Assignment Guidelines.

- This assignment covers material in Module 3.
- Submission details:
  - Solutions to these questions must be placed in files a2q1.rkt, a2q2.rkt, a2q3.rkt, and a2q4.rkt, respectively, and must be completed using Racket.
  - 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. A list of functions described in each module of the lecture slides can be found at [https://www.student.cs.uwaterloo.ca/~cs115/built_in](https://www.student.cs.uwaterloo.ca/~cs115/built_in)
  - 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. Catalan Numbers. The Catalan Numbers are a sequence of numbers that are often used in combinatorics. The nth Catalan number is given by
\[
\frac{(2n)!}{n!(n+1)!}
\]
(a) Factorial. The factorial of a natural number is \( n! = 1 \times 2 \times 3 \times 4 \times \ldots (n-1) \times n \).

Write a function \((\text{factorial } n)\) that consumes a \text{Nat} and computes \( n! \).

For example,
\[ (\text{factorial 4}) \Rightarrow 24 \]

(b) Catalan.

Write a function \((\text{catalan-list } n)\) that returns a list of the first \( n \) Catalan numbers.

For example,
\[ (\text{catalan-list 5}) \Rightarrow (\text{list} 1 1 2 5 14) \]

2. Numerical integration. The area under a curve may be approximated by summing the areas of a set of rectangles. This is known as a Riemann sum. The area of each rectangle is the product of the height and width.

For example, the area under the curve \( y = x^2 + 1 \) between \( x = 2 \) and \( x = 4 \), shown in Figure 1, is approximately
\[ 5 \times 0.5 + 7.25 \times 0.5 + 10 \times 0.5 + 13.25 \times 0.5 = 17.75 \]
since the widths are all 0.5, and the heights are 5, 7.25, 10, and 13.25.

Write a function \((\text{riemann-sum } F \ x_{\text{min}} \ x_{\text{max}} \ n_{\text{steps}})\) that uses a Riemann sum to approximate the area under \( F \), using \( n_{\text{steps}} \) steps between \( x_{\text{min}} \) and \( x_{\text{max}} \).

\[
\text{(define (square-plus-one x) (+ (* x x) 1))}
\]
\[
(\text{riemann-sum square-plus-one 2 4 4}) \Rightarrow 17.75
\]
\[
(\text{riemann-sum (lambda (x) 3) 1 5 100}) \Rightarrow 12
\]

Use the following contract for \text{riemann-sum}:
\[
;; \text{riemann-sum: Function Num Num Nat -> Num}
\]
You may assume \( x_{\text{min}} \leq x_{\text{max}} \), but include that as a requirement.
3. Strings and Sentences.

**Exercise**
Write a function \( \text{join}(L) \) that consumes a non-empty \((\text{listof Str})\), and returns the \(\text{Str} \) that results from appending all the values in \(L\), with a space (" ") between them. For example,
\[
(\text{join} \ ((\text{list} \ "Hello" \ "how" \ "are" \ "you?"))) => \"Hello how are you?\"
\]

**Hint**
Read the documentation on \texttt{first} and \texttt{rest}.


Two values important in statistics are the \textit{mean} and \textit{standard deviation}.

The mean \( \mu \) of a list of \(n\) values is the sum of the values divided by the number of values. That is,
\[
\mu = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

\( (a) \) \textit{Mean}. Write a function \( \text{mean}(L) \) that returns the mean of a non-empty \((\text{listof Num})\)
\[
(\text{mean} \ ((\text{list} \ 4 \ 5 \ 6))) => 5
\]
\[
(\text{mean} \ ((\text{list} \ 4 \ 12 \ 4 \ 4))) => 6
\]

To compute the \textit{standard deviation}, calculate the sum of the squares of the differences from the mean, divide by the number of items, then take the square root of that.
\[
\sigma = \sqrt{\frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + (x_3 - \mu)^2 + \ldots + (x_n - \mu)^2}{n}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}
\]

For example, for \((\text{list} \ 4 \ 12 \ 4 \ 4)\):
- the mean is 6
- the differences from the mean are \((\text{list} \ -2 \ 6 \ -2 \ -2)\)
- the squares of these are \((\text{list} \ 4 \ 36 \ 4 \ 4)\)
- their sum is 48
- dividing by \(n = 4\) gives 12, so
- the standard deviation is \(\sqrt{12}\).

\( (b) \) \textit{Standard deviation}. Write a function \( \text{std}(L) \) that consumes a non-empty \((\text{listof Num})\) and returns the standard deviation.
\[
(\text{std} \ ((\text{list} \ 4 \ 12 \ 4 \ 4))) => \#i3.4641016151...
\]
\[
(\text{std} \ ((\text{list} \ 0 \ 0 \ 10 \ 10))) => 5
\]