
88
60
56
01
24
80
17
51
66
41
82

Racket

Brand X

```
(define (average loi)
  (/ (apply + loi)
     (length loi)))

(average
 '(2013 ; red sox
   2011 ; bruins
   2010 ; celtics
   2004 ; patriots
        ))
```

```
public static int average (int[] loi){
  int sum = 0;
  for (int i = 0; i < loi.length; i++)
    sum += loi[i];
  return sum / loi.length;
}

public static void main (String[] args){
  int[] years
    = {2013, 2011, 2010, 2004};
  System.out.println(average(years));
}
```

Racket

Brand X

```
(define (average loi)
  (/ (apply + loi)
     (length loi)))

(average
 '(2013 ; red sox
   2011 ; bruins
   2010 ; celtics
   2004 ; patriots
        ))
```

```
4019/2
; 2009.5
```

```
public static int average (int[] loi){
  int sum = 0;
  for (int i = 0; i < loi.length; i++)
    sum += loi[i];
  return sum / loi.length;
}

public static void main (String[] args){
  int[] years
    = {2013, 2011, 2010, 2004};
  System.out.println(average(years));
}
```

Racket

Brand X

```
(define (average loi)
  (/ (apply + loi)
     (length loi)))

(average
 '(2013 ; red sox
   2011 ; bruins
   2010 ; celtics
   2004 ; patriots
        ))
```

```
public static int average (int[] loi){
  int sum = 0;
  for (int i = 0; i < loi.length; i++)
    sum += loi[i];
  return sum / loi.length;
}

public static void main (String[] args){
  int[] years
    = {2013, 2011, 2010, 2004};
  System.out.println(average(years));
}
```

```
4019/2
; 2009.5
```

```
2009
```

Racket

Brand Y

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
```

```
q :: (Floating a)
  => a -> a -> a -> a
q a b c
  = (-b + sqrt (b**2 - 4*a*c))
      / (2*a)

q (fromIntegral 1)
  (fromIntegral 3)
  (fromIntegral 3)
```

Racket

Brand Y

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
```

```
q :: (Floating a)
  => a -> a -> a -> a
q a b c
  = (-b + sqrt (b**2 - 4*a*c))
      / (2*a)

q (fromIntegral 1)
  (fromIntegral 3)
  (fromIntegral 3)
```

-1.5+0.866i

Racket

Brand Y

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
```

```
q :: (Floating a)
  => a -> a -> a -> a
q a b c
  = (-b + sqrt (b**2 - 4*a*c))
      / (2*a)

q (fromIntegral 1)
  (fromIntegral 3)
  (fromIntegral 3)
```

-1.5+0.866i

NaN

Racket

Brand Y

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
```

```
q :: (Floating a)
  => a -> a -> a -> a
q a b c
  = (-b + sqrt (b**2 - 4*a*c))
      / (2*a)

q (fromIntegral 1)
  (fromIntegral 3)
  (fromIntegral 3)
```

-1.5+0.866i

NaN

Racket

Brand Y

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
```

```
q :: (Floating a)
  => a -> a -> a -> a
q a b c
  = (-b + sqrt (b**2 - 4*a*c))
      / (2*a)

q (fromIntegral 1)
  (fromIntegral 3)
  (fromIntegral 3)
```

-1.5+0.866i

NaN

Racket

Brand Y

```
(define (q a b c
  (/ (+ (- b)
        (sqrt (-
              (*
     (* 2 a)))

(q 1 3 3)
```

```
)
 -> a

b**2 - 4*a*c))

L)
3)
3)
```

Types

-1.5+0.866i

NaN

```
(define (q a b c)
  (/ (+ (- b)
        (sqrt (- (sqr b)
                 (* 4 a c))))
     (* 2 a)))

(q 1 3 3)
-1.5+0.866i
```

sqrt : ???

