

Kerem Aksoy 64243

Ahmet Hakan Hafif 64092

PS 2 ANSWERS

Problem 1

$(\text{list } 1 (\text{list } 2 \ 3) \ 4) \rightarrow (1 \ (2 \ 3) \ 4)$

$(\text{list } (\text{list } (\text{list } 1 \ 2) \ 3 \ 4) \ (+ \ 2 \ 3) \ (- \ 8 \ 2)) \rightarrow (((1 \ 2) \ 3 \ 4) \ 5 \ 6)$

$(\text{list } (\text{list } '())) \rightarrow ((()))$

$(\text{cons } 1 \ 2) \rightarrow (1 \ . \ 2)$

$(\text{cons } 1 \ '(2)) \rightarrow (1 \ 2)$

$(\text{car } (\text{cons } 1 \ '(2))) \rightarrow 1$

$(\text{cdr } (\text{cons } 1 \ '(2))) \rightarrow (2)$

$(\text{car } (\text{cdr } (\text{cdr } '(1 \ 2 \ 3 \ 4)))) \rightarrow 3$

$(\text{car } (\text{car } (\text{cdr } (\text{cdr } '(a \ b \ (c \ d \ e)))))) \rightarrow c$

$(\text{cons } 1 \ (\text{cons } 'a \ 'b)) \rightarrow (1 \ a \ . \ b)$

$(\text{cdr } (\text{cons } 'a \ '(b \ c))) \rightarrow (b \ c)$

Problem 2

$(1 \ (2 \ . \ 3) \ 4) \rightarrow (\text{cons } 1 \ (\text{list } (\text{cons } 2 \ 3) \ 4))$

$(1 \ 2) \rightarrow (\text{list } 1 \ 2)$

$(1 \ . \ 2) \rightarrow (\text{cons } 1 \ 2)$

$(1 \ 2 \ (3 \ 4) \ (5 \ 6)) \rightarrow (\text{list } 1 \ 2 \ (\text{list } 3 \ 4) \ (\text{list } 5 \ 6))$

$((()) \ 1 \ 2) \rightarrow (\text{list } '() \ 1 \ 2)$

$(1 \ 2 \ . \ 3) \rightarrow (\text{cons } 1 \ (\text{cons } 2 \ 3))$

$(a \ b \ (c \ d) \ (e \ . \ f)) \rightarrow (\text{cons } 'a \ (\text{list } 'b \ (\text{list } 'c \ 'd) \ (\text{cons } 'e \ 'f)))$

$(a \ b \ . \ c) \rightarrow (\text{cons } 'a \ (\text{cons } 'b \ 'c))$

Problem 3

PART A

```
(define (even-numbers lst)
  (if (null? lst)
      '()
      (if (even? (car lst))
          (cons (car lst) (even-numbers (cdr lst)))
          (even-numbers (cdr lst)))
      )))
```

PART B

```
(define (substitute replacer word sentence)
  (if (null? sentence)
      '()
      (if (eq? (car sentence) word)
          (cons replacer (substitute replacer word (cdr sentence)))
          (cons (car sentence) (substitute replacer word (cdr sentence))))
      ))
)
```

Problem 4

```
(define incrementby (lambda (n) (lambda (x) (+ x n)))) → Undefined
(incrementby 2) → #<procedure> ( Which is “(lambda (x) (+ x 2))” )
((lambda (n) (lambda (x) (+ x n))) 2) → #<procedure> ( Which is “(lambda (x) (+ x 2))” )
(lambda (x) (+ x 2)) → → #<procedure>
(define f1 (incrementby 6)) → Undefined
(f1 4) → 10
(define f2 (lambda (x) (incrementby 6))) → UD
(f2 4) → #<procedure>
```

`((f2 4) 6) → 12`

`(define (compose f g) (lambda (x) (f (g x)))) → UD`

`((compose (lambda (p) (if p "hi" "bye"))) (lambda (x) (> x 0))) -5) → "bye"`

`(define add2 (lambda (n) (+ n 2))) → UD`

`(define add4 (compose add2 add2)) → ud`

`(add4 7) → 11`

Problem 5

Part A

`(define (inc n) (+ n 1))`

`(define (double aFunc) (lambda (x) (aFunc (aFunc x))))`

`((double (double double)) inc) 5) → 21`