Assignment Guidelines.

- This assignment covers material in Module 3.
- Submission details:
  - Solutions to these questions must be placed in files `a02q1.rkt`, `a02q2.rkt`, `a02q3.rkt`, and `a02q4.rkt`, respectively, and must be completed using Racket *Intermediate Student*.
  - Unless otherwise indicated in the question you may use only the built-in functions and special forms introduced in the lecture slides from CS115 up to and including the modules covered by this assignment.
  - 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.
  - All solutions must be submitted to MarkUs. No solutions will be accepted through email, even if you are having issues with MarkUs.
  - Verify using MarkUs and your basic test results that your files were properly submitted and are readable on MarkUs.
  - For full style marks, your program must follow the CS115 Style Guide.
  - Be sure to review the Academic Integrity policy on the Assignments page.
  - For the design recipe, helper functions only require a purpose, a contract and an example.
- When a function returns an inexact answer, use a tolerance of 0.0001 in your tests.
- Restrictions:
  - Unless the question specifically describes exceptions, you are restricted to using the functions and special forms covered in or before Module 3.
  - Read each question carefully for additional restrictions.
- **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.**

1. An Approximation. You no doubt have encountered the number $\pi \approx 3.14$. π is irrational, so it is impossible to write down its exact decimal value. But we can approximate it. Here we will develop an approximation for π which involves the factorial.

(a) Factorial. The factorial of a natural number is $n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots (n-1) \cdot n$.

Exercise Write a function `(factorial n)` that consumes a `Nat` and computes $n!$.
For example,
```
(factorial 4) => 24
(factorial 0) => 1
```

(b) Odd-Double-Factorial. We can write a similar function $d(n)$ that is the product of the first n **odd** natural numbers. For example,

$$d(1) = 1 = 1$$
$$d(2) = 1 \cdot 3 = 3$$
$$d(3) = 1 \cdot 3 \cdot 5 = 15$$
$$d(6) = 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot 11 = 10395$$

(You can read probably more than you want about this function on the OEIS.)

Exercise Write the function `(odd-double-factorial n)` that consumes a `Nat` returns the product of the first `n` odd natural numbers.
```
(check-expect (odd-double-factorial 6) 10395)
(check-expect (map odd-double-factorial (range 0 8 1))
              (list 1 1 3 15 105 945 10395 135135))
```

(c) The upper crust number. It turns out that π may be approximated as

$$\pi = 2\left(\frac{1}{1} + \frac{1}{1 \cdot 3} + \frac{1 \cdot 2}{1 \cdot 3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{1 \cdot 3 \cdot 5 \cdot 7} + \ldots\right)$$

(Notice that each numerator is a factorial, and each denominator is an odd double factorial.)

Exercise Write a function `(approx-pi n)` that consumes a `Nat` and returns the approximation of π using `n` terms. For example, `(approx-pi 1) => 2` and `(approx-pi 2) => 2.`$\overline{6}$.
```
(check-expect (approx-pi 1) 2)
(check-expect (approx-pi 2) (* 2 (+ 1 (/ 1 3))))
(check-expect (approx-pi 3) (* 2 (+ 1 (/ 1 3) (/ 2 15))))
(check-within (approx-pi 20) 3.14159 0.00001)
```

2. Area under a Curve. You likely know the formula for the area of a rectangle: it is the product of the width and height, $b \cdot h$.

There are formulas for many other shapes, but some shapes don't have an area formula. Sometimes it is enough to *approximate* the area of a figure.

The area between a curve and the x-axis may be approximated by summing the areas of a set of rectangles.

For example, the area under the curve $y = x^2 + 1$ between $x = 2$ and $x = 4$, shown in Figure 1, is approximately

$$5 \cdot 0.5 + 7.25 \cdot 0.5 + 10 \cdot 0.5 + 13.25 \cdot 0.5 = 17.75$$

since the widths are all 0.5, and the heights are 5, 7.25, 10, and 13.25.

