UTD

Karan Sahu
KXS 190007

CS/CE 4337 - Assignment#1
Due Date: 9/13/20, 11:59 pm

1-Draw the internal representation for the following lisp list(s).

- (cons '((apple (orange ()) banana)) '(()))
- (cons '((apple (()orange (grape)) banana)) '((apple ())))
- (cons '((() apple (orange () (grape)) banana)) '((apple)))

2-Determine the output of the following functions (You must show your works)

- (cadaar '(((() apple (orange () (grape)) banana)) (apple)))
- (cdadadr'(()(orange (apple (() orange (grape)) banana))))
- (cdadaar'(((apple (() () (grape))) (())))

3-Give combinations of cars and cdrs that will pick 7 from each of the following lists ,(1 (2 3) 7), (((1 (7)))), and (1(2(3(4(5(6(7)))))))

4-Fully explain the following scheme procedure.

```
(define (x lis)
  (cond
    ((null? lis) 0)
    ((not (list? (car lis)))
      (cond
        ((eq? (car lis) #f) (x (cdr lis)))
        (else (+ 1 (x (cdr lis))))))
    (else (+ (x (car lis)) (x (cdr lis))))))
```

5-Write a Scheme procedure that takes a list and returns the sum of the odd element less than 7 in the list. For example, (sumodd '(9 2 3 4 5)) returns 8 and (sumodd '(9 2 3 (4 5)) returns 8. Trace the function with one of the provided examples.

6-Write a Scheme procedure that takes a list and returns a list identical to the parameter except the element before the last element has been deleted. For example, (deleteitem '(a b c d e)) returns '(a b c e) ; (deleteitem '(a b (c d) e)) returns '(a b e). Trace the function with one of the provided examples.

7-Write a Scheme procedure that takes a list and returns the list created by reversing the list. For example, (newlist '((a b) (c d) a b)) returns '(b a (d c) (b a)) and (newlist '(a b (c) b d)) returns '(d b (c) b a). Trace the function with one of the provided examples.

8-Write a procedure that takes a tree (represented as a list) and returns a list whose elements are all the leaves of the tree arranged in right to left order. For example, (leaves '(((1 2) (3 4)) ((1 2) (3 4)))) returns '(4 3 2 1 4 3 2 1).

9-Define a scheme procedure that takes a list and returns the depth of the most nested parentheses in a list. For example, (EXP-DEPTH '(I J ((K) L) M)) returns 3

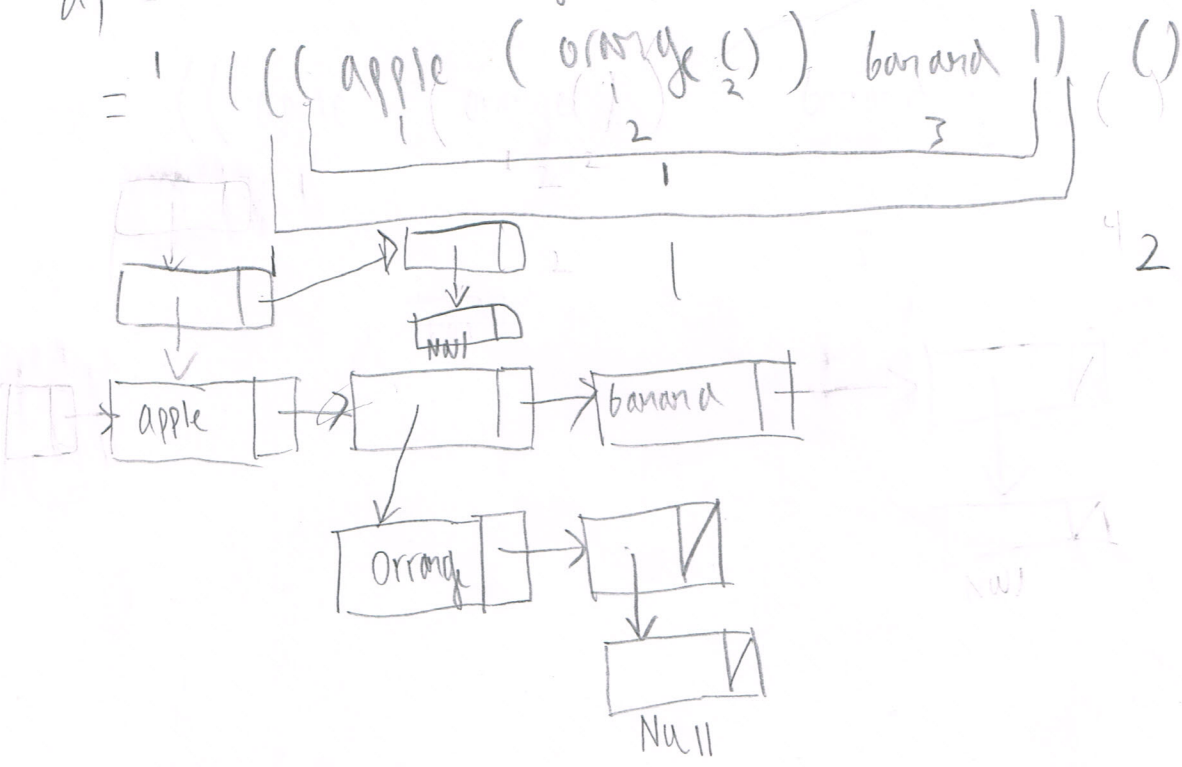
10-Define a scheme procedure that takes a set (a list of distinct elements) and generates a list of all subsets of the set. For example, (subsets '(a b c)) returns '((a b c) (b c) (a c) (c) (a b) (b) (a) ()).

