

---

```
; Ejercicio 01 - Diego
; Con la funcion de inserte se debe poder insertar valores a la lista ya
existente y ordenarlos de manera ascendente,
; primero se agrega el nuevo valor y despues se va ordenando mediante la
recursion y detiene hasta que esten completamente ordenados
(define (insert n lista)
  (cond ((null? lista) (list n))
        ((<= n (car lista)) (cons n lista))
        (else (cons (car l)
                     (insert (cdr lista) n)))))
```

---

```
; Ejercicio 03 - Sebas
;Funcion recursiva que rota n elementos a la izquierda. Se hizo uso de
funcion llamada append que basicamente devuelve una lista con los elementos
de las dos listas dadas.
;A diferencia de list esta no genera una lista con listas dentro. Otra
funcion usada es la funcion reverse que invierte la lista entera.
(define (rotate-left n lista)
  (cond
    ((null? lista) lista)
    ((= n 0) lista)
    ((< n 0) (rotate-left (+ n 1) (append (list (car (reverse '(a b c d
e f g)))) (reverse (cdr (reverse '(a b c d e f g)))))))
    (else (rotate-left (- n 1) (append (cdr lista) (list (car
lista))))))
  )
)
```

---

```
; Ejercicio 06 - Pablo
; Si la lista está vacía la regresa vacía. Sino, por cada elemento de la
lista si es lista se vuelve a llamar la función para hacer el reverse del
elemento y sino solo invierte el orden de ese elemento con
; lo demás que queda de la lista.

(define (deep-reverse l)
  (if (null? l)
      '()
      (if (list? (car l))
          (append (dr (cdr l)) (list (dr (car l))))
          (append (dr (cdr l)) (list (car l)))))
  )
)
```