Problem 1: Find the last element of a list.

Example: (my-last '(a b c d))
(D)

Solution 1:

```
(define (my_last lista)
  (if (null? lista)
      ()
    (if (null? (cdr lista))
     lista
      (my_last (cdr lista))
      )
    )
  )
```

Problem 2: Find the last two elements of a list.

Example: (my-but-last '(a b c d))
(C D)

Solution 2:

```
      (define (penultimo lista)
        (let ((reverso (reverse lista)))
          (cond
           ((null? reverso) () )
           ((<= (length reverso) 2)  lista)
           (#t (list (cadr reverso) (car reverso)))
           )
          )
        )
```

Problem 3: Find the K'th element of a list.

The first element in the list is number 1.

Example: (element-at '(a b c d e) 3)
C

Solution 3:

```
(define (element-at lista n)
  (if (= n 1)
     (car lista)
   (element-at (cdr lista) (- n 1))
   )
  )
```

Problem 4: Find the number of elements of a list.

Solution 4:

```
(define (no_of_elements lista)
  (if (null? lista)
     0
   (+ 1 (no_of_elements (cdr lista)))))
```

Problem 5: Reverse a list.
Solution 5:

```
(define (inverte lista)
 (inverte-aux lista () )
  )
(define (inverte-aux lista resto)
 (if (null? lista)
     resto
(inverte-aux (cdr lista) (cons (car lista) resto) )
   )
  )
```

Problem 6: Find out whether a list is a palindrome.

A palindrome can be read forward or backward; e.g. (x a m a x).

Solution 6:

```
(defun palin (lista)
  (equal lista (reverse lista))
  )
```

Problem 7: Flatten a nested list structure.

Transform a list, possibly holding lists as elements into a `flat' list by replacing each list with its elements (recursively).

Example: (my-flatten '(a (b (c d) e))))
(A B C D E)

Hint: Use the predefined functions list and append.

Solution 7:

```
 (define (flatten orig-list)
   (if (null? orig-list)
      ()
      (let ((elem (car orig-list)) (resto-list (cdr orig-list)))
         (if (list? elem)
            (append (flatten elem) (flatten resto-list))
            (append (cons elem ()) (flatten resto-list))))))
```

Source:
http://www.ic.unicamp.br/~meidanis/courses/mc336/2006s2/funcional/L-99_Ninety-Nine_Lisp_Problems.html