

Assignment 2- Functional Programming

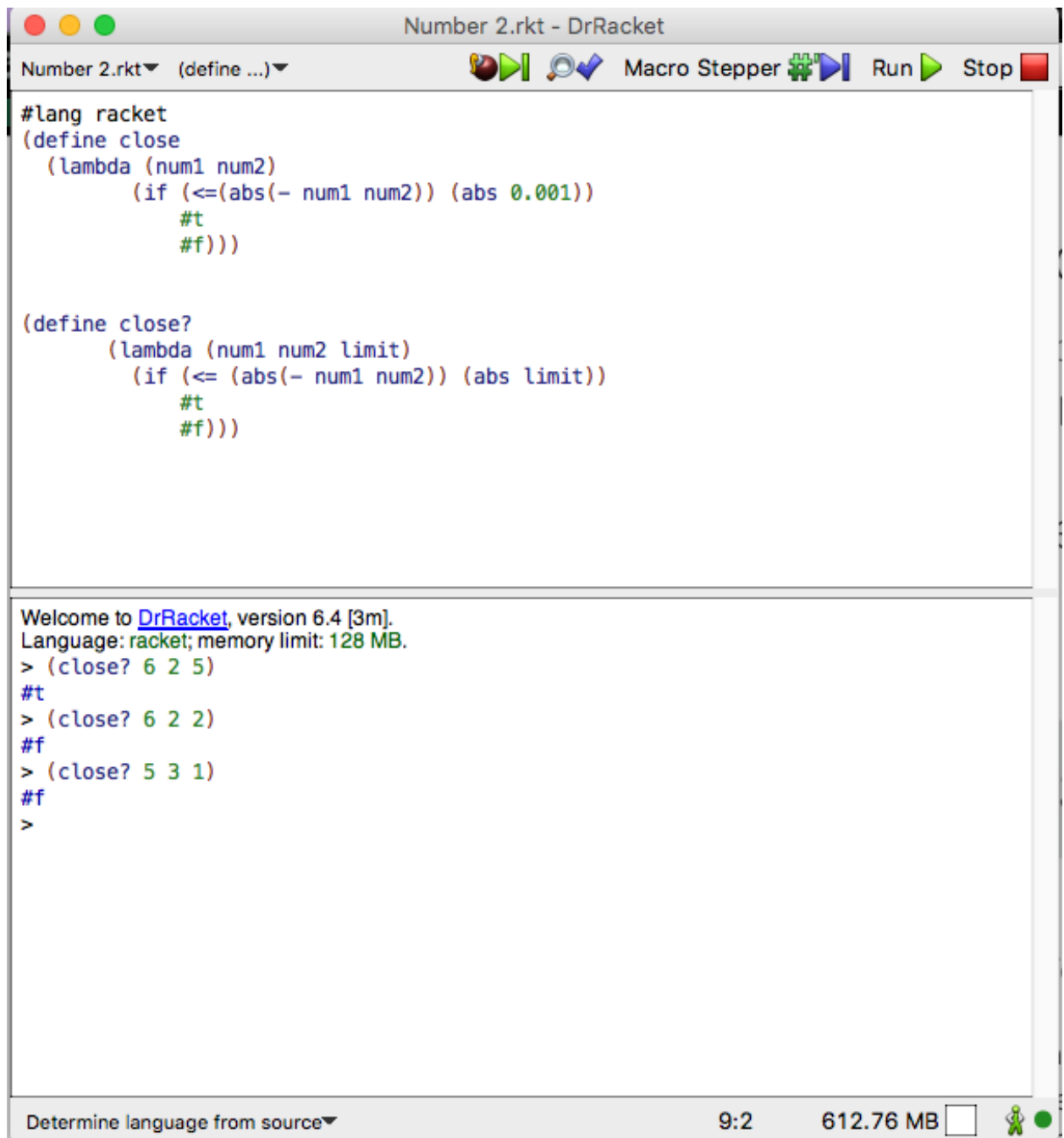
1)

```
#lang racket
(define sphereVolume
  (lambda (r)
    (* 4 3.14 (/ (* r r r) 3))))

(define shellVolume
  (lambda (outterR innerR)
    (- (sphereVolume outterR) (sphereVolume innerR))))
```

```
Welcome to DrRacket, version 6.4 [3m].
Language: racket; memory limit: 128 MB.
> (shellVolume 5 2)
489.84000000000003
> (shellVolume 3 4)
-154.90666666666664
> (sphereVolume 2)
33.49333333333333
> (sphereVolume 1)
4.1866666666666665
> |
```

