Assignment 1
Due 4:00 PM on Monday, October 4

For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch. Do not copy the purpose directly from the assignment description. The purpose should be written in your own words and include reference 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. Test data for all questions will always meet the stated assumptions for consumed values.

Please read the course Web page for more information on assignment policies and how to organize and submit your work. Be sure to download the interface file from the course Web page and to follow all the instructions listed in the style guide (on the Web page). Specifically, your solutions should be placed in files `a1qY.rkt`, where Y is a value from 1 to 4. For full marks, it is not sufficient to have a correct program. Be sure to follow all the steps of the design recipe, including the definition of constants and helper functions where appropriate.

Language level: Beginning Student.
Coverage: Modules 1 & 2

1. Write a function called `calc-bmi` that will calculate a person's body mass index (BMI) based on their height in feet and inches and their weight in pounds. A person's BMI can be calculated as follows: mass/height^2, where the mass is measured in kilograms and the height is measured in metres. However, many people in North America only know their height in feet and inches and their mass in pounds. For example they might say 5 feet, 6 inches tall and 150 pounds. Your function will consume a height in feet, a number of inches given as an integer between 0 and 11 inclusive, and a mass in pounds. Use the approximate values of 2.2 pounds for one kilogram, and 2.54 centimetres for one inch. Use the actual values of 12 inches per foot and 100 centimetres per metre. For example, `(calc-bmi 5 6 150)` will produce approximately `24.26126`. To make testing easier, you can compare the value produced by your function to an approximation of BMI. For this question only, you may use `check-within` with a tolerance of 0.0001.

2. When training for long distance runs people often use the run/walk method. This means they will run for some number of minutes and then walk for one minute to recover and then start running again. This cycle continues until the end of the run. As they improve their training, the length of time for the running portion will increase; however the walking time will always be one minute. Write a function called `calc-distance` that will determine the distance traveled by someone who is training using this method. Your function will consume the walking rate of the person in km/hour, the running rate of the person in km/hour, the running interval time in minutes, and the total time

running/walking in minutes. Recall that rate = distance/time. You may assume that the total time training will be a time that has the person running at the end, not walking. For example `(calc-distance 6.3 12.5 4 52)` will produce `9.8` where the person ran for 4 minutes then walked for 1 minute ten times, and ended running for 2 minutes and was able to travel a total of 9.8 km. You may assume all times are non-negative integers.

3. Scientific notation is a way of writing very large or very small numbers using a few digits (known as the significant digits) and a power of ten. For example 1.475×10^9 is equivalent to 1 475 000 000. The number of digits included before the exponent are known as the significant digits. In the previous example there are four significant digits. Write a function called `sn->num` that consumes a string representing the scientific notation version of a positive number and the number of significant digits and produces the equivalent numeric value. You may assume that all numbers written in scientific notation will include a single digit before a decimal point, at least one digit following the decimal point, the characters `"x10^"` (meaning "times 10 to the power of"), and the exponent that may be any integer. For example `(sn->num "1.457x10^9" 4)` will produce `1457000000` and `(sn->num "1.23456x10^-4" 6)` will produce `0.000123456`. The built-in function `expt` will allow you to calculate an exponent. The built-in function `string->number` will produce the numerical equivalent of a consumed string that contains a valid number. For example `(string->number "1.23")` produces `1.23` and `(string->number "-5")` produces `-5`.

4. In August 2010, the University of Waterloo hosted the International Olympiad in Informatics (http://www.ioi2010.org). Every contestant, leader, and guest was given a bag of swag items when they arrived. The bags were assembled in the Math building and later transported to Village 1. The packing room had large, medium, and small boxes that could hold 10, 8, or 6 of the bags respectively for transportation. Write a function called `total-boxes` that will determine the total number of boxes used to transport the bags. The function will consume the total number of bags needing to be transported, the number of large boxes, the number of medium boxes, and the number of small boxes. You should use up the boxes that hold 10 first, then if necessary use boxes that hold 8, then if necessary use boxes that hold 6 bags. Note that you may not need to use any boxes holding 8 or 6 bags, and it is possible that the final box used is not filled to capacity. You may assume that there are at least enough boxes available to transport all of the bags. For example `(total-boxes 1000 50 50 50)` will produce `117`. The built-in functions `min`, `max`, and/or `ceiling` may be useful. You may **not** use `cond` in your solution for this question.