that, move onto the next test.

As you progress through the tests, take the time to refactor your implementation for readability and expressiveness and then go on to the next test.

Try to use standard C99 facilities in preference to writing your own low-level algorithms or facilities by hand.