HW#1: 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] 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 future 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
…
```

[1] This may give you a disaster in 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 functional programming paradigm***.

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 1,162 Korean won

2. (0.5) Write the function digit_sum, which consumes three integer numbers and returns the sum of them.

3. (0.5) An Inchworm can travel *n* inches per one hour (speed). Write the function *inchworm-travel*, which consumes two integer numbers that denote speed *n*, and hours, and returns the distance an Inchworm can travel in that time in centimeters. Assume that an inch is 2.54 centimeters.

4. (0.5) Write the function *is-odd*?, which consumes an integer number and returns whether the number is odd.

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). Implement the factorial function as well. (Do not use a built-in function in Racket.)

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

6. Complete the following tasks
   a. (0.5) Define the type PERSON, which is Professor, UndergraduateStudent, or GraduateStudent. Professor has two attributes: 'courses' for the number of courses taught and 'projects' for the number of research projects conducted. UndergraduateStudent has one attribute: 'courses' for the number of courses taken. GraduateStudent has two attributes: 'courses' and 'papers' for the number of papers submitted.
   b. (0.5) Define the function *have-courses*, which consumes any person instance and produces the number of courses taken or taught for the given person.

c. (0.5) Define the function *ready-to-graduate*, which consumes a person. If a given person is a GraduateStudent with more than or equal to three papers, then return true, otherwise produce false.

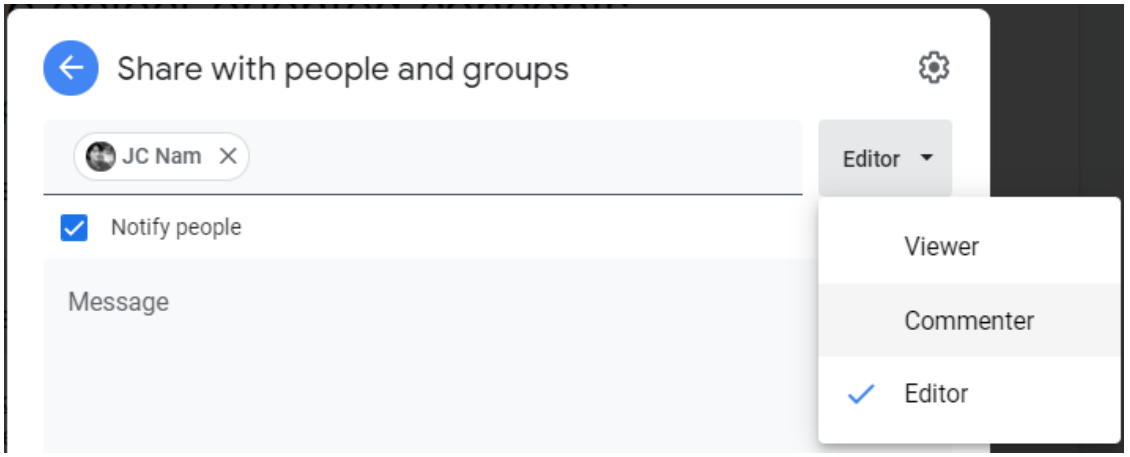
7. (1.0) Define the function *name-alphabet*, which consumes a list of alphabets and produces a corresponding list of an alphabetical character with names starting with the alphabet character; it names all occurrences of 'a with 'alice, 'c with 'cherry, 'j with 'jc, 'k with 'kate and keeps the other characters as *unnamed* (either 'unnamed or empty '()' is ok).
'(a b n) => '(alice unnamed unnamed)

8. (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)

What to submit

1. Create your own google drive directory for homework and your score sheet.
   - Directory name: [Section#]_[student id]_[Name] e.g.
     01_19800179_JaechangNam
     or
     02_19800179_JaechangNam
     based on your PL class section.
   - Share the directory with JC (jcnam@handong.edu) and TAs (yhchoi@handong.edu, 21700613@handong.edu)

i. When you share the directory, give JC and TAs "Editor" for the directory and files in it.



ii. Then give JC Nam **'Make Owner'** for the shared directory and all other files in the directory!

1. If you forget this, 0.2 point will be deducted after the deadline

2. Upload the following code file that contains your answers of HW#1 in your google directory.
   - HW1_[Student_id]_[name].rkt
     e.g., HW1_19800179_JaechangNam.rkt
   - Optional: if you want to get some comments while you solve each problem, create a google doc (name the doc title as 'HW1_[Student_id]_[name]') and paste your answer code when you solve each problem.
3. Make a google sheet like this in the directory
   - https://docs.google.com/spreadsheets/d/14TGYarbvCkDMJIndT37FR9fpg_A8OwFtvvGHhzDV-IY/edit#gid=0

   Change the title with your student id and name. This will be used for your personal score sheet and you will get feedback about your homework/quizzes and exams.

Due Date

1. Create and share your Google drive directory
   a. 22:00, September 6 (Mon) 2021.
2. Create (copy) your score sheet in your shared google directory
   a. 22:00, September 6 (Mon) 2021.
3. Upload your code file
   a. 22:00, September 12 (Sun) 2021.

Evaluation (Full mark: 6 points)

- Late submission
  - The number of days delayed * -2
- Did not give JC Ownership for your shared google directory
  - -0.2
- 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?)

HW1 FAQ
https://docs.google.com/document/d/1cAOtOM_NI5aLBli7LtPHC1Ul73v9ergVxvqgKvvm0Uo/edit
You can put questions in HW1 FAQ. (Tag me and TAs, yhchoi@handong.edu, 21700613@handong.edu in a comment then we get notified.) When you tag us, put one comment for one question.