HW1: Practice Dr.Racket (6 points)

***This HW#1 can give some sense whether you can follow this course or not. If you can't solve and understand these problems, please, consider not to take this course now[[1]](#footnote-0) but you can take it later when you have better coding sense.***

***All problems can be solved by the knowledge from slides in Lecture 2 and 3 and your common coding sense if you clearly understand the slides. You may discuss with your classmates and get some help from others but for each answer you must add comments whether you get help from your friends, TAs, JC or other materials except for slides, textbook, Racket documents. However, try doing HW#1 by yourself as much as possible. If you do not fully practice, you cannot solve other HW tasks. Prepare yourself for the next HW tasks. Simply copying the code from others is considered cheating. Please, add time taken as well. For function definitions, please add [contract], [purpose], [tests] (define at least two test cases) as comments. For example,***

| *; Problem 1:*  *; Solved by myself:* ***N*** *(TA helped me. I could not solve this problem by myself without TA's help. I also see the example code from https://address.to.the.example.code)*  *; Time taken:* ***about 30 mins***  *;* ***[contract] mile****->****km: number -> number***  *;* ***[purpose] To convert mile to km***  *;* ***[tests] (test (mile****->****km 1) 1.6)***  ***; (test (mile****->****km 2) 3.2)***  ***(define (mile****->****km mile)***  ***...***  ***(test (mile****->****km 1) 1.6)***  ***(test (mile****->****km 2) 3.2)***  *; Problem 2:*  *; Solved by myself:* ***Y***  *…*  *; Problem 8:*  *; Solved by myself:* ***N (It is too difficult. I'm considering to drop this course.)***  *; Time taken:* ***about 100 mins***  *…* |
| --- |

# Problems

1. (0.5) Write the function *dollar->won*, which consumes an integer number (dollar) and returns the integer (won). Suppose that 1 dollar is converted as 1,342 Korean won
2. (0.5) Write the function *digit\_sum*, which consumes three integer numbers and returns the sum of them.
3. (0.5) Write the function *volume-sphere*, which consumes an integer number denoting a radius of a sphere and produces the volume of the sphere. (Use *pi* in Racket rather than 3.14)
4. (0.5) Write the function *is-even*?, which consumes an integer number and returns whether the number is even.
5. (0.5) Write the function *combination*, which consumes two integer numbers and returns the number of combinations that can be there, when first function parameter is *n* (which is the number of elements in the set), and second function parameter is *k* (which is the number of elements in each subset).

*\* What is the combination?*https://en.wikipedia.org/wiki/Combination

1. Complete the following tasks
   1. (0.5) Define the type Vehicle, which has three variants, Bicycle, Car, Airplane.   
      The Variant `Bicycle' has one attribute, ‘wheels’, which indicates the number of wheels.   
      Car has two attributes, ‘wheels’ and ‘windows’, which indicates the number of wheels and windows.   
      Airplane has three attributes, ‘wheels’, ‘windows’, and ‘engines’, which indicates the number of wheels, windows, and engines.
   2. (0.5) The government is collecting the tax for vehicles. Tax for a vehicle differs on the number of wheels, windows and engines. Define the function *vehicle-tax*, which consumes an instance of Vehicle, tax per wheel, tax per window, tax per engine, and returns the total amount of tax that should be paid for that vehicle. Note that the number of wheels, windows, and engines are not fixed. For instance, a Car can have 4 or even 8 wheels, or an Airplane can have 0 windows.
   3. (0.5) The government is making sure that all vehicles meet the safety standards. Define the function *is-vehicle-safe* that consumes a Vehicle instance and returns a string “safe” or “unsafe” according to whether the vehicle is safe or not.   
      Here are the safety standards. For a Bicycle, wheels must be less than 4. For a Car, wheels must be more than 3, windows must be more than 2. For an Airplane, wheels must be more than 2, windows must be more than 10, engines must be more than 1. If any of the conditions are not satisfied, the vehicle is not safe.
2. (1.0) Define the function *update-name*. The new function consumes two symbols, called old and new, and a list of symbols. It produces a list of symbols by replacing all occurrences of old by new. For example,  
   (update-name 'cherry 'claire (cons 'jc (cons 'cherry (cons 'kate empty))))  
    produces  
   '(jc claire kate)
3. (1.0) Define the function *binary-search*. This function consumes a list of numbers which is sorted by ascending order and a number which is the target, called ‘lst’ and ‘target’. It produces a list of numbers which is binary-search traversal history. There are no duplicate numbers in the input list. And also, the target number always exists in the input list. For example,
   1. (binary-search '(1 2 3) 3) produces '(2 3)
   2. (binary-search '(1 2 3 4 5 6 7 8) 3) produces '(4 2 3)
   3. (binary-search '(1 2 3 4 5 6 7 8 9 10) 9) produces '(5 8 9)
   4. (binary-search '(1 2 3 4 5 6 7) 6) produces ‘(4 6)
   5. (binary-search '(1 2 3 4 5 6 7 8 9) 4) produces ‘(5 2 3 4)
   6. (binary-search '(1 2 3 4 5 6 7 8 9) 3) produces ‘(5 2 3)

# What to submit

1. Submit via LMS and name your file as follows:
   * HW1\_[Student\_id]\_[name].rkt  
     e.g., HW1\_19800179\_JaechangNam.rkt

# Due Date

1. Upload your code file by
   1. 23:59, September 8 (Thu) 2022.

# Evaluation (Full mark: 6 points)

* Late submission
  + The number of days delayed \* -2
* Missing comments and test cases required
  + -0.1 for each
* Test cases (we make) were not passed.
  + -0.2 for each test case
* Incorrect or wrong implementation
  + 40% deduction for each problem.

# Others (need help?)

Please ask questions via the LMS Q&A board. Anybody can ask asd answer questions. If you initiate a good discussion or answer well, you can get a bonus point (+0.5). Please be proactive!!

1. This may give you a disaster this semester if you can't solve HW1 by yourself. However, if you can solve and at least understand them, ***you will experience the beauty of coding and the functional programming paradigm***. [↑](#footnote-ref-0)