Karan Sahu CS 3347 Assignment 1 KXS170007

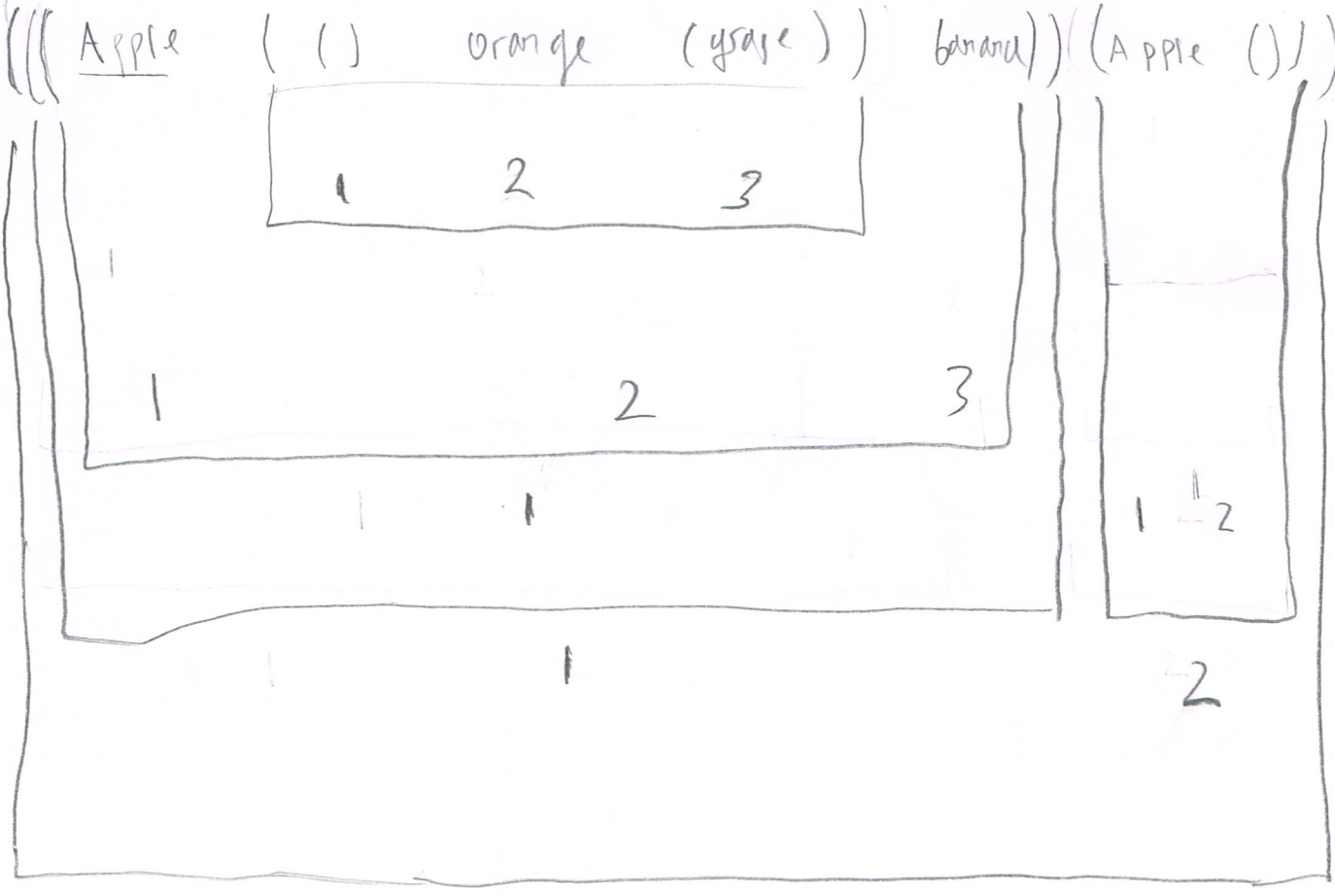
1.

