CMPSCI4250 Project2

10/11/19

```
#lang racket

(define pi 3.1416)

(define (my_area flag radius)
  (cond
    [(< radius 0) #f]
    [(= flag 1) (* pi (* radius radius))]
    [(= flag 2) (* (/ 4 3) pi (* radius radius radius))]
    [else #f]))

#|
(define (my_area flag radius)
  (if (< radius 0)
      #f
      (if (= flag 1)
          (* pi (* radius radius))
          (if (= flag 2)
              (* (/ 4 3) pi (* radius radius radius))
              #f))))
|#

;; (my_area 1 -1) => #f
;; (my_area 1 0) => 0
;; (my_area 1 1) => 3.1416
;; (my_area 2 -1) => #f
;; (my_area 2 0) => 0
;; (my_area 2 1) => 4.1888
;; (my_area 0 1) => #f
;; (my_area 3 1) => #f

(define (rem_second list)
  (if (< (length list) 2)
      '()
      (cons (car list) (cdr (cdr list)))))

;; (rem_second '()) => '()
;; (rem_second '(1)) => '()
;; (rem_second '(1 2)) => '(1)
;; (rem_second '(1 2 3)) => '(1 3)
;; (rem_second '(1 (2 3))) => '(1)
;; (rem_second '((1 2) 3)) => '((1 2))
;; (rem_second '((1 2) (3 4) (5 6))) => '((1 2) (5 6))

(define (my_union a b)
  (cond
    [(null? b) a]
    [(member (car b) a) (my_union a (cdr b))]
    [else (my_union (cons (car b) a) (cdr b))]))

;; (my_union '() '()) => '()
```

```
;; (my_union '(1 2) '()) => '(1 2)
;; (my_union '() '(1 2)) => '(2 1)
;; (my_union '(1 2) '(1 2)) => '(1 2)
;; (my_union '(1 2 (3 4)) '(1 2 (3 4))) => '(1 2)
;; (my_union '(1 2 3) '(1 2)) => '(1 2 3)
;; (my_union '(1 2) '(1 2 3)) => '(3 1 2)
;; (my_union '(1 2 1) '(1 2 1)) => '(1 2 1) NOT WORK

(define (my_delete atom list)
  (cond
    [(null? list) list]
    [(list? (car list)) (cons (my_delete atom (car list)) (my_delete atom (cdr list)))]
    [(equal? atom (car list)) (my_delete atom (cdr list))]
    [else (cons (car list) (my_delete atom (cdr list)))]))

;; (my_delete 'a '(a)) => '()
;; (my_delete 1 '(1)) => '()
;; (my_delete 'abc '(abc)) => '()
;; (my_delete 'a '(3 4 5)) => '(3 4 5)
;; (my_delete 1 '(1 2 3 (1 2 3 (1 2 3 a b c) a b c) a b c (1 2 3 (1 2 3 a b c) a b c))) => '(2 3
;; (my_delete 1 '(1 (1 (1 (1) 1) 1) 1 (1 (1 (1) 1) 1) 1)) => '((((())) (((()))))
```

PDF document made with CodePrint using Prism

```
Welcome to DrRacket, version 7.4 [3m].
Language: racket, with debugging; memory limit: 128 MB.
> (my_area 1 -1)
#f
> (my_area 1 0)
0
> (my_area 1 1)
3.1416
> (my_area 2 -1)
#f
> (my_area 2 0)
0
> (my_area 2 1)
4.1888
> (my_area 0 1)
#f
> (my_area 3 1)
#f
>
```

```
Welcome to DrRacket, version 7.4 [3m].
Language: racket, with debugging; memory limit: 128 MB.
> (rem_second '())
'()
> (rem_second '(1))
'()
> (rem_second '(1 2))
'(1)
> (rem_second '(1 2 3))
'(1 3)
> (rem_second '(1 (2 3)))
'(1)
> (rem_second '((1 2) 3))
'((1 2))
> (rem_second '((1 2) (3 4) (5 6)))
'((1 2) (5 6))
>
```

```
Welcome to DrRacket, version 7.4 [3m].
Language: racket, with debugging; memory limit: 128 MB.
> (my_union '() '())
'()
> (my_union '(1 2) '())
'(1 2)
> (my_union '() '(1 2))
'(2 1)
> (my_union '(1 2) '(1 2))
'(1 2)
> (my_union '(1 2 (3 4)) '(1 2 (3 4)))
'(1 2 (3 4))
> (my_union '(1 2 3) '(1 2))
'(1 2 3)
> (my_union '(1 2) '(1 2 3))
'(3 1 2)
> (my_union '(1 2 1) '(1 2 1))
'(1 2 1)
>
```

```
Welcome to DrRacket, version 7.4 [3m].
Language: racket, with debugging; memory limit: 128 MB.
> (my_delete 'a '(a))
'()
> (my_delete 1 '(1))
'()
> (my_delete 'abc '(abc))
'()
> (my_delete 'a '(3 4 5))
'(3 4 5)
> (my_delete 1 '(1 2 3 (1 2 3 (1 2 3 a b c) a b c) a b c (1 2 3 (1 2 3 a b c) a b c)))
'(2 3 (2 3 (2 3 a b c) a b c) a b c (2 3 (2 3 a b c) a b c))
> (my_delete 1 '(1 (1 (1 (1) 1) 1) 1 (1 (1 (1) 1) 1) 1))
'((((())) ((()))))
>
```