```
(define (ejemplouno)
       (define lista (cons 1 (cons 2 (cons 3 (cons 4 '()) ) ) ) )
       (display (car lista))
       (display (cdr lista))
)
>(ejemplouno)
1
(2 3 4)
---------------------------------------------------------
(define (ejemplodos)
       (define lista1 '(1 2 3 4))
       (display (cons 'hola lista1))
)
>(ejemplodos)
(hola 1 2 3 4)
---------------------------------------------------------
(define (ejemplotres)
            (define lista1 (list 1 2 3))
            (define lista2 (list 1 lista1  2  3))
            (display lista2)
)
>(ejemplotres)
(1 (1 2 3) 2 3)
---------------------------------------------------------
(define (concatenar L1 L2)
  (if (null? L1)
               L2
              (display (cons (car L1) (append (cdr L1) L2)))
     )
)
(define (ejemplocuatro)
          (define l1 (list 1 3 5 7))
          (define l2 (list 2 4 6 8))
          (concatenar l1 l2)
)
>(ejemplocuatro)
(1 3 5 7 2 4 6 8)
---------------------------------------------------------
(define (invierte L1 L2)
  (if (null? L1)
             L2
             (invierte (cdr L1) (cons (car L1) L2))
  )
)
(define (ejemplocinco)
          (define l1 (list 1 3 5 7))
          (define l2 (list 2 4 6 8))
          (invierte l1 l2)
)
>(ejemplocinco)
'(7 5 3 1 2 4 6 8)
```

```
 (define (elimina L x)
                (if (null? L)
                    '()
                    (if (= (car L) x)
                        (elimina (cdr L) x)
                        (cons (car L) (elimina (cdr L) x))
                    )
                )
)
(define (ejemploseis)
          (define l (list 1 3 5 7))
          (define x 5)
          (elimina l x)
)
>(ejemploseis)
'(1 3 7)
---------------------------------------------------------
(define (intercala A B)
                (if (and (null? A) (null? B))
                    '()
                    (if (null? A)
                        (cons (car B) (intercala A (cdr B)))
                        (cons (car A) (intercala B (cdr A)))
                    )
                )
)
(define (ejemplosiete)
          (define L1 (list 1 3 5 7))
          (define L2 (list 2 4 6 8))
          (intercala L1 L2)
)
>(ejemplosiete)
'(1 2 3 4 5 6 7 8)
---------------------------------------------------------
(define (reemplaza L x y)
                (if (null? L)
                    '()
                    (if(= (car L) x)
                       (cons y (reemplaza (cdr L) x y))
                       (cons (car L) (reemplaza (cdr L) x y))
                    )
                )
)
(define (ejemploocho)
          (define L1 (list 1 3 5 7 3 11 13 3))
          (define a 3)
          (define b 5)
          (reemplaza L1 a b)
)
>(ejemploocho)
'(1 5 5 7 5 11 13 5)
```