a) (cons '((apple (orange ()) banana)) '(()))

= '(((apple (orange ()) banana)) (()))

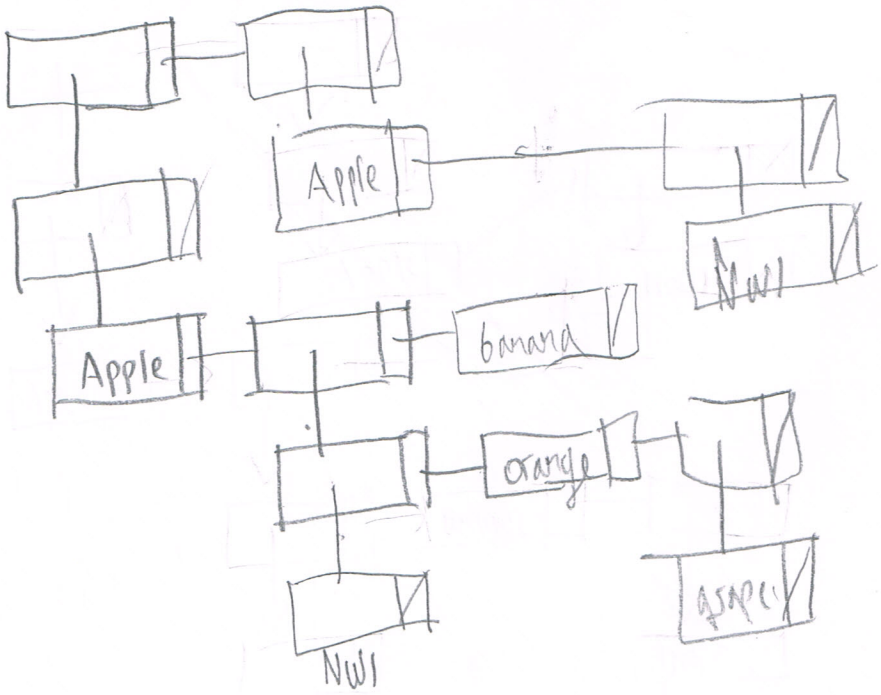


b) = '(((Apple (()) orange (grape)) banana)) ((Apple ()))

