דף עזר למבחן בפונקציו-לוגי

**תנאים**

(cond ((> 3 3) 'greater)

((< 3 3) 'less)

(else 'equal))

**let**

(let\* ((x 10)

(y (+ x 6)))

**פונקצית map**

|  |  |
| --- | --- |
| (b e h) | (map cadr '((a b) (d e) (g h))) |
| (1 4 27 256) | (map (lambda (n) (expt n n)) '(1 2 3 4)) |
| (5 7 9) | (map + '(1 2 3) '(4 5 6)) |
| (l1+x,l2+x,…) | (map **(**lambda (a) (+ a ,x)**)** ,L) |

**מאקרו**

|  |  |  |
| --- | --- | --- |
| (defmacro if3 (cond1 do1 cond2 do2 doElse)  `(if ,cond1 ,do1  (if ,cond2 ,do2 ,doElse))) | (defmacro if3  (cond1 do1 cond2 do2 doElse)  `(if ,cond1 ,do1  (if ,cond2 ,do2 ,doElse))) | (defmacro limitToOne (x)    `(let ((value ,x))       (if (> value 1) 1 value))) |

|  |  |
| --- | --- |
| (defmacro switch-func (x . cases)  (define (expand-func-list funcs)  ; Expands functions list to (and (func1 \_x) (func2 \_x) ...)  ; or just to (func \_x) if there is only one function  (if (null? (cdr funcs))  `(,(car funcs) \_x)  `(and ,@(map (lambda (f) `(,f \_x)) funcs))))    (define (expand-case case)  ; Expands a single case  (if (eq? (car case) 'else)  `(else ,(cadr case))  (list (expand-func-list (car case)) (cadr case))))  `(let ((\_x ,x))  (cond ,@(map expand-case cases)))) |  |

* בתוך המאקרו, במקום לחשב בכל פעם את הערך שמתקבל, פשוט נשים אותו ב-let בתחתית המאקרו ובתוך החישובים של המאקרו נשתמש באותו המשתנה שהגדרנו

‘(let ((val, x))

(cond ,@(map expandRule rules))) ->> and then use val inside the helper in the macro

**רשימה**

הדפסת האיבר הראשון/שני/שלישי: (first\second\third lst) – חוסך את השימוש ב-car לפעמים

השוואה- (string<? string1 string2)

**בית חרושת**

* יש להחזיר בחזרה פונקציה, לדוגמה:

|  |  |  |
| --- | --- | --- |
| (define (make-encrypt code)  (lambda (msg)  (map (lambda (value) (encrypt-value value code)) msg))) | (define (makeDoFonXandL f)  ; This is the returned function.  (lambda (x L)  (map (lambda (y) (f x y)) L))) |  |

1. הפונקציה )test-ip ip prefix( מקבלת כתובת IP ותחילית (לפי הייצוג שהוגדר כרשימות), ומחזירה t# אם כתובת ה-IP נמצאת ברשת אותה מייצגת התחילית prefix, או f# אם לא.
2. פונקציית בית החרושת\* )make-ip-filter prefix( מייצרת פונקציית סינון לרשימות של כתובות IP. פונקציית סינון מקבלת רשימה של כתובות IP ומחזירה רשימה ובה רק כתובות ה-IP התואמות לתחילית prefix באמצעותה נוצרה פונקציית הסינון.

; A

(define (test-ip ip prefix)

(cond ((null? prefix) #t)

((null? ip) #f)

((eq? (car prefix) (car ip)) (test-ip (cdr ip) (cdr prefix)))

(else #f)))

; B

(define (make-ip-filter prefix)

(lambda (lst) (filter (lambda (ip) (test-ip ip prefix)) lst)))