Problem 1

```
;-----------------------------Problem 1 start--------------------------------
(define (sphere-volume radius)
  (/ (* 4 (* 3.14 (cube radius))) 3))

(define (shell-volume inner-radius outer-radius)
  (- (sphere-volume outer-radius) (sphere-volume inner-radius)))

(define (cube n)
  (* n (* n n)))

;-----------------------------Problem 1 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student; memory limit: 128 MB.
> (shell-volume 5 8)
1620.24
> (shell-volume 3 5)
410.293
>
```

Problem 2

```
;-----------------------------Problem 2 start--------------------------------
(define (close? number-1 number-2 limit)
  (<= (abs (- number-1 number-2)) limit))
;-----------------------------Problem 2 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student; memory limit: 128 MB.
> (close? 1.785 1.7851 0.001)
#true
> (close? 1.785 1.781 0.001)
#false
>
```

Problem 3

```
;----------------------------Problem 3 start--------------------------------
(define (how-many a b c)
  (cond
  [(> (calc a b c) 0) 2]
  [(= (calc a b c) 0) 1]
  [(< (calc a b c) 0) 0]))

(define (calc a b c)
  (- (* b b )(* 4 a c)))
;----------------------------Problem 3 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student; memory limit: 128 MB.
> (how-many 1 0 -1)
2
> (how-many 2 4 2)
1
>
```

Problem 4

```
;----------------------------Problem 4 start--------------------------------
(define (filter-out-symbol list symbol)
  (cond
  [(null? list) '()]
  [(eq? symbol (car list))
    (filter-out-symbol (cdr list) symbol)]
  [else (cons (car list)
    (filter-out-symbol (cdr list) symbol))]))
;----------------------------Problem 4 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student; memory limit: 128 MB.
> (filter-out-symbol '(no no a thousand times no) 'no)
(list 'a 'thousand 'times)
> (filter-out-symbol '(yes yes a thousand times no) 'yes)
(list 'a 'thousand 'times 'no)
>
```

Problem 5

```
;-----------------------------Problem 5 start--------------------------------
(define (pMinMax ls)
  (if (null? ls) '() (list(list-min ls)(list-max ls))))
(define (list-min ls)
  (cond
    [(null? (cdr ls)) (car ls)]
    [(< (car ls) (list-min (cdr ls))) (car ls)]
    [else (list-min (cdr ls))]))
(define (list-max ls)
  (cond
    [(null? (cdr ls)) (car ls)]
    [(> (car ls) (list-max (cdr ls))) (car ls)]
    [else (list-max (cdr ls))]))
;-----------------------------Problem 5 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student; memory limit: 128 MB.
> (pMinMax '(3 2 1))
(list 1 3)
> (pMinMax '())
'()
```

Problem 6

```
;-----------------------------Problem 6 start--------------------------------
(define (incnth n)
  (lambda (x) (+ n x)))
;-----------------------------Problem 6 end----------------------------------
```

```
Welcome to DrRacket, version 8.0 [cs].
Language: Intermediate Student with lambda; memory limit: 128 MB.
> ((incnth 3) 2)
5
> ((incnth -2) 3)
1
>
```