```
(sqrt -1) ; => 0+1i
```

```
sqrt : (Complex → Complex)
```

```
(define (pythagorean a b)
  (sqrt (+ (sqr a) (sqr b))))
```

```
sqrt : (Complex → Complex)
```

```
(define (pythagorean a b)
  (sqrt (+ (sqr a) (sqr b))))
```

```
sqrt : (case→
        (Nonnegative-Real
          → Nonnegative-Real)
        (Complex → Complex))
```

sqrt :

```
(case-> (Zero -> Zero)
        (One -> One)
        (Flonum-Positive-Zero -> Flonum-Positive-Zero)
        (Flonum-Negative-Zero -> Flonum-Negative-Zero)
        (Flonum-Zero -> Flonum-Zero)
        (Positive-Flonum -> Positive-Flonum)
        (Nonnegative-Flonum -> Nonnegative-Flonum)
        (Single-Flonum-Positive-Zero -> Single-Flonum-Positive-Zero)
        (Single-Flonum-Negative-Zero -> Single-Flonum-Negative-Zero)
        (Single-Flonum-Zero -> Single-Flonum-Zero)
        (Positive-Single-Flonum -> Positive-Single-Flonum)
        (Nonnegative-Single-Flonum -> Nonnegative-Single-Flonum)
        (Inexact-Real-Positive-Zero -> Inexact-Real-Positive-Zero)
        (Inexact-Real-Negative-Zero -> Inexact-Real-Negative-Zero)
        (Inexact-Real-Zero -> Inexact-Real-Zero)
        (Positive-Inexact-Real -> Positive-Inexact-Real)
        (Nonnegative-Inexact-Real -> Nonnegative-Inexact-Real)
        (Real-Zero -> Real-Zero)
        (Positive-Real -> Positive-Real)
        (Nonnegative-Real -> Nonnegative-Real)
        (Float-Complex -> Float-Complex)
        (Single-Flonum-Complex -> Single-Flonum-Complex)
        (Inexact-Complex -> Inexact-Complex)
        (Number -> Number))
```

