Scheme Exercises

① Write a procedure that deletes all the occurre of a given atom from a given list?

```
( DEFINE    (deleteall  atm  lst)
    ( COND
        (( NULL ?  lst)  '())
        ((EQ?  atm   (CAR lst)) ( deleteall  atm  (CDR  lst)) )
        (ELSE  (CONS (CAR lst) (deleteall atm (CDR lst)) )
    )
 )
```

② Write a procedure that reverses the elements of a li

```
(DEFINE    (reverse  lis)
  (COND
    ((NULL?  lis) '())
    (ELSE  (APPEND  (reverse  (CDR lis)) (CONS ((CAR lis)'(
  ))
```

③ What does this function do?

```
(define   (fun-empty   y  lis)
   (cond
     ((null?   lis)  '() )
     (lequal?   y  (car  lis))  lis )
     (else  (fon_empty   y  (cdr  lis)))
   ))
```

4 insertion sort

```
(define (insert x lst)
  (if (null? lst)
      (list x)
      (let ((y (car lst))
            ((ys (cdr lst)))
        (if (<= x y)
            (cons x lst)
            (cons y (insert x ys))))))

(define (insertionsort lst )
  (if (null? lst)
      '()
      (insert (car lst)
          (insertionsort (cdr lst)))) )
```

experiments

```
(DEFINE x (list 1 5 3 4 5))
(display (reverse x))
(display (deleteall 5 x))
(display (insertionsort x))
```