Deadline: Wednesday, January 29, 2020 at 10:00am on MarkUs

Language level: Beginning Student

Coverage: Modules 01, 02

Files to submit: `a02q1.rkt, a02q2.rkt, a02q3.rkt`

Assignment Guidelines

- Solutions for Questions 1-3 are expected to follow the requirements of the Style Guide: (`https://www.student.cs.uwaterloo.ca/~cs115/coursenotes1/styleguide.pdf`).

  This includes all relevant design recipe elements, proper use of constants, and proper use of helper functions.
- Submission details:
  - All solutions must be submitted through MarkUs. Solutions will not be accepted through email.
  - For Questions 1-3, verify your basic test results using MarkUs to ensure that your files were submitted properly and are readable on MarkUs.
    Note: passing the basic tests does not guarantee that you will pass all our correctness tests!
- Download the interface files 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.
- **Restrictions:**
  - Unless specifically allowed in the description of the question, you may only use the built-in functions and special forms introduced in the lecture slides in Modules 01 and 02. For details, see `https://www.student.cs.uwaterloo.ca/~cs115/#allowed`
  - In particular, **you are not allowed to use conditional statements** in your solutions for this assignment. Do not use `cond` on this assignment!
  - Read each question carefully to see if any additional restrictions apply.
  - Test data for correctness tests will always meet the stated assumptions for consumed values.
  - **Reminder:** Do NOT copy/paste text from the Assignment PDF to your definitions window. This can cause errors!
- 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.

Plagiarism: The following applies to all assignments in CS115.

All work in CS 115 is to be done individually. The penalty for plagiarism on assignments (first offense) is a mark of 0 on the affected questions and a 5% reduction of the final grade, consistent with School of Computer Science policy. In addition, a letter detailing the offense is sent to the Associate Dean of Undergraduate Studies, meaning that subsequent offenses will carry more severe penalties, up to suspension or expulsion.

To avoid inadvertently incurring this penalty, you should discuss assignment issues with other students only in a very broad and high-level fashion. Do not take notes during such discussions, and avoid looking at anyone else's code, on screen or on paper. If you find yourself stuck, contact the ISA or instructor for help, instead of getting the solution from someone else. Do not consult other books, library materials, Internet sources, or solutions (yours or other people's) from other courses or other terms.

Read more course policies at: `https://www.student.cs.uwaterloo.ca/~cs115/#policies`

1. A family is traveling on their annual vacation. They need to know the total cost of the vacation. Total cost includes the cost of air flight and additional daily costs. Air flight is $\$3,000$ for one adult, $\$2,000$ for a child between 2 and 12 years, and $\$200$ for an infant below 2 years. Each traveler incurs a fixed additional daily cost that is based on the number of vacation days (`days`), a daily rate (`rate`), and a discount amount (`discount`). It is assumed that the additional daily cost is never $0.

   Write a Racket function `travel-cost` that consumes four natural numbers `adults, children, infants,` and `days`, where `days` > 0, and two positive numbers `rate` and `discount`, where $0 <$ `discount` < 1. The function produces the total cost of the vacation by combining the cost of air flight and the total daily cost for all travelers, which is calculated as follows:

   ```
   daily-cost = days * (adults + children + infants) * rate * (1 − discount)
   ```

   Note that `discount` is applied on the daily cost only. There is no discount on air flight.

   Sample examples:

   - `(travel-cost 2 2 1 10 10 0.1)` ⇒ 10650
   - `(travel-cost 2 2 3 4 85.24 0.15)` ⇒ 12628.712

   Place your definitions into the file **`a02q1.rkt`**.

2. At Goose University, students are given a unique username based on their name and a special ID number. The username starts with the special ID number, followed by the first letter of their last name, and then followed by their first name. All of the usernames are restricted to a maximum of 9 letters, therefore it is often the case that only a portion of the first name is included. All of the usernames are in lower case. The special ID number is a positive integer that is less than 1000. The email address is the username with “@uragoose.ca” appended to it.

   Write a Racket function `goose-mail` that consumes two **non-empty** strings `first-name`, `last-name`, and a non-negative integer `ID-num`, where $0 \leq$ `ID-num` ≤ 999 and produces the email address for the student with the specified first and last names and ID number. The first name and last name are provided in lower case.

   Sample examples:

   - `(goose-mail “alexander” “bell” 303)` ⇒ “303balexa@uragoose.ca”
   - `(goose-mail “becky” “anderson" 12)` ⇒ “12abecky@uragoose.ca”

   Place your definitions into the file **`a02q2.rkt`**.

