Extra Practice Problems (Module 2)

You may use only features covered until the end of M2

1. Create a function `add-num-to-str` that consumes a string `st` and a positive integer `n` and produces a string by adding `n` to the end of `st`. Example:
   ```lisp
   (add-num-to-str "CS" 115) => "CS 115"
   ``

2. Create a function `add-bonus` that consumes two numbers `grade` and `bonus`, where your grade is between 0 and 100 and the bonus is between 1 and 10. The function produces a new grade not exceeding 100%. Examples:
   ```lisp
   (add-bonus 98 5) => 100
   (add-bonus 60 4) => 64
   ``

3. Create a function `sum-digits` that consumes a positive integer `n` and produces the sum of the digits of `n`. Examples:
   ```lisp
   (sum-digits 100) => 1
   (sum-digits 143) => 8
   ``

4. Create a function `official` that consumes two strings, `fname` and `lname`, and produces the last name followed by a comma and the initial of the first name followed by a period. Example:
   ```lisp
   (official "Mbabi" "Tema") => "Tema, M."
   ``

5. Create a function `introduction` that consumes one string `fname` and produces an introductory sentence. Example:
   ```lisp
   (introduction "Bettina") => "Hello, my name is Bettina"
   ``

6. Create a function `sodd?` that consumes a string `s` and produces `true` if the length of `s` is odd and `false` otherwise. Examples:
   ```lisp
   (sodd? "CS") => false
   (sodd? "uni") => true
   ```
7. Write the Purpose, Contract (& requirements), Examples and Tests of the following two functions (i.e. part a and b):

a. 

```lisp
(define (root-plus a b c)
  ; (+ (* -1 b)
  (sqrt (- (sqr b) (* 4 a c))))
  (* 2 a)))
```

b. 

```lisp
(define (root-minus a b c)
  ; (- (* -1 b)
  (sqrt (- (sqr b) (* 4 a c))))
  (* 2 a)))
```

c. Write a function `x-values` that consumes three numbers `(a, b, c)` and produces a string in the following format:

"For `ax^2 + bx + c = 0`, the values of `x` which are the solutions of the equation are **x-value1** and **x-value2**."

Note: Where **x-value1** is the bigger root of the equation and the **x-value2** is the smaller root.

Example:

```
(x-values 1 2 -15) => "For 1x^2 + 2x + -15 = 0, the values of x which are the solutions of the equation are 3 and -5."
```

**Hint:** Use the function `number->string` to convert numbers to strings

8. Write a function `pad3` that consumes a natural number `n` and produces a `Str`. The produced `Str` contains the most left 3 digits of `n`. For `n` that has less than 3 digits, zeros are added at the left to make it of length 3.

Examples:

```
(pad3 7) => "007"
(pad3 42) => "042"
(pad3 245) => "245"
(pad3 3141592) => "314"
```