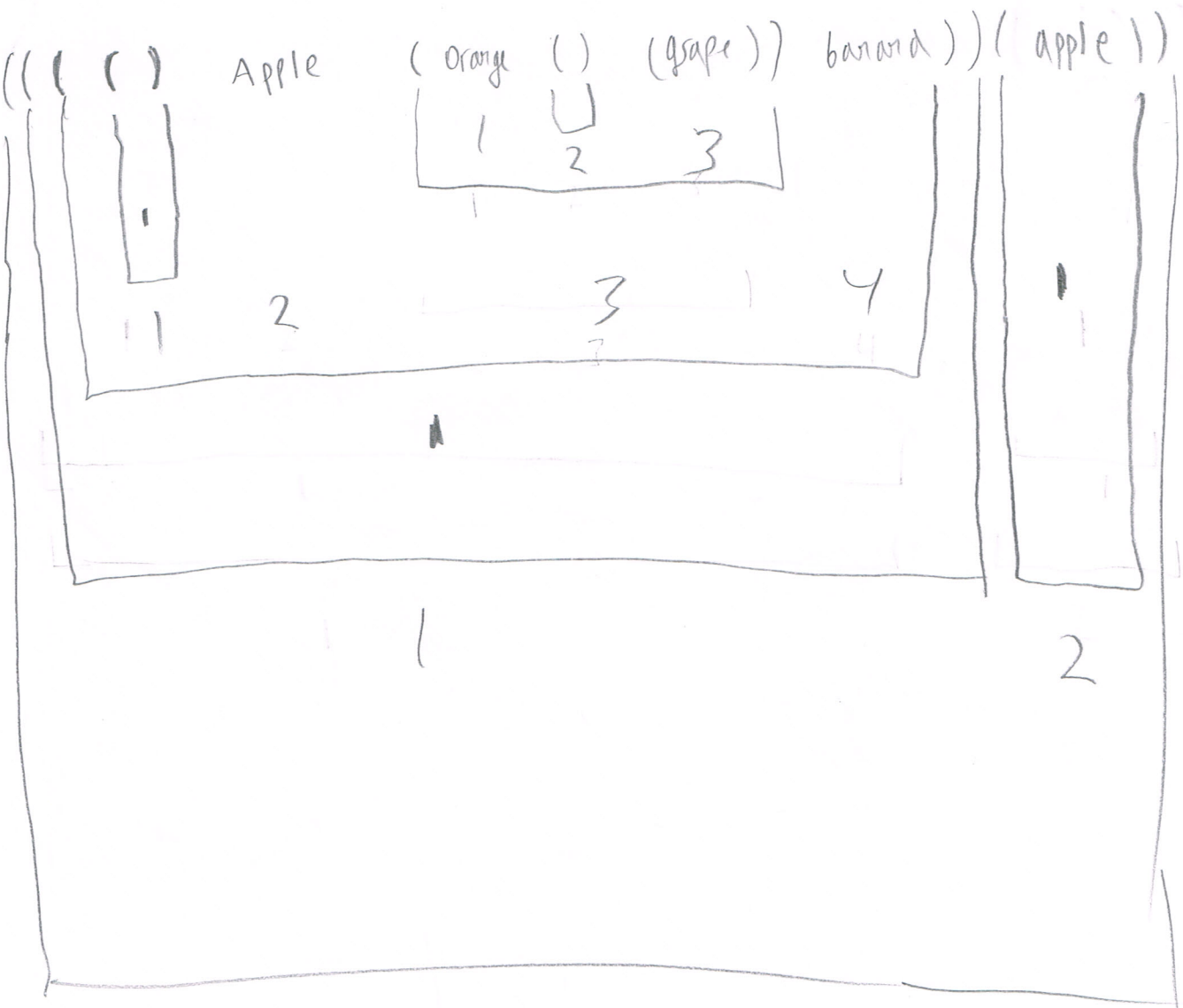


1 b)

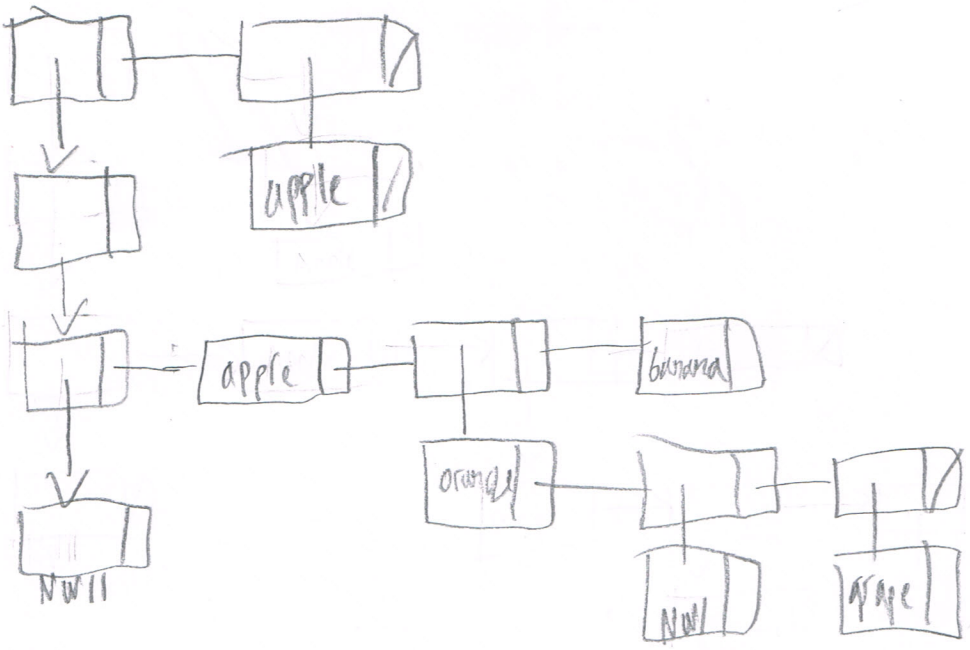
drawing



1 c)



1 c)



2. a)

(((( ) apple (orange ( ) (grape)) banana)) (apple))

