**Atbash Cipher**

## Introduction

Create an implementation of the atbash cipher, an ancient encryption system created in the Middle East.

The Atbash cipher is a simple substitution cipher that relies on transposing all the letters in the alphabet such that the resulting alphabet is backwards. The first letter is replaced with the last letter, the second with the second-last, and so on.

An Atbash cipher for the Latin alphabet would be as follows:

Plain: abcdefghijklmnopqrstuvwxyz

Cipher: zyxwvutsrqponmlkjihgfedcba

It is a very weak cipher because it only has one possible key, and it is a simple monoalphabetic substitution cipher. However, this may not have been an issue in the cipher's time.

Ciphertext is written out in groups of fixed length, the traditional group size being 5 letters, and punctuation is excluded. This is to make it harder to guess things based on word boundaries.

## Examples

* Encoding test gives gvhg
* Decoding gvhg gives test
* Decoding gsvjf rxpyi ldmul cqfnk hlevi gsvoz abwlt gives thequickbrownfoxjumpsoverthelazydog

## 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.