• 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, your solution must include the definition of constants and helper functions where appropriate.
• 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.
• 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.

• Unless otherwise indicated in the question you may use only the built-in functions and special forms introduced in the lecture slides from CS15 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/~cs15/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 send any code files by email to your instructors or any other course staff. Course staff will not accept it as an assignment submission. Course staff will not debug code emailed to them.
• 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. Passing the basic tests only guarantees that your solutions can be automatically checked for correctness. It does not mean your solution is guaranteed to be correct for all test cases.
• 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 a03qY.rkt, where Y is a value from 1 to 3.

**Language Level:** Beginning Student

**Coverage:** Module 3

**Notes:**
• For all questions, you may assume that the years are on or after 1AD.
• The interface file contains constants that have been defined that identify the strings required for the months of the year and the days of the week. Your solutions should use these constants to ensure that the correct values are being exactly produced from your functions.
Q1. Write a function `is-leap-year?` which consumes year and produces `true` if the year is a leap year and `false` otherwise.
Example:

(is-leap-year? 2017) => false

The leap year rule is:
Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but the years that end with 00 are leap years if they are exactly divisible by 400. [citation: US Naval Observatory]

Q2. Write a function `day-of-year` which consumes a string containing a properly formatted date and produces which day of the year that date is. For example, Jan. 2 is always the 2nd day of the year. The string parameter, called date, will always have the following format:

```
MMM dd YYYY
```

Meaning there will be a three-letter upper-case abbreviation of the month, followed by a one or two digit day of the month, followed by a four-digit year (including leading zeroes, if necessary). The three letter abbreviation always consists of the first three letters of the month. For example, Christmas Day this year would be written as: “DEC 25 2017” and New Year’s Day is “JAN 1 2016”.

Example:

(day-of-year “DEC 25 2017”) => 359

You may use the `string->number` DrRacket function to convert a string to the equivalent number value. Example:

(string->number “23”) => 23

BONUS: A bonus can be awarded if your `day-of-year` method accounts for leap years.

Q3. Write a function `new-years-holiday` which consumes a number representing a year and produces what day of the week the New Year’s Day Statutory Holiday is. This is on January 1st when it falls between Monday and Friday, or on the following Monday if it’s on Saturday or Sunday. As examples, if January 1st is on a Tuesday, then the holiday will be on a Tuesday. If January 1st is on a Saturday, the the holiday will be on the following Monday.

Use an algorithm called Gauss’ method for determining the day of the week for January 1 of a given year. It is as follows:

\[
\text{rem}(1+5\times\text{rem}(A-1,4)+4\times\text{rem}(A-1,100)+6\times\text{rem}(A-1,400),7)
\]

Where \(\text{rem}(x,y)\) is the remainder after division of \(x\) by \(y\), and \(A\) is the year. Note that this algorithm produces 0 for Sunday, 1 for Monday, and so on, through to 6 for Saturday. You do not need to use constants for the magic numbers contained within this formula.
The expected output is one of “Sunday”, “Monday”, etc...

Example:
(new-years-holiday 2017) => “Monday” ; despite JAN 1 2017 being Sunday

Q4. 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

Note: the use of https is important; this 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/Watlam ID and password. Once you are logged in, if you haven’t already done so, 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 stepping problems in the “Assignment 3 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 steps 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.