Assignment:	1
Due:	Thursday, May 21 at 9:00 am
Language level:	Beginning Student
Coverage:	Module 1

For this assignment only, you are not required to use the entire design recipe when writing functions. For each function, a well-written definition (header and body) is sufficient. For your convenience, an interface file which contains the headers of the required functions is available on the course webpage.

Do not send any code files to course staff; they will not be accepted. Submissions must be made via MarkUs as described on the course webpage. After submission, check your basic test results to ensure your files were properly submitted. Solutions that do not pass the basic tests are unlikely to receive any correctness marks.

Remember, the solutions you submit must be **entirely your own work**.

1. The value of an investment on which compound interest accrues is given by the expression

   $$P\left(1+\frac{r}{n}\right)^{nt}. \tag{1}$$

   Here P is the principal amount invested, r is the annual rate of interest (a number normally between 0 and 1), n is the number of times interest is compounded each year, and t is the number of years of the investment.

   For example, after 5 years a \$1000 investment with 10% interest compounded monthly is worth

   $$\$1000\left(1+\frac{0.1}{12}\right)^{12\times 5} \approx \$1645.31.$$

   You are curious how much more profitable an investment is when it has interest compounded daily instead of annually. Write a Racket function *extra-interest* which will compute this value for you. Given the principal amount invested, the annual interest rate, and the number of years of the investment (*principal*, *interest-rate*, and *num-years*, in that order), the function should produce how much extra interest is earned if the interest is compounded daily instead of annually. For simplicity, assume that there are exactly 365 days in a year, and perform the calculations without any rounding.

   Suggestion: Write a function *compound-interest* which calculates the expression in (1) and use that to compute *extra-interest*.

2. Neglecting air resistance, a BASE jumper has a velocity $v = \sqrt{2dg}$ after jumping off of a building and travelling a distance of d straight downward. Here g is the acceleration due to gravity, which may be assumed to have a value of $9.81\text{m}/\text{s}^2$ downward.

   (a) Write a Racket function *velocity* which consumes d, the distance travelled in metres, and produces the downward velocity of the jumper in metres per second. (Note this quantity will be nonnegative.)

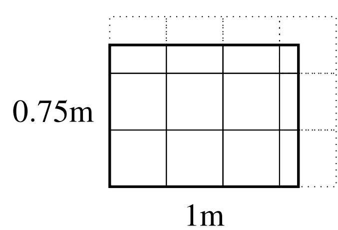
   (b) Write a Racket function *distance-travelled* which consumes the downward velocity v of the jumper in metres per second and produces how far the jumper has travelled in metres.

   (c) Write a Racket function *kph→mps* which consumes a velocity in kilometres per hour and produces the same velocity but represented in metres per second. You will need to use the fact that there are 1000 metres in a kilometre and 3600 seconds in an hour. (Note that in Racket code, the arrow is expressed by the characters ->.)

   (d) Write a Racket function *building-height* which consumes the downward velocity a jumper wants to reach in kilometres per hour and the height at which they will open their parachute in metres (v and *parachute-height*, in that order) and produces the height of the building in metres for which they will reach velocity v when *parachute-height* metres above the ground.

   Hint: Your answer can make use the functions defined in (b) and (c).

3. You are tiling the floor of a rectangular room with square tiles with a side length of 0.3m. The room will be tiled in the simplest manner possible, namely, by placing one tile in the corner of the room and working outwards. Because the dimensions of the room may not be evenly divisible by the tile length, it may be necessary to cut the tiles along the edge so that they will fit in the room. Once a tile has been cut, the remainder of the tile will be discarded.



   (a) Write a Racket function *num-tiles* which consumes the *length* and *width* of the room being tiled in metres, and produces the number of tiles which will be used to completely tile the room.

   Hint: You may find the built-in Racket function *ceiling* helpful (see the Racket documentation).

   (b) Write a Racket function *wasted-area* which consumes the *length* and *width* of the room being tiled in metres, and produces the area (in square metres) of the discarded parts of the tiles which were cut.