POETRY OF PROGRAMMING

CLOJURE PRACTICE PROBLEMS

version-2017.06.24

These problems require relatively little work, usually just writing a function is sufficient (it is of course possible to break the solution down into several functions).

(1) Write a function `capitalize-words` that makes every word starting with a capital letter in a sentence given in a string.

```
(capitalize-words ”Hello world!”)
”Hello World!”
(capitalize-words ”So much universe, and so little time.”)
”So Much Universe, And So Little Time.”
```

Hint: check functions in `clojure.string`: `split`, `join`, `trim`, `capitalize`; or use `interpose`, `str`.

(2) The *median* is the middle value of a set of (here numerical) observations, when observations are ordered. For odd number of data points, it is the middle number, for an even number it is the average of the middle two. Write a function `median` that takes a collection of number and returns their median.

```
(median [9 1 3 8 3 5 7 8])
5
(median [2 4 5 6 1 8 2 9])
9/2
```

Hint: Keep in mind that Clojure starts indexing from `0`. Check functions: `sort`, `int`.

(3) Write a function `exp` that takes two non-negative integers m and n and returns m^n.

```
(exp 2 3)
8
(exp 10 2)
100
```

How many multiplications does the function do in order to calculate the result? Can it be reduced? If yes, write a function `fast-exp` that computes the power with less multiplications! Hint: $5^{100} = 5^{50} \cdot 5^{50}$.