CS 2500, Spring 2014
Problem Set 5

Due date: Tuesday, February 11 @ 7pm

Programming Language: BSL

Purpose: This problem set concerns the design and processing of self-referential data definitions.

Finger Exercises HtDP/2e: 135, 136, 141, 149, 150, 151, 152, 154, 162, 164, 173;

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:

a) 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.

b) 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.

a) Provide a data definition for an `entry` which names a figure and also contains a corresponding example.

b) 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`.

c) 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:
Recall the data definition for a Shape from Problem Set 4.

a) Provide a data definition for Lists of Shapes.
b) Provide a template for processing such lists.
c) Design the function shape-list-length, which counts how many Shapes are on a given List of Shapes.
d) Design the function yellow-shapes, which changes the color of all of the shapes in a List of Shapes to yellow.
e) Design the function draw-shapes, which consumes a List of Shapes and adds them to an empty scene of 500 x 500.
f) Design shape-member?. The function consumes a List of Shapes, losh, and a Shape s and determines whether s occurs in losh.

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 scene 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 Txt to the currently visible slide image. Use 30pt font and blue for the color of the text.