1 2 3 4

1 2 3

1

cadaar

a - ((( ) apple (orange ( ) (grape)) banana)) (apple))

d - (((apple))

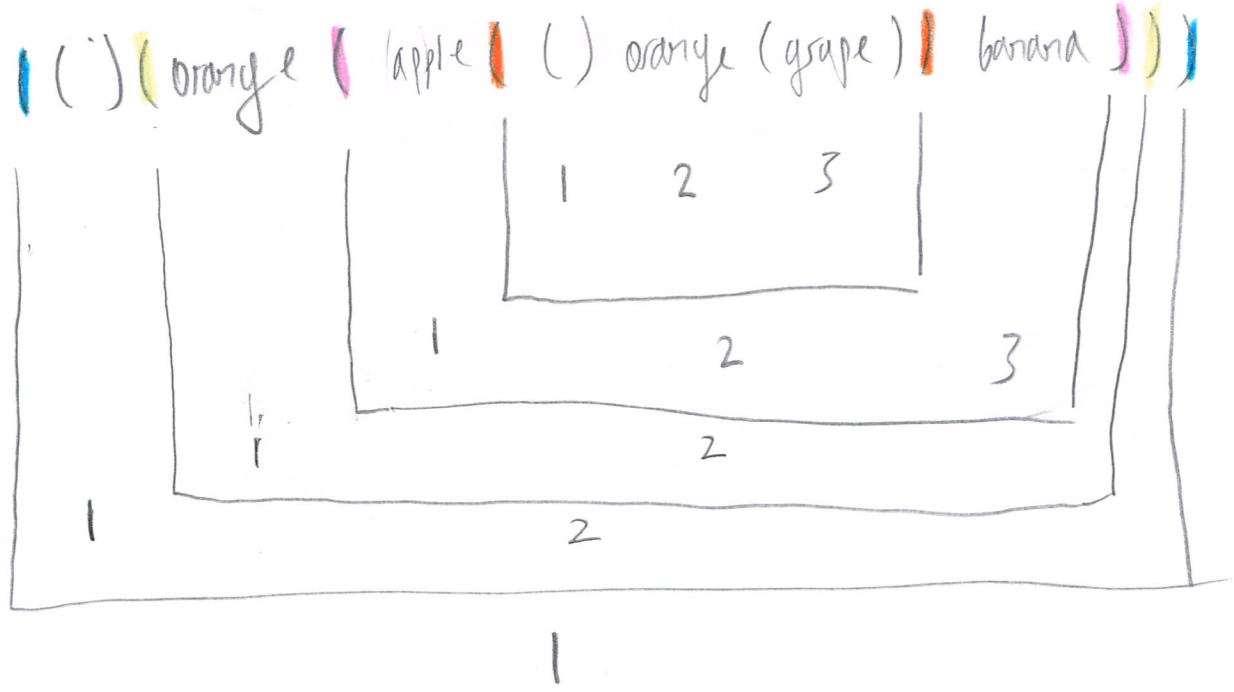
a - (apple)

a - apple

r done

Answer

2. 6) ( ) (orange (apple ( ) orange (grape)) banana))



cdadadr

d - ((orange (apple ( ) orany (grap)) banana))

a - (orange (apple ( ) orange (grape)) banana))

d - ((apple ( ) orange (grape)) banana))

a - (apple ( ) orange (grap)) banana)

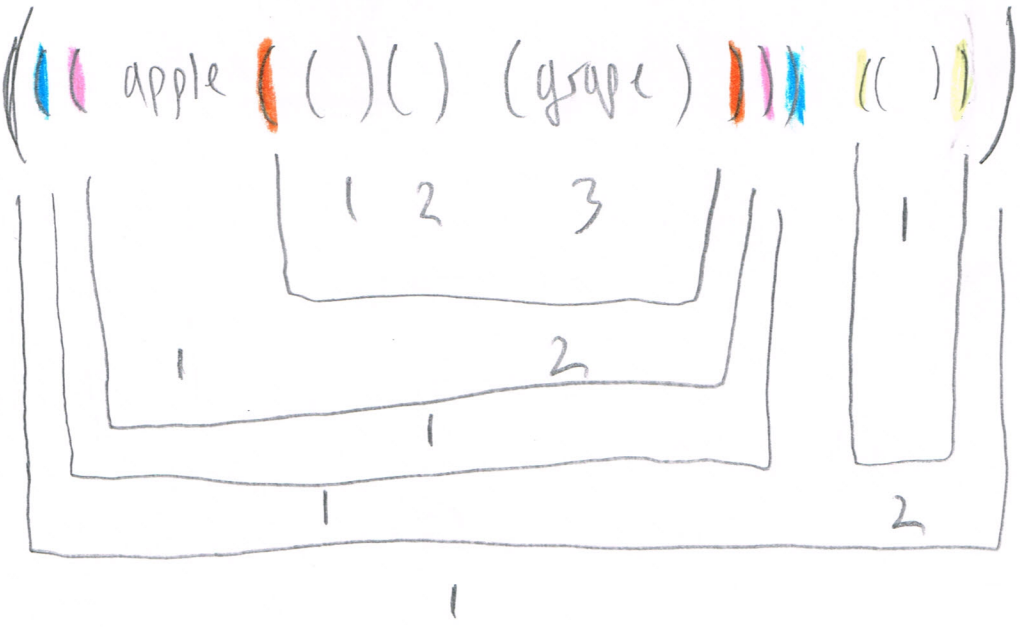
d - (( ) orange (grape)) banana)

