Problem Set 6

Lists

1. **(cdr ‘((a) b (c d))) =>**
2. **(cadr ‘((a) b (c d))) =>**
3. **(cddr ‘((a) b (c d))) =>**
4. **(cdddr ‘((a) b (c d))) =>**
5. **(cddddr ‘((a) b (c d))) =>**
6. **(cons ‘(a b c) ‘((a) b (c d))) =>**
7. **(cons ‘d ‘(e)) =>**
8. **(cons ‘(a b) ‘(c d)) =>**
9. **(cons ‘a (cons ‘b (cons 'c '( )))) =>**
10. **(map (lambda (x)**

    (+ 1 (* x 5))) '(2 4 6)) =>

Scheme

1. **Define the function (smaller-than-all? n lst) where n is a number and lst is a list of numbers, (lst0 lst1 … lstn), and the value of the function is #t (true) if n < lsti for every lsti in lst., and if lsti is null, return false, otherwise #f (false)**

2. **Define the function (chain init fns) where init is some scheme data and fns is a list of functions. (chain 0 (list f g h)) is (f (g (h 0)))**

1. **(b (c d))**
2. **b**
3. **((c d))**
4. **( )**
5. ***error***
6. **((a b c) (a) b (c d))**
7. **(d e)**
8. **((a b) c d)**
9. **(a b c)**
10. (11 21 31)

Answer

1.

```
(define (smaller-than-all n lst)
  (if (null? lst) #f
      (and (< n (car lst))
        (smaller-than-all n (cdr lst)))))
```

2.

```
(define (chain init fns)
    (if (null? fns) init
      ((car fns)
        (chain init (cdr fns)))))
```

You may have different solutions!