Exercise 2.18.

Define a procedure `reverse` that takes a list as argument and returns a list of the same elements in reverse order:

```
(reverse (list 1 4 9 16 25))
(25 16 9 4 1)
```

Answer.

To `reverse` a list, do the following:

- If the list contains only one element, then just return the list.
- Otherwise, `reverse` all but the last element of the list, and `cons` that last element onto the result.

The second case above can be implemented with the help of two procedures: `last-element` that picks the last element in a given list and `former-sublist` which constructs a list that contains all but the last element of the original list.

```
(define (reverse items)
  (define (last-element items)
    (if (null? (cdr items))
        (car items)
        (last-element (cdr items))))
  (define (former-sublist items)
    (if (null? (cdr items))
        nil
        (cons (car items) (former-sublist (cdr items)))))
  (if (null? (cdr items))
      items
      (cons (last-element items) (reverse (former-sublist items)))))
```