Quiz 0.3 Rubric

1. (3 points) The procedure first-letters below is missing a base case. What will happen if I call (first-letters ‘(this is a sentence)) ? You must explain your reasoning – only 1 point will be given for the correct answer, the remaining 2 points are for the explanation.

(define (first-letters sent)

(sentence (first (first sent)) (first-letters (bf sent))))

1 point for saying that it will error.

2 points for saying that the problem is that the procedure tries to use the recursive case even when sent is empty, which should be a base case. (first sent) will cause an error because first can only work on non-empty sentences (and words).

Note: If they say that there will be an infinite loop because of the missing base case, they can get up to 2 points.

1. (3 points) Write a procedure to reverse a sentence. **We will cut points for bad style!**

> (reverse ‘(anne i vote more cars race rome to vienna))

(vienna to rome race cars more vote i anne)

(define (reverse sent)

(if (empty? sent)

‘()

(sentence (reverse (bf sent)) (first sent))))

Grading: 1 point for base case, 1 point for the recursive call, 1 point for combining the recursive call with the first or last word (depending on implementation) in the right order using sentence. Take off 0.5 points for minor errors, such as domain-range errors like returning the empty word as the base case.

1. (4 points) Write a procedure progressive-squares? that takes a sentence of numbers as its argument. It should return #t if each number (other than the first) is the square of the number before it. **We will cut points for bad style!**

> (progressive-squares? '(3 9 81 6561))

#t

> (progressive-squares? '())

#t

> (progressive-squares? '(6))

#t

> (progressive-squares? '(25 36 49 64))

#f

(define (square x) (\* x x))

(define (progressive-squares? sent)

(cond ((or (empty? sent) (empty? (bf sent)))

#t)

((= (square (first sent)) (first (bf sent)))

(progressive-squares? (bf sent)))

(else #f)))

1.5 points for getting both base cases. 1 point if one is missing.

1.5 points for the test condition in the second cond clause.

1 point for the recursive call.

Same guidelines for dealing with minor errors.