Exercise 2.34

Eduardo Acuña Yeomans

Exercise from the *Sequences as Conventional Interfaces* section of Structure and Interpretation of Computer Programs.

Evaluating a polynomial in x at a given value of x can be formulated as an accumulation. We evaluate the polynomial

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

using a well-known algorithm called Horner's rule, which structures the computation as

$$(\ldots(a_n x + a_{n-1})x + \ldots + a_1)x + a_0$$

In other words, we start with a_n, multiply by x, add a_{n-1}, multiply by x and so on until we reach a_0. Fill in the following template to produce a procedure that evaluates a polynomial using Horner's rule. Assume that the coefficients of the polynomial are arranged in a sequence, from a_0 through a_n.

```
(define (horner-eval x coefficient-sequence)
  (accumulate (lambda (this-coeff higher-terms) <??>)
              0
              coefficient-sequence))
```

For example, to compute $1 + 3x + 5x^3 + x^5$ at $x = 2$ you would evaluate `]=>(horner-eval 2 (list 1 3 0 5 0 1))`

```
(define (accumulate op initial sequence)
  (if (null? sequence)
      initial
      (op (car sequence)
          (accumulate op initial (cdr sequence)))))
```

horner-eval : *number* $\times$ *list* $\rightarrow$ *number*
Return the evaluation of the polynomial with the coefficients given by the `coefficient-sequence` at a given number `x` using Horner's rule.

```
(define (horner-eval x coefficient-sequence)
  (accumulate (lambda (this-coeff higher-terms)
                (+ this-coeff (* x higher-terms)))
              0
              coefficient-sequence))
```