

CS 2500, Spring 2013

Problem Set 5

Due date: Tuesday, February 12 @ 11:59pm

What to submit:

Using Blackboard, submit a single Racket file containing all of the code and documentation for this assignment. Place your name and husky email address in a comment at the beginning of your file.

Name your file: hw5-yourlastname.rkt

The goal of this problem set is to study the design and processing of self-referential data. You must follow the design recipe in your solutions: graders will look for data definitions, contracts, purpose statements, examples/tests, and properly organized function definitions. For the latter, you must follow templates. You do not need to include the templates with your homework, however, unless the question asks for it.

Problem 1.

Develop the function `check-pass-6-10?` which consumes a list of passwords (represented as strings) and produces a Boolean indicating whether all are at least 6 characters but no more than 10 characters long.

Generalize the function to `check-pass?` which consumes a list of passwords and a minimum and maximum length and produces a Boolean indicating whether all passwords are within the allowed length span.

Problem 2.

The 2htdp/image teachpack contains many functions which create images of simple geometric figures: `circle`, `ellipse`, `line`, `triangle`, and so on.

- Provide a data definition for an entry which names a figure and also contains a corresponding example.
- A catalog contains any number of entries. Provide a data definition for a catalog and construct a specific example of a catalog of at least five different entries.
- Now develop the function `show-example`, which consumes the name of a figure (represented as a symbol) and a catalog. It produces the corresponding image or false if the named figure was not in the catalog.

Problem 3.

Develop the function `cesarify` which consumes a list of symbols and returns the same list but with every instance of `'pizza` doubled. For example,

```
(cesarify (cons 'wurst (cons 'huevos (cons 'pizza (cons 'pants
empty)))))
```

would be expected to return:

```
(cons 'wurst (cons 'huevos (cons 'pizza (cons 'pizza (cons 'pants
empty)))))
```

Problem 4.

Suppose we have the following data definition:

```
(define-struct ball (x y color))
;; Ball = (make-ball Number Number Color)
;; Color is one of 'red, 'yellow, 'blue, etc.
```

- a) Think of instances of `ball` as a Cartesian point, specifying where the ball is located, and the color of the ball.
- b) Provide a data definition for lists of `Balls`.
- c) Provide a template for processing such lists.
- d) Design the function `lob-length`, which counts how many `Balls` are on a given list of `Balls`.
- e) Design the function `lob-yellow`, which changes the color of all of the balls in a list of `Balls` to yellow.
- f) Design the function `lob-draw`, which consumes a list of `Balls` and adds them to an empty scene of 300 x 300 as appropriately colored circles of radius 3.
- g) Design `lob-member?`. The function consumes a list of `Balls`, `lob`, and a `Ball` `b` and determines whether `b` occurs in `lob`.

Problem 5.

The goal of this problem is to develop a component of a slide-show program such as PowerPoint or Keynote. The component displays a single, animated slide. That is, it starts with a plain background and adds phrases to the slide at the rate of one every second. Here are the data definitions:

```
(define-struct txt (content x y))
;; Txt = (make-txt String Number Number)

;; LoTxt is one of:
;; -- empty
;; -- (cons Txt LoTxt)

(define-struct world (image hidden))
;; World = (make-world Image LoTxt)
;; Interpretation:
;; The world's image represents the image that the
;; audience can see.
;; The world's list of Txt represents the
;; yet-to-be-revealed elements.
```

Create a world with an empty 400 x 400 canvas to which the program will add the following three phrases: "On your mark.", "Get set.", and "Go!", which the program will add one step at a time to the canvas.

Design the function `display`, which consumes a world and returns its current image. Design the function `next`, which consumes a world and adds the next hidden `Text` to the currently visible slide image. Use 30pt font and blue for the color of the text.

Optional: Make the program run and display the animated slide from above with these lines:

```
(big-bang WORLD-0
  (on-tick next 1)
  (to-draw display))
```

If you do, be sure to delete them or comment them out before you turn in your solution.