Homework 1

CMSC 15100 Introduction to Computer Science I

Your answers for this assignment should be written in a text format (using Microsoft Word, Notepad, LaTeX, etc) and submitted via Canvas before the start of the next lecture. From the next assignment onwards, you will use DrRacket to write your code.

When you write a function, you'll need to write several parts: type annotation, purpose, the code, and tests. On this assignment, this applies to problems 3, 4, 5, and 6.

Problems

1. (3 points) Write the following mathematical expressions as expressions using Racket's syntax (do not simplify the expressions). You can assume that all the variables have already been defined.

   (a) $\sqrt{151 - 7 \cdot 20}$

   (b) $\frac{GmM}{R^2}$

   (c) 2 if $b^2 - 4ac \geq 0$, otherwise 0

2. (3 points) Write a reasonable type annotation for each of the following functions.

   (a) `average-3`, which takes in three numbers and computes their average

   (b) `even-numerator?`, which takes in a fraction and outputs true if the numerator is an even number

   (c) `chicago-string`, which takes as input two strings, appends them with the word "chicago" between.

3. (4 points) Write functions `c->f` and `f->c` that convert temperatures from degrees Celsius to degrees Fahrenheit, and vice versa. Both of these functions should have the type `Exact-Rational -> Exact-Rational`.

4. (4 points) Write the function `not-a-percentage?` which takes one argument and determines if the input is either greater than 100 or less than 0. The type of this function should be `Real -> Boolean`.

5. (5 points) You just got a new debit card, which you plan to use right away to buy a fancy new dress (black and blue, of course). When you use the debit card, money will be taken from your bank account and used to pay for your purchase. However, if you have under $50 left in your account after you use the card, your bank will charge you a low-balance fee, which naturally you want to avoid. Further complicating things, the cost of the dress doesn't include tax (for example, if the dress price is $100 and there is 7% sales tax, you would pay $107 in total).

   Write a function `enough-money?` that takes the amount of money in your bank account, the cost of the dress, and the sales tax rate, and returns whether or not you have enough money to buy the dress and avoid the low-balance fee. The type of the function should be `Exact-Rational Exact-Rational Exact-Rational -> Boolean`.

6. (6 points) It's summer time, so let's make some lemonade! Your lemonade recipe uses one third of a lemon for every adult and one quarter of a lemon for every child who will be drinking it. Of course, you can't buy a partial lemon at the store, so you'll need to round up.

   Write the function `lemons` which takes in two parameters: the number of adults and the number of children who will be drinking your lemonade, and yields the number of whole lemons you need to buy to make the drink. Each of these values must be an `Integer`. In other words, the function's type must be `Integer Integer -> Integer`. The function should not tell you to buy more lemons than are strictly necessary; at worst, some fraction of a single lemon should be left over after you make your drink.

   In your implementation, you might find the following built-in function to be helpful: `exact-ceiling`, which consumes a number and produces the nearest integer equal to or greater than it.

   You notice that on particularly hot days, everyone drinks more lemonade. If the temperature outside is less than 85°F, then you need the same number of lemons as before. But if the temperature is greater than or equal to 85°F, then you need one half of a lemon for every adult and one third of a lemon for ever child. Write a second function, `more-lemons`, that takes three inputs: the number of adults, the number of children, and the outside temperature, and yields the number of lemons you need to purchase. The type of this function should be `Integer Integer Exact-Rational -> Integer`