WWSD:

scm> (+ 4 3)

\_\_\_\_\_\_

scm> (4 + 3)

\_\_\_\_\_\_

scm> (+ 1 2 3 4)

\_\_\_\_\_\_

scm> (- 1 2 3 4)

\_\_\_\_\_\_

scm> (if True (+ 2 3))

\_\_\_\_\_\_

scm> (if False (+ 2 3))

\_\_\_\_\_\_

scm> (if False (/ 1 0) 4)

\_\_\_\_\_\_

scm> (and True False True)

\_\_\_\_\_\_

scm> (and True 0 True)

\_\_\_\_\_\_

scm> (and 3 5 2)

\_\_\_\_\_\_

scm> (or True False True)

\_\_\_\_\_\_

scm> (or False 4 True)

\_\_\_\_\_\_

scm> (define x 2)

\_\_\_\_\_\_

scm> x

scm> (define (f x) (\* x x))

\_\_\_\_\_\_

scm> (f 4)

\_\_\_\_\_\_

scm> (define (g x) (lambda (y) (\* x y)))

\_\_\_\_\_\_

scm> ((g 4) 5)

\_\_\_\_\_\_

scm> (cons 1 (cons 2 nil))

\_\_\_\_\_\_

scm> (cons 1 2)

\_\_\_\_\_\_

scm> (car (cons 1 (cons 2 nil)))

\_\_\_\_\_\_

scm> (cdr (cons 1 (cons 2 nil)))

\_\_\_\_\_\_

scm> (cdr (cons 1 2))

\_\_\_\_\_\_

scm> (null? (cons 1 nil))

\_\_\_\_\_\_

scm> (null? (cdr (cons 1 nil)))

\_\_\_\_\_\_

scm> (list 1 2 3 4)

\_\_\_\_\_\_

scm> (cdr (list 1 2 3 4))

\_\_\_\_\_\_

CODE WRITING :D

(define (filter pred lst)

; YOUR CODE HERE

)

; Tests

scm> (define (less-3 x) (< x 3))

less-3

scm> (filter less-3 (list 1 2 3 4))

(1 2)

(define (construct value lists)

(if \_\_\_\_\_\_

\_\_\_\_\_\_

(cons \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_)))

)

; Tests

scm> (define lists '((10) () (4 3) (2 5 3)))

lists

scm> (construct 6 lists)

((6 1) (6) (6 4 3) (6 2 5 3))

scm> (construct 6 '())

()