2)



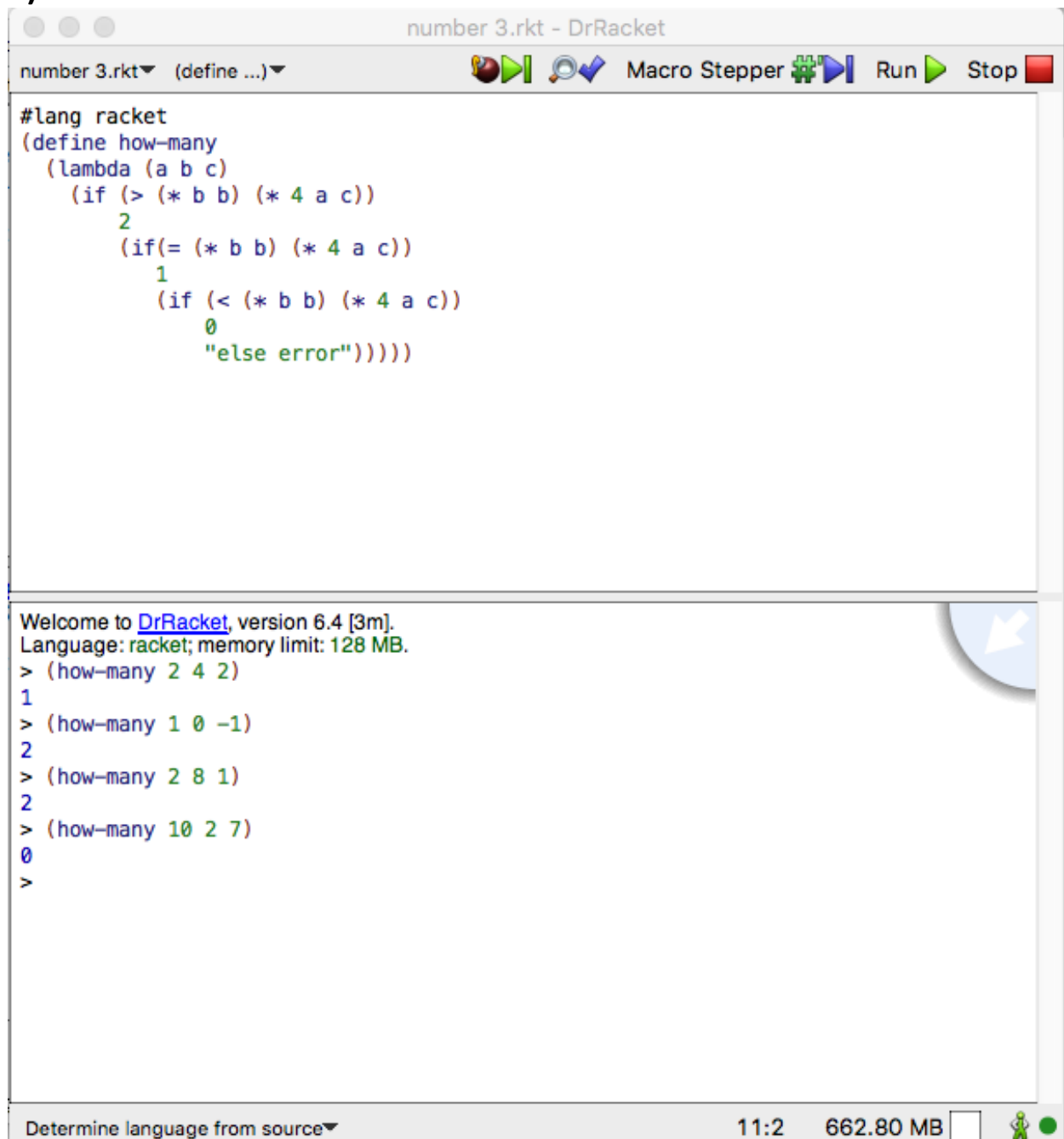
```
Number 2.rkt - DrRacket
Number 2.rkt (define ...)
#lang racket
(define close
  (lambda (num1 num2)
    (if (<= (abs(- num1 num2)) (abs 0.001))
        #t
        #f)))

(define close?
  (lambda (num1 num2 limit)
    (if (<= (abs(- num1 num2)) (abs limit))
        #t
        #f)))

Welcome to DrRacket, version 6.4 [3m].
Language: racket; memory limit: 128 MB.
> (close? 6 2 5)
#t
> (close? 6 2 2)
#f
> (close? 5 3 1)
#f
>

Determine language from source 9:2 612.76 MB
```

