Report

SICP Question

Not much to report. It went smoothly; finger exercises were a bit of a pain.

Lab

Problem 1

No issue here. Adding items to lists isn't very difficult.

Problem 2

When passed the unmodified `(change-person)` procedure is passed, `'(you are not being very helpful to me)`, it returns `(you are not being very helpful to you)`. It dumbly replaces `me` with `you`.

When the *modified* `(change-person)` procedure (that is, the suggested replacement pairs are placed at the end of the procedure) is passed the same phrase, it returns `(you are not being very helpful to you)` — no difference from the unmodified procedure. When the replacement pairs are introduced the the `(change-person)` procedure at the beginning of the procedure, the same input returns `(i am not being very helpful to i)`. The bug present is that that `(change-person)` procedure changes *every* mention of the second-person into the first person, which is not what the goal of problem 2 is.

Basically, pronouns are being double-replaced. I'm considering using a map operation instead.

Problem 3

I want to document how the list `past-respones` is handled by the program loop. It is initially created in the procedure `(visit-doctor)`, and is passed through every iteration of the `(doctor-driver-loop)`. When the procedure `(reply)` is called, `past-responses` is passed to the `(reply)` method. 12.5% of the time, the `(reply)` method gets a past response of the user.

Problem 4

I chose to modify both the `(doctor-driver-loop)` and the `(visit-doctor)` procedures in order to implement an end condition (reached when the user enters `(suppertime)` when the doctor program asks, `(who are you)`). Initially, I chose to modify the `(doctor-driver-loop)` exclusively, but doing so made the `(doctor-driver-loop)` program too unwieldy, with far too many nested conditional statements. Instead, I chose to use `(visit-doctor)` as the procedure that handled entering and exiting the doctor loop. That is, if the patient's name is `suppertime`, the program terminates. Otherwise, it "sees" the next patient.