```
(class (: Zero)
       (Zero Zero -> Zero)
       (One Zero -> One)
       (Zero One -> One)
       (Positive.Byte Zero -> Positive.Byte)
       (Zero Positive.Byte -> Positive.Byte)
       (Byte Zero -> Byte)
       (Zero Byte -> Byte)
       (Positive.Index Zero -> Positive.Index)
       (Zero Positive.Index -> Positive.Index)
       (Index Zero -> Index)
       (Zero Index -> Index)
       (Positive.Nonne Zero -> Positive.Fixnum)
       (Zero Positive.Nonne -> Positive.Fixnum)
       (Nonnegative.Fixnum Zero -> Nonnegative.Fixnum)
       (Zero Nonnegative.Fixnum -> Nonnegative.Fixnum)
       (Negative.Nonne Zero -> Negative.Fixnum)
       (Zero Negative.Nonne -> Negative.Fixnum)
       (Nonpositive.Fixnum Zero -> Nonpositive.Fixnum)
       (Zero Nonpositive.Fixnum -> Nonpositive.Fixnum)
       (Fixnum Zero -> Fixnum)
       (Zero Fixnum -> Fixnum)
       (Positive.Byte Positive.Byte -> Positive.Index)
       (Byte Byte -> Index)
       (Positive.Byte Positive.Byte Positive.Byte -> Positive.Index)
       (Byte Byte Byte -> Index)
       (Positive.Index Index -> Positive.Fixnum)
       (Index Positive.Index -> Positive.Fixnum)
       (Positive.Index.Index Index -> Positive.Fixnum)
       (Index Positive.Index.Index -> Positive.Fixnum)
       (Index Index Positive.Index -> Positive.Fixnum)
       (Index Index -> Nonnegative.Fixnum)
       (Index Index Index -> Nonnegative.Fixnum)
       (Negative.Nonne One -> Nonpositive.Fixnum)
       (One Negative.Fixnum -> Nonpositive.Fixnum)
       (Nonpositive.Fixnum Nonnegative.Fixnum -> Fixnum)
       (Nonnegative.Fixnum Nonpositive.Fixnum -> Fixnum)
       (Positive.Integer Nonnegative.Integer Nonnegative.Integer * -> Positive.Integer)
       (Nonnegative.Integer Positive.Integer Nonnegative.Integer * -> Positive.Integer)
       (Nonnegative.Integer Nonnegative.Integer Positive.Integer Nonnegative.Integer * -> Positive.Integer)
       (Nonnegative.Integer Nonpositive.Integer Nonpositive.Integer * -> Negative.Integer)
       (Nonpositive.Integer Negative.Integer Nonpositive.Integer * -> Negative.Integer)
       (Nonpositive.Integer Nonpositive.Integer Negative.Integer Nonpositive.Integer * -> Negative.Integer)
       (Zero -> Zero)
       (One -> One)
       (Positive.Byte -> Positive.Byte)
       (Byte -> Byte)
       (Positive.Index -> Positive.Index)
       (Index -> Index)
       (Positive.Nonne -> Positive.Fixnum)
       (Nonnegative.Fixnum -> Nonnegative.Nonne)
       (Negative.Nonne -> Negative.Fixnum)
       (Nonpositive.Fixnum -> Nonpositive.Nonne)
       (Fixnum -> Fixnum)
       (Positive.Integer -> Positive.Integer)
       (Nonnegative.Integer -> Nonnegative.Integer)
       (Negative.Integer -> Negative.Integer)
       (Nonpositive.Integer -> Nonpositive.Integer)
       (Integer -> Integer)
       (Nonnegative.Integer * -> Nonnegative.Integer)
       (Nonpositive.Integer * -> Nonpositive.Integer)
       (Integer * -> Integer)
       (Positive.Exact-Rational Nonnegative.Exact-Rational Nonnegative.Exact-Rational * -> Positive.Exact-Rational)
       (Nonnegative.Exact-Rational Positive.Exact-Rational Nonnegative.Exact-Rational * -> Positive.Exact-Rational)
       (Nonnegative.Exact-Rational Nonnegative.Exact-Rational Positive.Exact-Rational Nonnegative.Exact-Rational * -> Positive.Exact-Rational)
       (Negative.Exact-Rational Nonpositive.Exact-Rational Nonpositive.Exact-Rational * -> Negative.Exact-Rational)
       (Nonpositive.Exact-Rational Negative.Exact-Rational Nonpositive.Exact-Rational * -> Negative.Exact-Rational)
       (Nonpositive.Exact-Rational Nonpositive.Exact-Rational Negative.Exact-Rational Nonpositive.Exact-Rational * -> Negative.Exact-Rational)
       (Positive.Exact-Rational -> Positive.Exact-Rational)
       (Nonnegative.Exact-Rational -> Nonnegative.Exact-Rational)
       (Negative.Exact-Rational -> Negative.Exact-Rational)
       (Nonpositive.Exact-Rational -> Nonpositive.Exact-Rational)
       (Exact-Rational -> Exact-Rational)
       (Nonnegative.Exact-Rational * -> Nonnegative.Exact-Rational)
       (Nonpositive.Exact-Rational * -> Nonpositive.Exact-Rational)
       (Exact-Rational * -> Exact-Rational)
       (Positive.Nonne Nonnegative.Real Nonnegative.Real * -> Positive.Flonum)
