Quiz 13a RUBRIC

1. (3 points) Write the logic rule(s) so that (list? \_\_\_\_\_\_\_\_) will be true if the thing in the blank is a proper list:

Query Input : (list? (a b c))

Query Output: (list? (a b c))

Query Input : (list? (a . b))

;; No Output

Query Input : (list? a)

;; No Output

(assert! (rule (list? ())))

(assert! (rule (list? (?car . ?cdr)) (list? ?cdr)))

-1 missing empty list rule

-1 forgot to add “dot”

-1 syntax error, but still understandable

-1 any other small mistakes

1. (3 points) Write a set of rules and/or assertions in the query system to implement an **odd-length** relation like this:

Query Input: (odd-length (she loves you))

Query Output: (ODD-LENGTH (SHE LOVES YOU))

Query Input: (odd-length (day tripper))

;; No Output

Queries using **odd-length** should succeed for lists containing an odd number of elements, and should fail for lists containing an even number of elements.

**Do not use lisp-value!**

(assert! (rule (odd-length (?a))))

(assert! (rule (odd-length (?car ?cadr . ?cddr)) (odd-length ?cddr)))

-1 missing one element case

-1 forgetting dot

-1 syntax error, but still understandable

Keep in mind that this isn’t the only solution. One could make a helper even rule and avoid having to the the ?cddr thing.

1. (4 points) You wish to write interleave (for finite lists) in the query evaluator. Here are some examples of interleave's behavior (you may assume it only needs to run forward):

Query Input : (interleave (a b) (1 2) ?what)

Query Output: (interleave (a b) (1 2) (a 1 b 2))

Query Input : (interleave (a b) (1 2 3) ?what)

Query Output: (interleave (a b) (1 2 3) (a 1 b 2 3))

Define all the rules necessary for interleave to work.

(assert! (rule (interleave () () ())))

(assert! (rule (interleave () ?x ?x)))

(assert! (rule (interleave ?x () ?x)))

(assert! (rule (interleave (?car1 . ?cdr1) (?car2 . ?cdr2) (?car1 ?car2 . ?x))

(interleave ?cdr1 ?cdr2 ?x)))

-1 for each base case missing

-1 if solution outputs same match more than once

-1 missing dot

-1 syntax error, but still understandable

-1 other mistakes