3. In $\mathbb{R}^2$, the Cartesian plane (or x, y-plane), if you are given two distinct points A $(a, 0)$ and B $(b, 0)$ on the x-axis and an angle θ, where $0° < \theta < 180°$, there exists a *unique* point $C(c, y)$ where y is positive and $\triangle ACB$ is an isosceles triangle, with $\angle ACB = \theta$. For simplicity, we assume $a < b$ here.

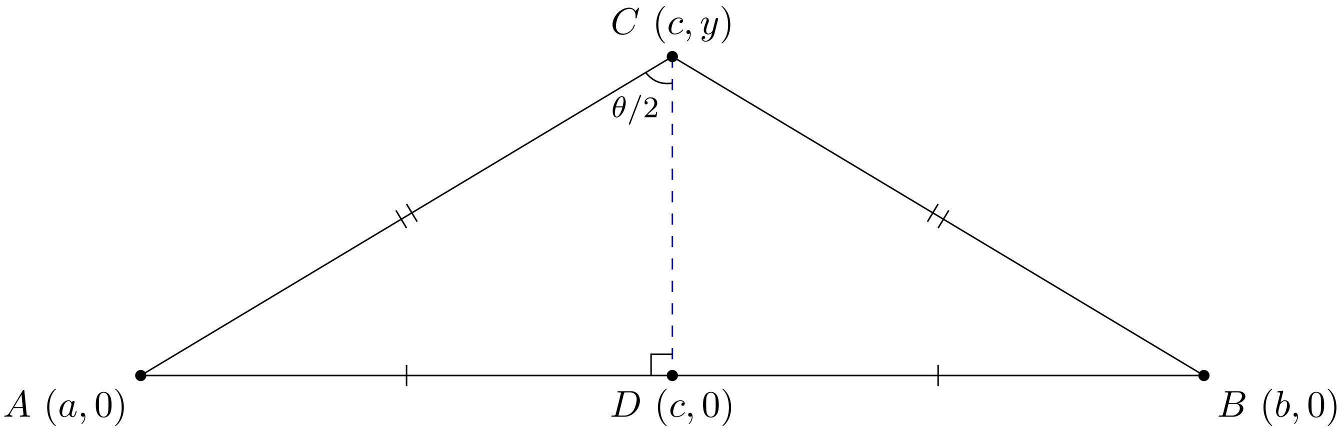
Write a Racket function `y-coord` that consumes two distinct numbers `a` and `b`, where `a` $<$ `b`, and one number `theta`, where $0 <$ `theta` < 180, and produces the y-coordinate of the point C (c, y) as defined above.

Sample examples:

- `(y-coord 1 5 60)` $\Rightarrow$ `3.4641016151377553`
- `(y-coord -2 6 120)` $\Rightarrow$ `2.309401076758504`

Some things to consider while performing your work:

- Racket's built-in trigonometric functions assume that the numbers consumed are in *radians*. To convert an angle from degrees to radians, you need to multiply your angle by $\frac{\pi}{180}$.
- It is strongly recommended that you use one or more helper functions to solve this problem. In particular, it is probably useful to create a helper function that converts degrees to radians.
- Bisecting $\angle ACB$ splits the triangle into two congruent triangles (and the bisector meets the line $\overline{AB}$ at its midpoint and forms a right angle!) We can label the midpoint of $\overline{AB}$ as D $(c, 0)$. A picture of the bisected triangle is included below.
- Recall from right-angle triangle trigonometry that, with the picture given below, we have $\tan\left(\frac{\theta}{2}\right) = \frac{\left|\overline{AD}\right|}{\left|\overline{CD}\right|}$, where $\left|\overline{AD}\right|$ and $\left|\overline{CD}\right|$ denote the lengths of $\overline{AD}$ and $\overline{CD}$, respectively.



TESTING YOUR SOLUTION:

Most of the values produced by the `y-coord` function will be *inexact*. As such, refer to Slides 25 and 26 in Module 02 on using the `check-within` form. We will use the same tolerance as specified in the slides, so checking output accuracy to rounding up to the nearest 0.00001. (i.e. rounded up to the nearest 10^{-5}).

For example, the first sample example from earlier would pass the following test:

- `(check-within (y-coord 1 5 60) 3.46410 0.00001)`

Place your definitions into the file **`a02q3.rkt`**.

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

   Note that 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 "CS115 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.