Problem 1

(1 2)
(3 1 2)
11
4

Problem 2

1
0
(x y)
((x y) z)
(x y)

Problem 3

#f #f #t
#f #f #t
#t #t #t
#f #f #t
#t #t #t

Problem 4

(define (quad-split n painter)
  (if (= n 0)
      painter
      (let ((smaller (quad-split (- n 1) painter)))
        (below (beside smaller painter)
               (below (painter smaller))))))

The last two lines could also be
  (beside (below (smaller painter))
          (below (painter smaller)))
Problem 5

(define (make-auction item-name current-price high-bidder)
  (define (bid price person)
    (if (> price current-price)
      (begin (set! current-price price)
             (set! high-bidder person)
             price)
      (list "your bid" price "is less than" current-price)))
  (define (sell)
    (list "the winner is" high-bidder "for" current-price))
  (define (dispatch m)
    (cond ((eq? m 'bid) bid)
          ((eq? m 'sell) sell)
          ((eq? m 'current-bid) current-price)
          ((eq? m 'current-high-bidder) high-bidder)
          (else (error "Unknown request" m)))
    dispatch)

((pez-sale 'bid) 10 'holly)

(pez-sale 'current-bid)

Problem 6

(define (filter test-op lst)
  (accumulate (lambda (x y) (if (test-op x) (cons x y) y))
              nil
              lst))