We can get a better approximation using more rectangles, and we can get as close as we like to the exact answer just by using more and more rectangles.

(This method of approximating area is called a Riemann sum.)

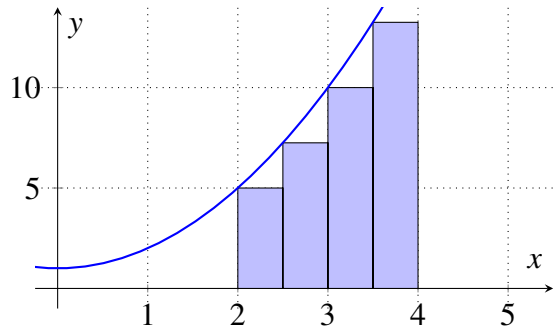

FIGURE 1. A plot of $y = x^2 + 1$

Exercise Write a function `(approx-area F xmin xmax nsteps)` that uses this technique to approximate the area under a function `F`, using `nsteps` steps between `xmin` and `xmax`.

Here are a few sample functions for testing. Make your solution work for these, but also for other functions. You might try $y = 2x$, for example. *Pick some of your own.*

```
(define (square-plus-one x) (+ (* x x) 1))
(define (constant-function-3 x) 3)
```

Use the following contract for `approx-area`:

```
;; approx-area: Function Num Num Nat -> Num
;; Requires: F has contract (Num -> Num).
```

Here are some tests using these functions:

```
(check-expect (approx-area square-plus-one 2 4 4) 17.75)
(check-expect (approx-area constant-function-3 1 5 100) 12)
```

3. Inquiry: next question up is recursive, yes?.

Exercise

Write a function `acronymize` that consumes a `(listof Str)`, where each `Str` is of length at least 1, and returns a `Str` containing the first letter of each item in the list.

```
(check-expect (acronymize (list "Portable" "Network" "Graphics")) "PNG")
(check-expect (acronymize (list "GNU's" "Not" "UNIX")) "GNU")
(check-expect (acronymize (list "Inquiry:" "next" "question" "up"
                                "is" "recursive," "yes?"))
              "Inquiry")
```

4. An Average Question. An average, or *mean*, is a number that is in some way near the "middle" of the set of values. But different "middles" make sense in different circumstances.

(a) Average. Usually when we say "mean" or "average" we mean the *arithmetic mean*, which is the sum of the items divided by the number of items. For example, the arithmetic mean of $[4,7,13]$ is $\frac{4+7+13}{3} = 8$; the arithmetic mean of $[5,5,5,5]$ is $\frac{5+5+5+5}{4} = 5$.

Exercise

Write a function `(mean L)` that returns the arithmetic mean of a non-empty `(listof Num)`.

```
(mean (list 16 4 2 2)) => 6
```

(b) Root Mean Square. The Root Mean Square of a series of measurements is of great importance to Physicists, Electrical Engineers, Biologists, and others.

RMS is calculated as the square root of the mean of the square of the values.

So for example, the RMS of $[1,2,-2,3]$ is

$$\begin{aligned}
&\sqrt{\frac{1}{4}((1)^2+(2)^2+(-2)^2+(3)^2)} \\
=&\sqrt{\frac{1}{4}(1+4+4+9)} \\
=&\sqrt{\frac{18}{4}} \\
=&\sqrt{4.5}
\end{aligned}$$

Exercise

Write a function `(root-mean-square L)` that consumes a `(listof Num)` and returns the Root Mean Square of the values.

```
(check-expect (root-mean-square (list 1 1 -1 1)) 1)
(check-expect (root-mean-square (list 2 2 -2 2)) 2)
(check-expect (root-mean-square (list -5 -1 -1)) 3)
(check-within (root-mean-square (list 2 1 2)) (sqrt 3) 0.00001)
(check-within (root-mean-square (list 1 2 -2 3)) (sqrt 4.5) 0.00001)
```