NodeJS Acceleration

Exercise 7

Preamble: Functional programming introduces a different thinking paradigm to coding, it encourages declarative coding as opposed to imperative coding, meaning you'd make abstraction work by all means possible. In the following tasks, you will work more on solidifying your understanding of functional programming concepts, and their implementation with a number of case scenarios.

Task A:

Problem statement: Repeat *Ex 6, Task B*, paying close attention to the accompanying instructions: you are to implement your own array-like data structure with the specified functions (custom-coded) enabled.

Requirement:

1. In addition to the required functions in the referenced task, you are to implement the following on your data structure:
   - `slice()`
   - `splice()`
   - `join()`

(opt)

Task B:

Problem statement: After you have gained adequate understanding of the following concepts, create `repls` with custom example codes (at least one `repl` per team member) showing your new-found understanding, accompanied by a draft submitted to the Organization's Medium publication.

Requirements:

1. The aformentioned concepts are:
   - Currying & Partial application
   - Composition & Pipelines
   - Functors & Monads
2. Your dissertation should ensure to show the similarities/differences where necessary between the pair of concepts
3. Each team member must have at least one `repl` per pair of concepts with custom examples

Submission guideline:

You are to create a github repo containing links to all `repls` from your team, this repo must also contain all the codes.

(20pt)

Task C:

Problem statement: Being as functional as possible, implement the system as described in the following requirements.

Requirements:

1. The system is able to accept input data representing students and their scores in an examination.
2. The system is able to classify students by grade (you are to design your grade range [A-F] at your own discretion).
3. The system is able to determine `x-quartile` of the students.
4. The system is able to determine the predominant grade in the class of students.
5. All of these operations can be composed in any order.
6. You test data must be present in your repo as a json file. *See https://www.json-generator.com/ — tool for generating random JSON data.*

(20pt)

P.S:

1. *Tasks with null points (0pt) add nothing to your grade when executed, but reduce it when they aren't*