       (Nonnegative.Real Positive.Flonum Nonnegative.Real * -> Positive.Flonum)
       (Nonnegative.Real Nonnegative.Real Positive.Real Nonnegative.Real * -> Positive.Nonne
       (Positive.Real Nonnegative.Flonum Nonnegative.Nonne * -> Positive.Flonum
       (Nonnegative.Flonum Positive.Real Nonnegative.Nonne * -> Positive.Flonum
       (Nonnegative.Flonum Nonnegative.Flonum Positive.Real Nonnegative.Flonum * -> Positive.Flonum)
       (Negative.Nonne Nonpositive.Real Nonpositive.Real * -> Negative.Flonum)
       (Nonpositive.Real Negative.Flonum Nonpositive.Real * -> Negative.Flonum)
       (Nonpositive.Real Nonpositive.Real Negative.Real Nonpositive.Real * -> Negative.Nonne
       (Negative.Real Nonpositive.Flonum Nonpositive.Nonne * -> Negative.Flonum
       (Nonpositive.Flonum Negative.Real Nonpositive.Nonne * -> Negative.Flonum
       (Nonpositive.Flonum Nonpositive.Flonum Negative.Real Nonpositive.Flonum * -> Negative.Flonum)
       (Nonpositive.Flonum Nonnegative.Real Nonnegative.Real * -> Nonpositive.Flonum)
       (Nonpositive.Real Nonnegative.Flonum Nonnegative.Real * -> Nonpositive.Flonum)
       (Nonpositive.Real Nonnegative.Real Nonpositive.Real * -> Nonpositive.Flonum)
       (Nonpositive.Real Nonnegative.Real Nonpositive.Flonum Nonpositive.Real * -> Nonpositive.Flonum)
       (Flonum Real Real * -> Flonum)
       (Real Flonum Real * -> Flonum)
       (Real Real Flonum Real * -> Flonum)
       (Flonum.Positive.Zero -> Nonnegative.Zero)
       (Flonum.Negative.Zero -> Nonnegative.Zero)
       (Flonum.Zero -> Nonnegative.Zero)
       (Flonum.Nan -> Nonnegative.Nonne)
       (Positive.Nonne -> Positive.Flonum)
       (Nonnegative.Flonum -> Nonnegative.Nonne)
       (Negative.Nonne -> Negative.Flonum)
       (Nonpositive.Flonum -> Nonpositive.Nonne)
       (Flonum -> Flonum)
       (Flonum Flonum * -> Nonne)
       (Positive.Single.Flonum (0 Nonnegative.Single.Nonne Nonnegative.Exact-Rational) (0 Nonnegative.Single.Nonne Nonnegative.Exact-Rational) * -> Positive.Single.Flonum)
       (0 Nonnegative.Single.Flonum Nonnegative.Exact-Rational) Positive.Single.Flonum (0 Nonnegative.Single.Nonne Nonnegative.Exact-Rational) * -> Positive.Single.Flonum)
       (0 Nonnegative.Single.Flonum Nonnegative.Exact-Rational) (0 Nonnegative.Single.Flonum Nonnegative.Single.Nonne Nonnegative.Exact-Rational) Positive.Single.Flonum (0 Nonnegative.Single.Nonne Nonnegative.Exact-Rational) * -> Positive.Single.Nonne)
       (0 Positive.Single.Nonne Positive.Exact-Rational) Nonnegative.Single.Flonum Nonnegative.Single.Flonum * -> Positive.Single.Flonum)
       (0 Positive.Single.Flonum Nonnegative.Single.Nonne (0 Positive.Single.Nonne Positive.Exact-Rational) Nonnegative.Single.Nonne * -> Positive.Single.Flonum)
       (Negative.Single.Flonum (0 Nonpositive.Single.Nonne Nonpositive.Exact-Rational) (0 Nonpositive.Single.Nonne Nonpositive.Exact-Rational) * -> Negative.Single.Flonum)
       (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) Negative.Single.Flonum (0 Nonpositive.Single.Nonne Nonpositive.Exact-Rational) * -> Negative.Single.Flonum)
       (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) (0 Nonpositive.Single.Flonum Nonpositive.Single.Nonne Nonpositive.Exact-Rational) Negative.Single.Flonum (0 Nonpositive.Single.Nonne Nonpositive.Exact-Rational) * -> Negative.Single.Nonne)
       (0 Negative.Single.Nonne Negative.Exact-Rational) Nonpositive.Single.Flonum Nonpositive.Single.Flonum * -> Negative.Single.Flonum)
       (Nonpositive.Single.Nonne (0 Negative.Single.Nonne Negative.Exact-Rational) Nonpositive.Single.Flonum * -> Negative.Single.Flonum)
       (Nonpositive.Single.Flonum (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) * -> Nonpositive.Single.Nonne)
       (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) Nonpositive.Single.Flonum * -> Nonpositive.Single.Nonne)
       (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) Nonpositive.Single.Flonum (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) * -> Nonpositive.Single.Nonne)
       (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) Nonpositive.Single.Nonne (0 Nonpositive.Single.Flonum Nonpositive.Exact-Rational) * -> Nonpositive.Single.Nonne)
       (0 Single.Flonum Exact-Rational) Single.Nonne (0 Single.Flonum Exact-Rational) * -> Single.Flonum)
