

As *answer* grows, each use of **cons** prefixes *answer* with the next element of *things*. Hence the list is reversed.

Each use of **cons** with a list ℓ and an element e as its second argument builds a list such that the **cdr** of ℓ is not e , but is **nil**. Thus we attain a list containing the pair ℓ not containing e , and e itself.

By using **append**, we return the list containing *answer* and the list containing only the element we want. Our program is simply

```
(define (square-list items)
  (define (iter things answer)
    (if (null? things)
        answer
        (iter (cdr things)
              (append answer
                      (list (square (car things)))))))
  (iter items nil))
```