Scheme Tutorial Exercises

Fall 2002

Problem Set 2: Lists and Trees

11. Develop the function *check-range1*, which consumes a list of temperature measurements and checks whether all measurements are between $5°C$ and $95°C$ inclusively. **(HtDP Exercise 9.5.4)**

    Generalize the function to *check-range*, which consumes a list of temperature measurements and a legal interval and checks whether all measurements are within the legal interval.

12. Develop the function *convert*. It consumes a list of digits and produces the corresponding number. The first digit is the least significant, and so on. **(HtDP Exercise 9.5.5)** For example:

```
(convert (cons 1 (cons 2 (cons 3 empty))))
> 321
```

13. Define the function *average-price*. It consumes a list of toy prices and computes the average price of a toy. The average is the total of all prices divided by the number of toys. **(HtDP Exercise 9.5.7)**

14. Develop *convertFC*. The function converts a list of Fahrenheit measurements to a list of Celsius measurements. **(HtDP Exercise 10.1.3)**

15. Develop the function *eliminate-exp* to eliminate expensive toys. The function consumes a number, called *ua*, and a list of toy prices, called *lotp*, and produces a list of all those prices in *lotp* that are below or equal to *ua*. **(HtDP Exercise 10.1.5)** For example:

```
(eliminate-exp 1.0 (cons 2.95 (cons .95 (cons 1.0 (cons 5 empty)))))
; expected value:
(cons .95 (cons 1.0 empty))
```

16. Define the function suffixes, which consumes a list *l*, and produces a list of all suffixes of *l*. For example:

```
(suffixes '(a b c d))
> ((a b c d) (b c d) (c d) (d) ())
```

17. Define a datatype for a family tree. A family tree is either:

    - Unknown

    or

    - A person, which has five fields:
        - name, which is a string
        - birthyear, which is a number
        - eyecolor, which is a symbol
        - father, which is family tree
        - mother, which is family tree

    For example, a small family tree looks like:

```
(person "Dave" 1977 'brown
        (person "Ken" 1945 'brown
                (unknown)
                (unknown))
        (person "Mary Ellen" 1946 'brown
                (unknown)
                (unknown)))
```

18. Develop *count-persons*. The function consumes a family tree node and produces the number of people in the corresponding family tree. **(HtDP Exercise 14.1.3)**

19. Develop the function *average-age*. It consumes a family tree node and the current year. It produces the average age of all people in the family tree. **(HtDP Exercise 14.1.4)**

20. Develop the function *eye-colors*, which consumes a family tree node and produces a list of all eye colors in the tree. An eye color may occur more than once in the list. **(HtDP Exercise 14.1.5)**

    Hint: Use the Scheme operation *append*, which consumes two lists and produces the concatenation of the two lists. For example:

```
(append (list 'a 'b 'c) (list 'd 'e)) = (list 'a 'b 'c 'd 'e)
```