Quiz 8a

1. (3 points) Fill in the blanks with the values of the expressions shown:

(define x 0)

(define mystery

(let ((x x)

(y 0))

(lambda ()

(set! y (+ y 1))

(set! x (+ x y))

x)))

(mystery) \_\_\_\_\_\_\_

(mystery) \_\_\_\_\_\_\_

(mystery) \_\_\_\_\_\_\_

x \_\_\_\_\_\_\_

2.(4 points) Write procedure msc (short for “make strange counter”) that returns a counter procedure. The first counter msc returns is a counter that starts from 1 and counts up by 1 at a time. Later on, when msc is called, it returns a counter that starts from k and counts up by k, where k is the largest number ever returned by *any* strange counters so far.

STk> (define c1 (msc)) ; msc is called for the first time

c1 ; c1 is a counter starting from 1, counting by 1

STk> (c1)

1

STk> (c1)

2

STk> (c1)

3

STk> (define c2 (msc))

c2 ; c2 is a counter starting from 3, counting by 3

STk> (c2)

3

STk> (c2)

6

STk> (c1) ; note it's c1 here

4

STk> (define c3 (msc))

c3 ; c3 is a counter starting from 6, counting by 6

3. (3 points)

Draw the environment diagram resulting from evaluating the following expressions, and show the result printed by the last expression where indicated.

> (define foo

(let ((var 5))

(define (max! var new)

(set! var (max var new))

var)

(lambda (x)

(define y (\* 2 x))

(max! y var)))))

> (foo 3)