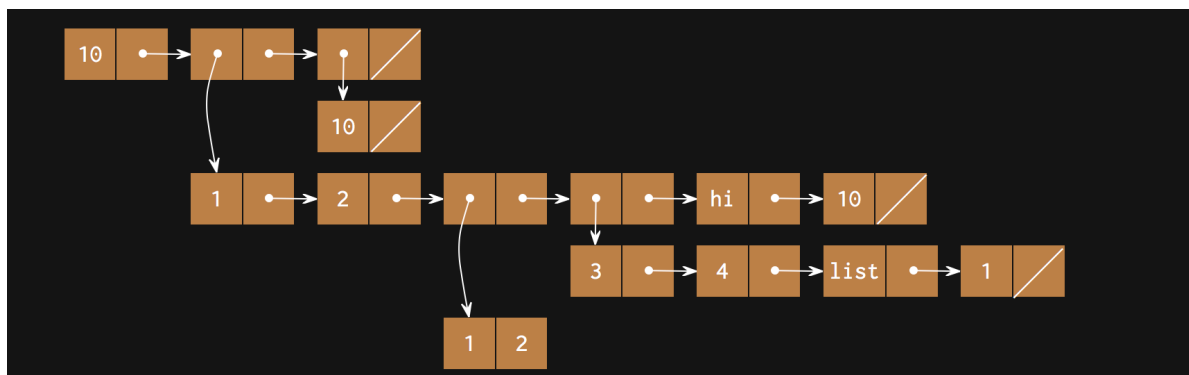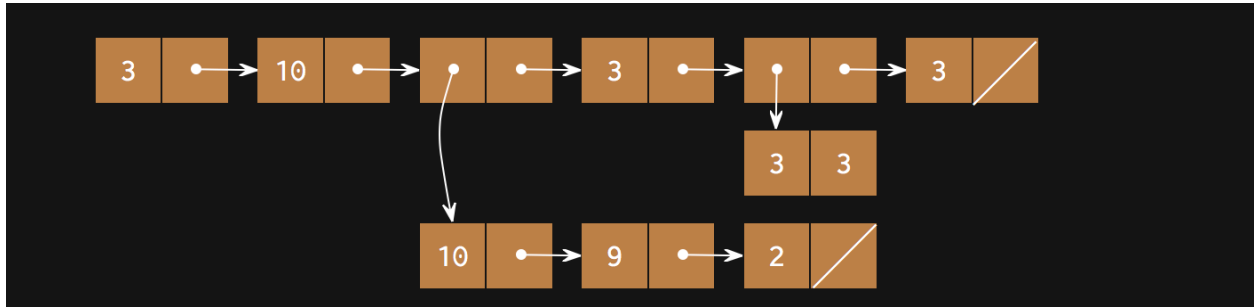# Worksheet 8 Solutions

```
scm> (+ 5 (- 2 3))
4
scm> (define x (- 3 (/ 4 2)))
x
scm> (* x x x 5 x)
5
scm> (define (y z a b c) (* (/ (+ (- c b) a) 1) z))
y
scm> (- 90 100 (+ (y 5 2 1 8) 30))
-85
scm> (+ 3 (define t 5))
Error
scm> (define x (define t 5))
x
scm> (+ x 5)
Error
scm> x
t
```

```
scm> (if (- 4 (+ 1 3)) 99 1)
99
scm> (if (and #t (or 2 3) (if (not (< 3 2)) (= 5 65) (/ 3 2))) (/ 1 0) 4)
4
```
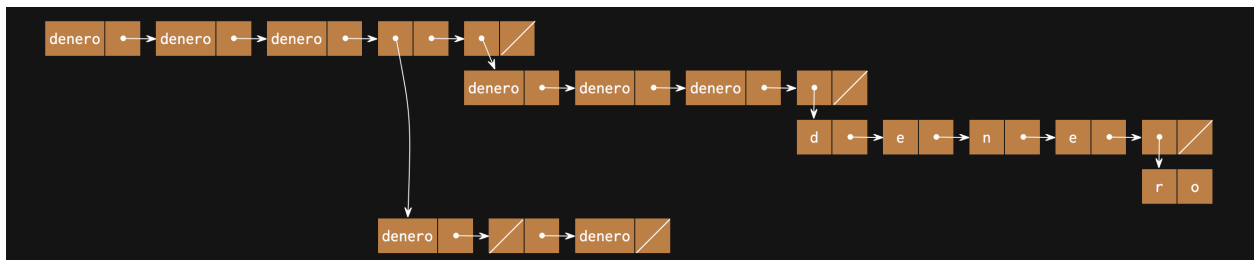
```
(define (multiply-out x)
        (lambda (y) (* x y))
)
;((multiply-out 5) 6)
;30
```

```
(10 (1 2 (1 . 2) (3 4 list 1) hi 10) (10))
```



```
(3 10 (10 9 2) 3 (3 . 3) 3)
```



```
(denero denero denero (denero () denero) (denero denero denero (d e n e (r .
o))))
```

```scheme
(define (deeply-doubled lst)
  (if (null? lst)
      nil
      (if (list? (car lst))
          (cons (deeply-doubled (car lst)) (deeply-doubled (cdr lst)))
          (cons (* 2 (car lst)) (deeply-doubled (cdr lst)))
          )
      )
  )
```