CS 115 – Winter 2018. Assignment 02. Due: Wednesday, January 24 at 10:00 a.m.

• Do not use any conditional expressions (cond) on this assignment.
• For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch, including helper functions.
• For full marks, it is not sufficient to have a correct program. Be sure to follow all the steps of the design recipe. Read the Style Guide carefully to ensure that you are following the proper conventions.
• In addition, and for full marks, your solution must include the definition of constants and helper functions where appropriate.
• Unless otherwise indicated in the question you may use only the built-in functions and special forms introduced in the lecture slides from CS115 up to and including the modules covered by this assignment. A list of functions described in each module of the lecture slides can be found at https://www.student.cs.uwaterloo.ca/~cs115/built_in
• Download the interface file from the course web page to ensure that all function names are spelled correctly, and each function has the correct number and order of parameters.
• Read each question carefully for restrictions.
• Test data for all questions will always meet the stated assumptions for consumed values.
• Do not copy the purpose directly from the assignment description. The purpose should be written in your own words and include references to the parameter names of your functions.
• The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.
• Familiarize yourself with the Plagiarism Rules of the University at the following links:
  https://uwaterloo.ca/academic-integrity/basic-info
  https://uwaterloo.ca/library/get-assignment-and-research-help/academic-integrity/academic-integrity-tutorial
• Do not send any code files by email to your instructors or tutors. Course staff will not accept it as an assignment submission. Course staff will not debug code emailed to them.
• You may post general assignment questions using the discussion groups on Waterloo LEARN. Choose Connect → Discussions. Read the guidelines for posting questions. Do NOT post any code as part of your questions.
• Check Markus and your basic test results to ensure that your files were properly submitted. In most cases, solutions that do not pass the basic tests will not receive any correctness marks.
• Read the course web page for more information on assignment policies and how to organize and submit your work. Follow the instructions in the Style Guide.
• Your solutions should be placed in files a02qY.rkt, where Y is a value from 1 to 3.

Language level: Beginning Student
Coverage: Modules 1 and 2

• Refer to the String documentation found at:
1. Complete a Racket function called `sum-digits` that consumes \( n \), a natural number between 0 and 100000 (0 \( \leq \) \( n \) \( \leq \) 100000), and produces the sum of all digits in \( n \).

For example:
- \( \text{(sum-digits 58)} \rightarrow 13 \) because \( 5 + 8 = 13 \)
- \( \text{(sum-digits 32850)} \rightarrow 18 \) because \( 3 + 2 + 8 + 5 + 0 = 18 \)
- \( \text{(sum-digits 99999)} \rightarrow 45 \) because \( 9 + 9 + 9 + 9 + 9 = 45 \)

2. Complete a Racket function called `username` that consumes two non-empty strings, `first` and `last`, and produces the username by combining the first two characters in `first`, the last two characters in `last`, the length of `first`, and the length of `last`. The function should include the entire string in the case where a string is less than two characters long. Hint: you may need to use the `min` and/or `max` functions.

For example:
- \( \text{(username "J" "Tompa") \rightarrow "Jpa15"} \)
- \( \text{(username "M-argaret" "A") \rightarrow "M-A91"} \)
- \( \text{(username "Stephen-Olaleye" "Karius") \rightarrow "Stus156"} \)

3. Complete a Racket function called `conversion` that consumes two strings, `first` and `second`, and a natural number \( n \). The function produces the natural number computed by summing the values assigned to each character in `first` and `second` as follows:
   - each of the first \( n \) characters in each of `first` and `second` has value 2, and
   - each remaining character in each of `first` and `second` has value 5.

Hint: you may need to use the `min` and/or `max` functions.

For example:
- \( \text{(conversion "Waterloo" "Univ" 5)} \rightarrow 33 \)
- \( \text{(conversion "" "Johnie" 1)} \rightarrow 27 \)

4. For this question, you will perform step-by-step evaluations of Racket programs, by applying substitution rules until you either arrive at a final value or you cannot continue. You will use an online evaluation tool that we have created for this purpose. You do not need to hand in any files for this question.

To begin, visit this webpage: [https://www.student.cs.uwaterloo.ca/~cs115/stepping](https://www.student.cs.uwaterloo.ca/~cs115/stepping)

Note: the use of https is important; that is, the system will not work if you omit the s. This link can also be found on the CS115 course webpage, under the Assignments heading.

You will need to authenticate yourself using your Quest/WatIAm ID and password. Once you are logged in, try the “Warmup questions” under “CS 115 Assignment 2,” in order to get used to the system. Note the “Show instructions” link at the bottom of each problem. Read the instructions before attempting a question!

When you are ready, complete the four stepping problems in the “Assignment 2 questions” category, using the semantics given in class for Beginning Student. You can re-enter a step as many times as necessary until you get it right, so keep trying until you completely finish every question. All you have to
do is complete the questions online—we will be recording your answers as you go, and there is no file to submit. The basic tests for this assignment will tell you whether or not we have a record of your completion of the stepper problems.

Note however that you are not done with a question until you see the message “Question complete!” You should see this once you have arrived at a final value and clicked on “Simplest form” (or “Error,” depending on the question).

You should not use DrRacket’s Stepper to help you with this question for several reasons. First, as mentioned in class, DrRacket’s evaluation rules are slightly different from the ones presented in class, but we need you to use the evaluation rules presented in class. Second, in an exam situation, you will not have DrRacket’s Stepper to help you, and there will definitely be step-by-step evaluation questions on at least one of the exams.