CS2613: Programming Languages Laboratory (FR02A)
Lab #14 – Winter 2024

Language: Racket (#3)
# of Tasks: 3
Topics:

- Lambda/Anonymous Functions
- Mapping and Filtering

*All tasks are to be completed individually in line with the academic offense guidelines detailed on the syllabus and are **due before the end of the lab period** unless stated otherwise.*

Task #1

Task Style: Programming in partners.
Submission Method: Move onto Task #2.

Description:
Using the following formula, you and your partner will both write a function that takes a time (in seconds) and returns the final velocity. The main difference is that each partner will use different constants for initial velocity and acceleration. See below for constants.

$$V_f = V_i + at$$

$$V_f = \text{final velocity } (\frac{\text{m}}{\text{s}})$$
$$V_i = \text{initial velocity } (\frac{\text{m}}{\text{s}})$$
$$a = \text{acceleration } (\frac{\text{m}}{\text{s}^2})$$
$$t = \text{time (s)}$$

Partner 1's Constants:

$$V_i = 5\text{m/s}$$
$$a = 0.5\frac{\text{m}}{\text{s}^2}$$

Partner 2's Constants:

$$V_i = 32\text{m/s}$$
$$a = -0.35\frac{\text{m}}{\text{s}^2}$$

Task #2

Task Style: Programming in partners.
Submission Method: Move onto Task #3.

Partner 1:

Create a function that returns a list of values from 0 -> 60. When complete, help your partner complete their function.

Partner 2:

Create a function that takes two lists of values and a starting position (likely 0). The function should return the position of the first occurrence in the list where the values are less than a tolerance apart. For this question, we will use a tolerance of 1 – remember to work with absolute values. If there are not 2 values within a tolerance apart, return a -1.

Example:

```
(matching '(6 2 4 7 8 9) '(1 5 7 3 7.5 9) 0) = 4
(matching '(6 2 4 7) '(1 5 7 3) 0) = -1
```

When complete, help your partner complete their function.

When both functions are complete. Move onto Task #3.

Task #3

Task Style: Programming in partners w/ Self-Assessment
Submission Method: Complete Self-Assessment on D2L.

Description:
Now that you and your partner both have 2 functions, send all functions to a single program.

Together, write a function named “same-speed” that takes the following 3 parameters: Task #1: Function 1, Task #1: Function 2, and a list of times (represented in seconds - 0->60 within the first minute of driving). The function should return the number of seconds that have passed when the two vehicles are within 1m/s of each other’s speed.

Effectively, one vehicle is accelerating for 60 seconds while the other is decelerating for 60 seconds. The speed of acceleration/deceleration is the same as the values found in Task #1. When are the two vehicles first going approximately the same speed?

When you think you have an answer, complete the Self-Assessment found on D2L. If incorrect, try again.

Example Function Call:

```
(same-speed accel decel (list-60 0)) = ???
```

Resources:

- The Racket Reference
  - 4.3.2 – abs
- https://learnxinyminutes.com/docs/racket/
  - map
  - drop