Design of Programming Languages Quiz 2

Alireza Habibzadeh `99109393`

1

پیاده‌سازی با تابع `flatten` رکت

```
#lang racket

(define (flatmap l func)
  (map func (flatten l)))
```

پیاده‌سازی بدون تابع `flatten` رکت

```
#lang racket

(define (flatten l)
  (cond
    [(null? l) null]
    [(list? l) (append (flatten (car l)) (flatten (cdr l)))]
    [else (list l)]))

(define (flatmap l func)
  (map func (flatten l)))
```

2

```
#lang racket

(define (rotate l n)
  (cond
    [(or (null? l) (zero? n) (not (list? l))) l]
    [(> n 0) (rotate (cons (car (reverse l))
          (reverse (cdr (reverse l)))) (- n 1))]
    [else (rotate (append (cdr l) (list (car l))) (+ n 1))]))
```

البته می‌شد جای `(last l )`، `( car (reverse l))` هم گذاشت.

3

```
#lang racket

(define (bignum-zero? X)
  (foldr (λ (p q) (and p q)) #t (map zero? X)))


(define (successor X N)
  (cond
    [(null? X) '(1)]
    [(eq? (car X) (- N 1)) (cons 0 (successor (cdr X) N))]
    [else (cons (+ (car X) 1) (cdr X))]))


(define (predecessor X N)
  (cond
    [(null? X) '(-1)]
    [(zero? (car X)) (cons (- N 1) (predecessor (cdr X) N))]
    [else (cons (- (car X) 1) (cdr X))]))

(define (display-bignum X)
  (reverse X))
```