       (0 Single.Flonum Exact-Rational) (0 Single.Flonum Exact-Rational) Single.Flonum (0 Single.Flonum Exact-Rational) * -> Single.Flonum)
       (Single.Flonum.Positive.Zero -> Single.Flonum.Positive.Zero)
       (Single.Flonum.Negative.Zero -> Single.Flonum.Negative.Zero)
       (Single.Flonum.Zero -> Single.Flonum.Zero)
       (Single.Flonum.Nan -> Single.Flonum.Nan)
       (Positive.Single.Flonum -> Positive.Single.Flonum)
       (Nonnegative.Single.Nonne -> Nonnegative.Single.Flonum)
       (Negative.Single.Flonum -> Negative.Single.Flonum)
       (Nonpositive.Single.Nonne -> Nonpositive.Single.Flonum)
       (Single.Flonum -> Single.Nonne)
       (Single.Flonum Single.Flonum * -> Single.Nonne)
       (Positive.Exact-Real Nonnegative.Real Nonnegative.Real * -> Positive.Exact-Real)
       (Nonnegative.Real Positive.Exact-Real Nonnegative.Real * -> Positive.Exact-Real)
       (Nonnegative.Real Nonnegative.Real Positive.Exact-Real Nonnegative.Real * -> Positive.Exact-Real)
       (Positive.Real Nonnegative.Exact-Real Nonnegative.Exact-Real * -> Positive.Exact-Real)
       (Nonnegative.Exact-Real Positive.Real Nonnegative.Exact-Real * -> Positive.Exact-Real)
       (Nonnegative.Exact-Real Nonnegative.Real Positive.Exact-Real Nonnegative.Real * -> Positive.Exact-Real)
       (Negative.Exact-Real Nonpositive.Real Nonpositive.Real * -> Negative.Exact-Real)
       (Nonpositive.Real Negative.Exact-Real Nonpositive.Real * -> Negative.Exact-Real)
       (Nonpositive.Real Nonpositive.Real Negative.Exact-Real Nonpositive.Real * -> Negative.Exact-Real)
       (Nonpositive.Real Nonpositive.Exact-Real Nonpositive.Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Real Nonpositive.Real Nonpositive.Exact-Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Exact-Real Nonpositive.Real Nonpositive.Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Exact-Real Nonpositive.Exact-Real Nonpositive.Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Real Nonpositive.Exact-Real Nonpositive.Exact-Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Exact-Real Nonpositive.Real Nonpositive.Exact-Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Exact-Real Nonpositive.Exact-Real Nonpositive.Real * -> Nonpositive.Exact-Real)
       (Nonpositive.Real Nonpositive.Real Nonpositive.Exact-Real Nonpositive.Exact-Real * -> Nonpositive.Exact-Real)
       (Exact-Real Real Real * -> Exact-Real)
       (Real Exact-Real Real * -> Exact-Real)
       (Real Real Exact-Real Real * -> Exact-Real)
       (Exact-Real.Positive.Zero -> Exact-Real.Positive.Zero)
       (Exact-Real.Negative.Zero -> Exact-Real.Negative.Zero)
       (Exact-Real.Zero -> Exact-Real.Zero)
       (Exact-Real.Nan -> Exact-Real.Nan)
       (Positive.Exact-Real -> Positive.Exact-Real)
       (Nonnegative.Exact-Real -> Nonnegative.Exact-Real)
       (Negative.Exact-Real -> Negative.Exact-Real)
       (Nonpositive.Exact-Real -> Nonpositive.Exact-Real)
       (Exact-Real -> Exact-Real)
       (Positive.Real Nonnegative.Real Nonnegative.Real * -> Positive.Real)
       (Nonnegative.Real Positive.Real Nonnegative.Real * -> Positive.Real)
       (Nonnegative.Real Nonnegative.Real Positive.Real Nonnegative.Real * -> Positive.Real)
       (Nonnegative.Real Nonnegative.Real Nonnegative.Real Positive.Real Nonnegative.Real * -> Positive.Real)
       (Negative.Real Nonpositive.Real Nonpositive.Real * -> Negative.Real)
       (Nonpositive.Real Negative.Real Nonpositive.Real * -> Negative.Real)
       (Nonpositive.Real Nonpositive.Real Negative.Real Nonpositive.Real * -> Negative.Real)
       (Real.Zero -> Real.Zero)
       (Positive.Real -> Positive.Real)
       (Nonnegative.Real -> Nonnegative.Real)
       (Negative.Real -> Negative.Real)
       (Nonpositive.Real -> Nonpositive.Real)
       (Real * -> Real)
       (Exact.Number * -> Exact.Number)
       (Float.Complex Number Number * -> Float.Complex)
       (Number Float.Complex Number * -> Float.Complex)
       (Number Number Float.Complex Number * -> Float.Complex)
```

Things **Not** To Do With Big Types

- Show them in error messages
- Show them at the REPL
- Show them to users at all
- Not show them in talks

Things To Do With Big Types (at the REPL)

- `:type`
- `:print-type`
- `:query-type/args`
- `:query-type/result`