

# Kata

A programming kata is a problem to be solved. You can solve the problem any way you want, no matter the language. The important thing is to train your practice and reasoning in solving problems. It can be applied with TDD and also with other methods.

## SinglePairOrOdd

Returns a single integer that is pair or odd in a list of integers, where we can have only one pair and several odd or one odd and several pairs.

Reference: <https://www.codewars.com/>

## Count the Digit

Take an integer  $n$  ( $n \geq 0$ ) and a digit  $d$  ( $0 \leq d \leq 9$ ) as an integer. Square all numbers  $k$  ( $0 \leq k \leq n$ ) between 0 and  $n$ . Count the numbers of digits  $d$  used in the writing of all the  $k^2$ . Call `nbDig` (or `nbDig` or ...) the function taking  $n$  and  $d$  as parameters and returning this count.

Example:

$n = 10$ ,  $d = 1$ , the  $k^2$  are 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100. We are using the digit 1 in 1, 16, 81, 100. The total count is then 4. `nbDig(25, 1)`: the numbers of interest are 1, 4, 9, 10, 11, 12, 13, 14, 19, 21 which squared are 1, 16, 81, 100, 121, 144, 169, 196, 361, 441 so there are 11 digits '1' for the squares of numbers between 0 and 25.

reference: <https://www.codewars.com/kata/count-the-digit/train/java>

## FizzBuzz

Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

reference: <https://codingdojo.org/kata/FizzBuzz/>

## FooBarQix

### Step 1

If the number is divisible by 3, write "Foo" instead of the number

If the number is divisible by 5, add "Bar"

If the number is divisible by 7, add "Qix"

For each digit 3, 5, 7, add "Foo", "Bar", "Qix" in the digits order.

Example:

1 => 1

3 => FooFoo (divisible by 3, contains 3)  
6 => Foo (divisible by 3)  
15 => FooBarBar (divisible by 3, divisible by 5, contains 5)  
21 => FooQix  
33 => FooFooFoo (divisible by 3, contains two 3)

Step 2

We have a new business request : we must keep a trace of 0 in numbers, each 0 must be replace par char “\*“.

Example:

101 => 1\*1  
303 => FooFoo\*Foo  
105 => FooBarQix\*Bar  
10101 => FooQix\*\*

reference: <https://codingdojo.org/kata/FooBarQix/>

## Fibonacci

The Fibonacci sequence is a sequence of numbers, where the number 1 is the first and second term of the order and the others are originated by the sum of their predecessors.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

## Factorial

The factorial of a number is calculated by multiplying that number by all its predecessors until it reaches number 1.

## Roman numbers

Convert decimal numbers into Roman numerals

## Amount Character

In a given text, return the specified amount of a character.

## Money, Money, Money

Mr. Scrooge has a sum of money 'P' that wants to invest, and he wants to know how many years 'Y' this sum has to be kept in the bank in order for this sum of money to amount to 'D'.

The sum is kept for 'Y' years in the bank where interest 'I' is paid yearly, and the new sum is re-invested yearly after paying tax 'T'

Note that the principal is not taxed but only the year's accrued interest

Example:

Let P be the Principal = 1000.00

Let I be the Interest Rate = 0.05

Let T be the Tax Rate = 0.18

Let D be the Desired Sum = 1100.00

After 1st Year --> P = 1041.00

After 2nd Year --> P = 1083.86

After 3rd Year --> P = 1128.30