Pre-Lab Exercise 2.1 The approximation is accurate to 4 decimal places for $k = 11$.

```
(define (cont-frac-r n d k)
  (define (cf i)
    (if (> i k)
        0
        (/ (n i)
           (+ (d i)
              (cf (+ i 1))))))
  (cf 1))

(define (cont-frac-i n d k)
  (define (cf-iter k sum)
    (if (= k 0)
        sum
        (cf-iter (- k 1)
                 (/ (n k)
                    (+ (d k)
                       sum)))))
  (cf-iter k 0))
```

Lab Exercise 2.2 The approximation is accurate to 2 decimal places for k approximately 120.

Pre-Lab Exercise 2.3

```
(define (atan-cf k x)
  (cont-frac (lambda (i)
               (if (= i 1)
                   x
                   (square (* (- i 1) x))))
             (lambda (i)
               (- (* 2 i) 1))
             k))
```

Lab Exercise 2.4 Given `(define pi (estimate-pi 10000))` where 10000 is arbitrarily chosen.

procedure	k	x	value
`atan-cf`	10	1	.785398135111635
`atan-cf`	30	1	.7853981633974483
`atan-cf`	100	1	.7853981633974483
`atan`		1	.7853981633974483
`atan-cf`	10	π	1.258297497839761
`atan-cf`	30	π	1.2626364386129623
`atan-cf`	100	π	1.2626364544599773
`atan`		π	1.262636454459977

Pre-Lab Exercise 2.5

```
(define (nested-acc op r term k)
  (define (na i)
    (if (> i k)
        r
        ((op i) (term i) (na (+ i 1)))))
  (na 1))

(define (f k x)
  (nested-acc (lambda (i)
                  (lambda (a b)
                    (sqrt (+ a b))))
                0
                (lambda (i) x)
                k))
```

Lab Exercise 2.6 The approximation is accurate to 4 decimal places for $k = 10$.

Pre-Lab Exercise 2.8

```
(define (repeated-build k n d b)
  (if (= k 0)
      0
      ((repeated (lambda (x) (build n d x)) (- k 1)) (build n d b))))
```

Lab Exercise 2.9

```
(repeated-build 11 1 1 0) ⟹ 89/144
```

Lab Exercise 2.10

```
(define (r k)
  (lambda (x)
    (repeated-build k 1 1 x)))
```