; 2016 юли, ДИ, КН

(map (lambda (x) (append (list x) x)) '((1 2) (3 4)))

(map (lambda (f) (map (lambda (x) (f 5 x)) '(1 2 3))) (list + - \* ))

(apply list `(+ 5 8))

; да обърна внимание на това:

;(list `(+ 1 2))

; 2016 септември, ДИ, КН

; не съм сигурен, че е коректно така да си дефинирам помощна процедура; бих могъл да ползвам  
; вградената member процедура

(define (member? x l)

(cond ((null? l) #f)

((= x (car l)) #t)

(else (member? x (cdr l)))))

(define (filterByElement x m)

(filter (lambda (l) (member? x l)) m))

( filterByElement 2 '( (1 2 3) (2 3 4) (3 4 5) ) )

(apply + (map (lambda (l) (apply max l)) '((5 -2) (1 9) (6 -8))))

; 2015 юли, ДИ, КН

(map

(car (list (lambda (couple) (+ (car couple) (cdr couple)))))

(apply append '(( (1 . 2) ) ( (3 . 4) ) ) ))

(map (lambda (x) (cons x (list x))) '(1 2 3 4 5))

(map (lambda (pred) (filter pred '(1 2 3 4 5))) (list even? odd?))

; 2015 септември, ДИ, КН

(define (merge l1 l2)

(cond

((null? l1) l2)

((null? l2) l1)

((< (car l1) (car l2)) (cons (car l1) (merge (cdr l1) l2)))

(else (cons (car l2) (merge l1 (cdr l2))))))

(merge `(1 3 5 7) `(2 2 6 10))

(define (even? x)

(if (= (mod x 2) 0) #t

#f))

((lambda (l) (filter even? l)) `(1 2 3 4 5))

; 2014 септември, ДИ, КН

; искат да се напише (permutations l), който връща

; списък от списъци - пермутациите на l

; 2010 юли, ДИ, КН

(define (max a b)

(if (> a b) a

b))

(define (find-max-el l)

(define (helper l1 result)

(cond

((null? l1) result)

(else (helper (cdr l1) (max result (car l1))))))

(helper l 0))

(define (l1-without-max-el l1 max-el)

(cond

((null? l1) `())

((= max-el (car l1)) (cdr l1))

(else (cons (car l1) (l1-without-max-el (cdr l1) max-el)))))

(define (sort-desc l)

(define (helper l1 result)

(let\* ((max-el (find-max-el l1))

(new-l1 (l1-without-max-el l1 max-el)))

(cond

((null? l1) result)

(else (helper new-l1 (append result (list max-el)))))))

(helper l `()))

(define (find-max l)

(define sorted-desc-l (sort-desc l))

(define (helper l1 result)

(cond

((null? l1) result)

(else (helper (cdr l1) (+ (\* result 10) (car l1))))))

(helper sorted-desc-l 0))

(find-max `(1 1 9 8 9 3 4 6 7 0 0))