Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

**ICS 365. Organization of Programming Languages.**

**Programming Assignment 4**

**Due**: See Syllabus

**Points**: 40

1. (10 points)

Write a scheme function named up-to-first-char that takes a list as its input and returns a list containing all the elements up to the first character element in the input list. Hint: think of using some predicate function(s) in your logic.

Solution:

(define (up-to-first-char ls)

(define value)

(define (fn-helper lst)

(cond

[(null? lst) '()]

[(symbol? (car lst)) '()]

[else (cons (car lst)

(fn-helper (cdr lst)))]

))

(if (null? (fn-helper ls))

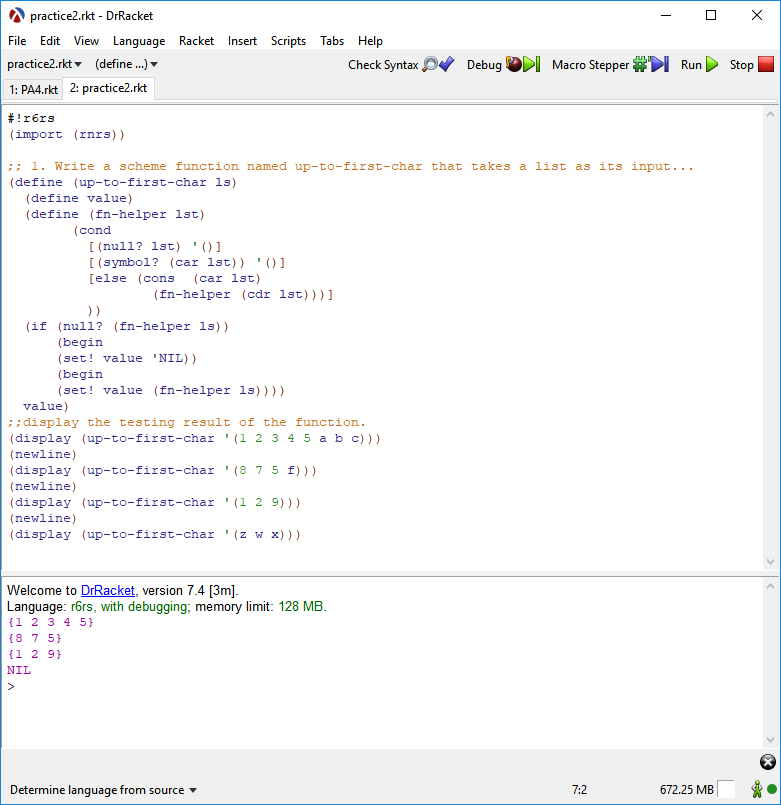
(begin

(set! value 'NIL))

(begin

(set! value (fn-helper ls))))

value)



Question 1 Screen shot result:

1. (15 points)

Write a Scheme function that takes a list and an atom as parameters and returns a list identical to its parameter list except with all top-level instances of the given atom deleted. Hint: recursive algorithm.

Solution:

(define (delete-top-level atom lst)

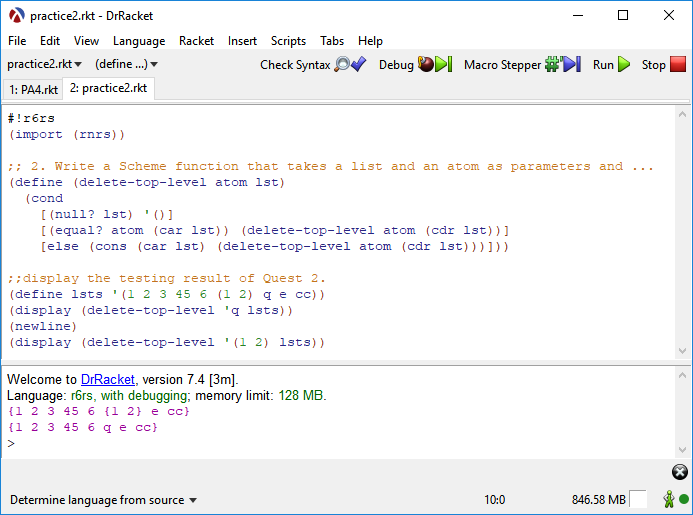
(cond

[(null? lst) '()]

[(equal? atom (car lst)) (delete-top-level atom (cdr lst))]

[else (cons (car lst) (delete-top-level atom (cdr lst)))]))

Question 2 Screen shot result:



1. (15 points)

Write a Scheme function that takes two atoms and a list as parameters and returns a list identical to the parameter list except all occurrences of the first given atom in the list are replaced with the second given atom, no matter how deeply the first atom is nested.

Solution:

(define (replace atom-1 atom-2 lst)

(cond

((null? lst) '() )

((list? (car lst)) (cons (replace atom-1 atom-2 (car lst)) (replace atom-1 atom-2 (cdr lst))))

((= (car lst) atom-1) (cons atom-2 (replace atom-1 atom-2 (cdr lst))))

(else (cons (car lst) (replace atom-1 atom-2 (cdr lst))))))

Question 3 Screen shot result:

