

Homework 4

Due: 14 Feb 2024

For each problem, include print statements that illustrate how the functions and methods are called on some concrete sample data and what their expected return values would be.

For full credit you should write these relatively idiomatically and not simply as a word-for-word translation of a C++ or Python solution. In all cases, you are free to write helper functions if you think they would be helpful.

Problem 4.1

Write the following function in Ruby: `first_abbrev` takes a list, and finds the alphabetically-first of the elements of the given list that are strings starting with a letter and ending with a period; or `nil` if there are no such elements in the list. You shouldn't assume that all the elements are strings or that they are all uppercase.

Problem 4.2

Write the following function in Ruby: `freq_build` takes a list and a positive integer, and builds a frequency table of elements of the given list whose frequency is at least the given number. (So, if the number is 1, it would include everything that occurs in the list, but if it's 2, it would include only the values that occur twice or more, and so on.) The return value should be a hash table whose keys are values from the original list, each associated to the correct count.

Problem 4.3

Write the following class in Ruby: A `Fraction` represents a rational number (and is constructed by providing a numerator and denominator). While a complete implementation might have dozens of methods, you'll focus on four:

- `times` produces another `Fraction` that is the product of this `Fraction` with a given other `Fraction`

- `times!` updates this `Fraction` by multiplying it with a given other `Fraction`
- `to_s` makes a string view of this `Fraction`, such as "2/3"
- `to_f` computes the floating-point approximation of this `Fraction`

You don't have to worry about reducing the fractions or anything like that.

A readme file in the directory should let me know where to find the three problems' functions/classes and their test cases.

Handing in:

Assuming you have made a subdirectory for this assignment, first go to that subdirectory (possibly with a sequence like

```
cd 210
cd hwk4
```

or similar, depending on your specific naming choices), then type the following:

```
handin cmsc210 hwk4 .
```

If you haven't made a subdirectory for it, you can replace the dot in that with all the files you need to hand in, separated by spaces (but the subdirectory way is certainly easier).