Quiz 10b

1. (2 points) Louis Reasoner decides to implement his own version of streams. He writes the following code:

(define (cons-stream a b)

(cons a (delay b)))

(define(stream-car strm)

(car strm))

(define (stream-cdr strm)

(force (cdr strm)))

(define (integers-starting-from n)

(cons-stream n (integers-starting-from (+ n 1))))

He then types in (define ints (integers-starting-from 1)) and is shocked to find out that it doesn’t work. What happens? What is wrong with his implementation? **You must give an explanation to get full credit.**

2. (3 points) Create a stream odd-squares which is a stream of odd integers, followed by its square

> odd-squares

(1 1 3 9 5 25 7 49 9 81 …)

3. (3 points) Show the first 3 elements of each of these two streams:

(define (crazy x y)

(if (> x y)

x

(- x y)))

(define barney (cons-stream 1

(stream-map crazy barney stinson)))

(define stinson (cons-stream 2

(stream-map + barney stinson)))

barney: \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

stinson: \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

4. (2 points) Define the infinite stream updown where every element in the stream is a list

-> (ss updown)

( (1) (1 2 1) (1 2 3 2 1) (1 2 3 4 3 2 1)

(1 2 3 4 5 4 3 2 1) . . . )

Hint: define a helper function that does the following:

-> (updown-helper 1)

(1)

-> (updown-helper 6)

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