**Code-Wars Exercises**

**Katas**

1. Two to One:

Take 2 strings s1 and s2 including only letters from a to z. Return a new **sorted** string, the longest possible, containing distinct letters - each taken only once - coming from s1 or s2.

**Examples:**

a = "xyaabbbccccdefww"

b = "xxxxyyyyabklmopq"

longest(a, b) -> "abcdefklmopqwxy"

a = "abcdefghijklmnopqrstuvwxyz"

longest(a, a) -> "abcdefghijklmnopqrstuvwxyz"

1. Complementary DNA:

Deoxyribonucleic acid (DNA) is a chemical found in the nucleus of cells and carries the "instructions" for the development and functioning of living organisms.

If you want to know more: http://en.wikipedia.org/wiki/DNA

In DNA strings, symbols "A" and "T" are complements of each other, as "C" and "G". Your function receives one side of the DNA (string, except for Haskell); you need to return the other complementary side. DNA strand is never empty or there is no DNA at all (again, except for Haskell).

More similar exercise are found here: http://rosalind.info/problems/list-view/ (source)

Example: (input --> output)

"ATTGC" --> "TAACG"

"GTAT" --> "CATA"

1. Cockroach Speed

The cockroach is one of the fastest insects. Write a function which takes its speed in km per hour and returns it in cm per second, rounded down to the integer (= floored).

**For example:**

1.08 --> 30

Note! The input is a Real number (actual type is language dependent) and is >= 0. The result should be an Integer.

1. List Filtering

In this kata you will create a function that takes a list of non-negative integers and strings and returns a new list with the strings filtered out.

**Example**

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b"}) => {1, 2}

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b", 0, 15}) => {1, 2, 0, 15}

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b", "aasf", "1", "123", 231}) => {1, 2

1. Count the divisors of a number

Count the number of divisors of a positive integer n.

Random tests go up to n = 500000.

Examples (input --> output)

4 --> 3 // we have 3 divisors - 1, 2 and 4

5 --> 2 // we have 2 divisors - 1 and 5

12 --> 6 // we have 6 divisors - 1, 2, 3, 4, 6 and 12

30 --> 8 // we have 8 divisors - 1, 2, 3, 5, 6, 10, 15 and 30

Note you should only return a number, the count of divisors. The numbers between parentheses are shown only for you to see which numbers are counted in each case.

1. Count of Occurrences

Create a function that accepts a string and a single character, and returns an integer of the count of occurrences the 2nd argument is found in the first one.

If no occurrences can be found, a count of 0 should be returned.

("Hello", "o") ==> 1

("Hello", "l") ==> 2

("", "z") ==> 0

str\_count("Hello", 'o'); // returns 1

str\_count("Hello", 'l'); // returns 2

str\_count("", 'z'); // returns 0

Notes

The first argument can be an empty string

In languages with no distinct character data type, the second argument will be a string of length 1

1. Highest and Lowest

In this little assignment you are given a string of space separated numbers, and have to return the highest and lowest number.

Examples

Kata.HighAndLow("1 2 3 4 5"); // return "5 1"

Kata.HighAndLow("1 2 -3 4 5"); // return "5 -3"

Kata.HighAndLow("1 9 3 4 -5"); // return "9 -5"

Notes

All numbers are valid Int32, no need to validate them.

There will always be at least one number in the input string.

Output string must be two numbers separated by a single space, and highest number is first.

1. String ends with?

Complete the solution so that it returns true if the first argument(string) passed in ends with the 2nd argument (also a string).

Examples:

solution('abc', 'bc') // returns true

solution('abc', 'd') // returns false

1. Sentence Smash

Sentence Smash

Write a function that takes an array of words and smashes them together into a sentence and returns the sentence. You can ignore any need to sanitize words or add punctuation, but you should add spaces between each word. Be careful, there shouldn't be a space at the beginning or the end of the sentence!

Example

['hello', 'world', 'this', 'is', 'great'] => 'hello world this is great'

1. Square(n)Sum

Complete the square sum function so that it squares each number passed into it and then sums the results together.

For example, for [1, 2, 2] it should return 9 because 1^2+2^2+2^2=9

1. Reversed Strings

Complete the solution so that it reverses the string passed into it.

'world' => 'dlrow'

'word' => 'drow'

1. Perfect Square

### A square of squares

You like building blocks. You especially like building blocks that are squares. And what you even like more, is to arrange them into a square of square building blocks!

However, sometimes, you can't arrange them into a square. Instead, you end up with an ordinary rectangle! Those blasted things! If you just had a way to know, whether you're currently working in vain… Wait! That's it! You just have to check if your number of building blocks is a perfect square.

### Task

Given an integral number, determine if it's a [square number](https://en.wikipedia.org/wiki/Square_number):

*In mathematics, a****square number****or****perfect square****is an integer that is the square of an integer; in other words, it is the product of some integer with itself.*

The tests will always use some integral number, so don't worry about that in dynamic typed languages.

### Examples

-1 => false

0 => true

3 => false

4 => true

25 => true

26 => false