**Square Root**

In Practice Mode

Introduction

Given a natural radicand, return its square root.

Check out the Wikipedia pages on [square root](https://en.wikipedia.org/wiki/Square_root) and [methods of computing square roots](https://en.wikipedia.org/wiki/Methods_of_computing_square_roots).

Note that the term "radicand" refers to the number for which the root is to be determined. That is, it is the number under the root symbol.

Recall also that natural numbers are positive real whole numbers (i.e. 1, 2, 3 and up).

When implementing this in a language with a built-in square root function, implement your own function.

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.