Chapter 3 Problem Set

Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor or teaching assistants).

Please formulate code to address each question. You will upload the code files.

1. [5 points] Build the following list using a Racket function utilizing cons, list and quote:
   (4 (7 22) “art” (“math” (8) 99) 100)

2. [15 points] Build a Racket function that will process a list. The function will pass in a function, such as + that takes two parameters, and a list. The list will be processed using the function passed in until there is only one answer remaining. Here is an example of how the processing will take place when the function is + and the list is (3 5 7 9): (3 5 7 9) → (8 7 9) → (15 9) → 24

3. [15 points] Build a Racket function that decreases a list of tax amounts by 2%. Make your solution tail recursive and generic so that you could manipulate the list in any way.

4. [10 points] Build a Racket function that averages a list of numbers. Do not use a built-in Racket function. Design your own function to accomplish this task.

5. [15 points] Build a Racket function that will accept a list where the head of the list is any one of the following operations: + , -, * , or /. The function you create will then call a function you design to perform the operation passed in on the remaining list of numbers. You will not use a map function. Instead you will design your own function for each operation to correctly handle the operation on a list of numbers.

6. [10 points] Build a Racket function that takes two lists and returns one list that contains the separate unique entities from both list, with no duplication.

7. [5 points] Build a Racket function that utilizes parallelism to process a list of numbers. The function will remove any negative numbers and take the square root of any remaining numbers.

8. [20 points] Create a collection in Java to store information about soccer teams. You will import the information from the soccerStats.txt file given. This file contains the following categories of data: Team Name, League, Goals, Shots per game, Possession Percentage, Pass Success Percentage, and Aerials Won. You will use lambda expressions to perform the following tasks:
    - Print the team names and Leagues
    - Print the name and league of the teams that have scored less than 5 goals
    - Average the goals of all the teams
    - Sort the teams by league, displaying the team name and league
    - Sort the teams by pass success percentage, displaying the team name and pass percentage
    - Sort the team by league and possession percentage
    - Reset each team's aerials won to 0

9. [+5 points] Build a Racket function that displays the leap years from 1800 through this year.