CS5001 Assignment 2

Programming Language: BSL

Due Date Wednesday 1/24 at 10:00pm

Purpose To write more simple functions that also include conditionals, design struct-type data.
.

For this assignment and all future assignments you must upload a .rkt file in the specified language to the Handin server (handins.ccs.neu.edu).

Exercise 1 Here is a table printed on the back of a speeding ticket:

Percentage of Excess Speed	Amount ($)
up to 10	50
up to 25	150
up to 100	400
beyond	2500

- Design an interval data definition for excess speed levels.
- Design `amount-of-ticket`. Given the speed of a car and the speed limit of some road, the function determines the penalty a driver has to pay.

Exercise 2 Consider the following structure definitions:

```
(define-struct painting (artist year style))

(define-struct album (title singer genre))

(define-struct senator (state party))

(define-struct person (name mother father height))

(define-struct shoe (style brand color))
```

- What are the names of the constructors, predicates and selectors that each of the structures adds to Racket?
- Provide data definitions for the structure definitions above. Make appropriate assumptions about what data goes with each field.
- Develop templates for functions that consume the structures above.

Exercise 3

- Provide a structure type definition and a data definition for representing 4-digit PIN numbers. A PIN number consists of digits represented with a number from 0 to 9.

- Design the function `difference`. It consumes two representations of PIN numbers and creates a new PIN number from the differences. For each position in the PIN, it subtracts the lesser number from the greater. If the two are the same the new digit is 0. Note: The problem statement mentions two different tasks: one concerning PINs and one concerning digits. This suggests that you design two functions.

Exercise 4 The drawing of a chameleon in Section 5.11 of HtDP/2e is a transparent image. To insert it into DrRacket, copy and save the image from HtDP/2e and insert it with the "Insert Image" menu item. Define the image with the name CHAM. Using this instruction preserves the transparency of the drawing's pixels.

When a partly transparent image is combined with a colored shape, say a rectangle, the image takes on the underlying color. In the chameleon drawing, it is actually the inside of the animal that is transparent; the area outside is solid white. Try out this expression in your DrRacket:

```
(overlay
     CHAM
     (rectangle (image-width cham)
                (image-height cham) "solid" "red"))
```

Design a world program that has the chameleon continuously walking back and forth across the screen (you should rotate the image 90 degrees so it is facing right). It starts walking from left to right and when it reaches the right end of the screen, it flips horizontally and begins walking from right to left and switches direction again when it reaches the left boundary.

Of course, like all chameleons, ours can change color, too: the key "r" turns it red, "b" blue, and "g" green.

Start with a data definition, `VCham`, for representing chameleons. You will need to represent the position, color and direction of the chameleon in your definition.

Here is a wish list for the functions you will need:

- `draw-cham` – renders the world as a chameleon facing left or right
- `next-cham` – compute the new location of the chameleon in one clock tick from now
- `change-cham` – changes the color of the chameleon if the "r", "g" or "b" key is pressed