3)



The screenshot shows the DrRacket IDE window titled "number 3.rkt - DrRacket". The interface includes a menu bar with "number 3.rkt" and "(define ...)", a toolbar with icons for running, stepping, and stopping, and a status bar at the bottom.

The main editing area contains the following Racket code:

```
#lang racket
(define how-many
  (lambda (a b c)
    (if (> (* b b) (* 4 a c))
      2
      (if(= (* b b) (* 4 a c))
        1
        (if (< (* b b) (* 4 a c))
          0
          "else error")))))
```

The bottom panel shows the welcome message and the execution results of several test cases:

```
Welcome to DrRacket, version 6.4 [3m].
Language: racket; memory limit: 128 MB.
> (how-many 2 4 2)
1
> (how-many 1 0 -1)
2
> (how-many 2 8 1)
2
> (how-many 10 2 7)
0
>
```

The status bar at the bottom indicates "Determine language from source", the time "11:2", and the memory usage "662.80 MB".

4)

The screenshot shows the DrRacket IDE window titled "Number 4.rkt - DrRacket". The interface includes a menu bar with "Number 4.rkt" and "(define ...)", a toolbar with icons for running, stepping, and stopping, and a tab bar with "1: Number 4.rkt" and "2: Untitled 7".

The main editor area contains the following Racket code:

