(define (f y)   (if (> y 0) (+ y y) "hi"))

(define a (let ([ans (f 7)])   (if (number? ans) (number->string ans) ans)))

(define (cube x)   (if (not (number? x))       {error "bad arguments")
        (* x x x)))

(define b (cube 7))

(define (f2 g)  (cons (g 7) (g #t)))

(define pair_of_pairs  (f2 (cons x (cons x x)))))

(define (pow-bad-type x) ; curried  
    (lambda (y)   (if (= y 0) 1
        (* x (pow-bad-type x (- y 1)))))); oops

(define (pow-bad-algorithm x) ; curried  
    (lambda (y)   (if (= y 0) 1
        (+ x ((pow-bad-algorithm x) (- y 1)))))); oops