r - ✓

Answer

✓

2 c)



cdadaar

d - (( ))

a - ( )

d - is an error

---

3. a) (1 (2 3) 7)

cddddr

b) (((1 (7))))

cadaaaar

c) ~~(1 (2 (3 (4 (5 (6 (7)))))))~~

3 c) (1(2(3(4(5(6(7)))))))

cdadadadadadaar

1 2 3 4 5 6

4) sums the elements and all the sub elements (elements within list which is in the list) of the List.

5) (sum-list-under 7- (9 2 3 4 5))

1pr

list? 9

false

2pr

5) (sumListUnder7 (9 2 3 4 5))

trace

1 pr null? (9 2 3 4 5)
#f

2 pr List? 9
#f

3 pr even? 9
#F

4 pr < 6 9 (sumListUnder7 (cdr LIST))
#T

1 Pr null? (2 3 4 5)
#f

2 Pr List? 2
#f

3 pr even? 2 (sumListUnder7 (3 4 5))
#t

1 Pr null? (3 4 5)
#f

2 Pr List? 3
#f

3 Pr even? 3
#f

4 pr < 6 3
#f

+8

+8

+8

#5

race

5pr List? (3 4 5)

#true

(+ 3 sumList(4 5))

1pr null? (4 5)

#f

2pr List? 4

#f

3pr even? 4

#true

sumList(5)

+5

1pr null? (5)

#f

2pr List? 5

#f

3pr even? 4

#f

4pr < 6 4

#f

5pr List (5)

true

(+ 5 sumList())

1pr null? ()

#true

0

+5

+0

6 (delete prelast (a b (c d) e))

trace 1pr null? (a b (c d) e)
#f

2pr = 4 2
#false

#t

✱ 3pr cons a + delete prelast (b (c d) e)
#t

1pr null? (b (c d) e)
#f

2pr = 3 2
#f

#t

✱ 3pr cons b + delete prelast ((c d) e)

1pr null? ((c d) e)

#t

✱ 2pr = 2 2
#t

e

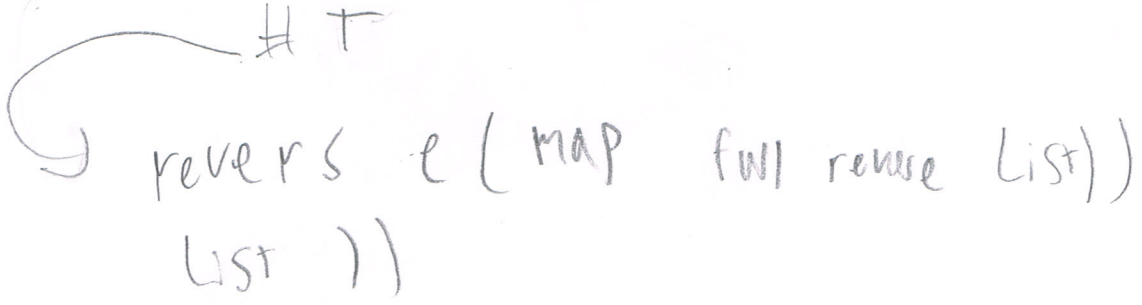
creates (a b e)

7. (fwl reverse ((a b) (c d) a b))

trace

(if (List? ((a b) (c d) a b))

#t



(map fwl reverses all elements in List)

(a b) -> #t rewt -> (b a)

(c d) -> #t rewt -> (d c)

a -> #f rewt -> a

b -> #f rewt -> b

fnw rewt. (b a (d c) (b a))