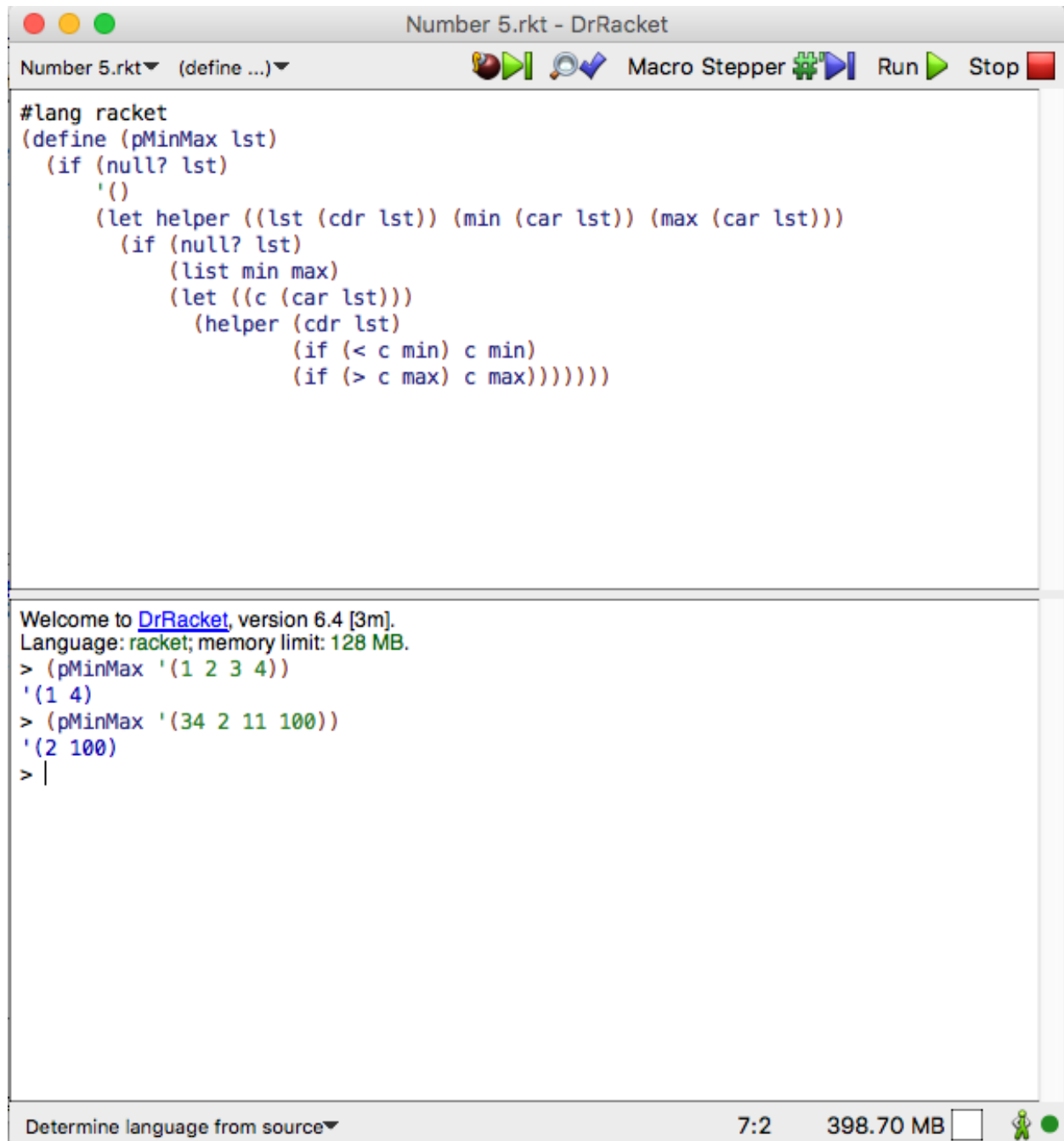
```
#lang racket
(define (filter-out list symbol)
  (cond ((null? list) '())
        ((eq? (car list) symbol)
         (filter-out (cdr list) symbol))
        (else (cons (car list)
                     (filter-out (cdr list) symbol)))))
```

The bottom panel shows the REPL output:

```
Welcome to DrRacket, version 6.4 [3m].
Language: racket; memory limit: 128 MB.
> (filter-out '(no no a thousand times no) 'no)
'(a thousand times)
> (filter-out '(hi hi how are you hi) 'hi)
'(how are you)
> |
```

The status bar at the bottom indicates "Determine language from source", "7:2", and "662.84 MB".

5)



The screenshot shows the DrRacket IDE window titled "Number 5.rkt - DrRacket". The interface includes a menu bar with "Number 5.rkt" and "(define ...)". The toolbar contains icons for running, stepping, and stopping, along with labels "Macro Stepper", "Run", and "Stop".

The main editing area contains the following Racket code:

```
#lang racket
(define (pMinMax lst)
  (if (null? lst)
      '()
      (let helper ((lst (cdr lst)) (min (car lst)) (max (car lst)))
        (if (null? lst)
            (list min max)
            (let ((c (car lst)))
              (helper (cdr lst)
                      (if (< c min) c min)
                      (if (> c max) c max))))))))
```

The bottom console area displays the following output:

```
Welcome to DrRacket, version 6.4 [3m].
Language: racket; memory limit: 128 MB.
> (pMinMax '(1 2 3 4))
'(1 4)
> (pMinMax '(34 2 11 100))
'(2 100)
> |
```

The status bar at the bottom shows "Determine language from source", the zoom level "7:2", the memory usage "398.70 MB", and a small icon of a person.

6)

