1. **Substitution**

Consider the example below. Notice that \( x \) is used in multiple places. When do we substitute for \( x \) and when don't we?

\[
\text{(define x-y*y (lambda (x y) (- x ((lambda (x) (* x x)) y))))}
\]

Use the substitution model to evaluate the following expression, and write each substitution step.

\[
(x-y*y 11 3)
\]

Value: ____

2. **Recursion**

2.1. a. Implement addition as a recursive procedure that employs repeated successor. In Scheme, this is the inc function, which increases its argument by 1; dec decreases its argument by 1.

(Hint: check for base case, then recursive case.)

\[
\text{(define (add x y)}
\]

b. Write the first 4 substitution steps for \( \text{add} \ 3 \ 2 \)
2.2. Implement subtraction as a recursive procedure that employs the `dec` function, which decreases its argument by 1.

(define (sub x y))

2.3. Implement exponentiation through repeated multiplication.

(define (expt x n)
  <your code will go here>)

a. recursive algorithm

b. iterative algorithm (Hint: Define a helper function.)