; 1. Napište lineárně rekurzivní proceduru build-cycle,

;    která pro dané nezáporné číslo n a seznam elementů l

;    zkonstruuje seznam délky n, který obsahuje prvky ze

;    seznamu l, které se cyklicky opakují a jsou ve stejném

;    pořadí jako ve výchozím seznamu. To jest, pokud je n

;    větší než délka seznamu l, prvky na pozicích n, n+1, ..

;    se berou ze seznamu l opět od začátku.

;    Proceduru napište tak, aby byl celkový počet přístupů

;    k párům tvořící seznam l roven nejvýš n.

; (build-cycle 2 '(a b c)) ;=> (a b)

; (build-cycle 3 '(a b c)) ;=> (a b c)

; (build-cycle 4 '(a b c)) ;=> (a b c a)

; (build-cycle 5 '(a b c)) ;=> (a b c a b)

; 2. Napište iterativní proceduru root, která pro daný mnohočlen

;    (polynom) jedné proměnné, který je reprezentovaný seznamem

;    jeho koeficientů vzestupně podle jejich stupňů,

;    vrací hodnotu mnohočlenu v bodě daném druhým argumentem

;    (například tedy seznam (2 0 3 1) reprezentuje mnohočlen

;    x^3 + 3x^2 + 2). Proceduru napište tak, aby během její

;    činnosti nebyly konstruovány žádné páry.

; (root '(4) 3)     ;=> 4  = 4 \* 3^0

; (root '(4 2) 3)   ;=> 10 = 4 \* 3^0 + 2 \* 3^1

; (root '(4 2 3) 3) ;=> 37 = 4 \* 3^0 + 2 \* 3^1 + 3 \* 3^2

; 3. Napište, na co se vyhodnotí následující výraz

;    a výslednou strukturu zakreslete pomocí boxové notace

;    s ukazateli:

;(let \* ((x (list 'a 'c))

;        (y (cons x (cdr x))))

;  `((,x) ,y))

; 4. Napište, na co se vyhodnotí následující výraz a zakreslete

;    hierarchii prostředí, která během vyhodnocení vzniknou.

;    Lokální prostředí označte indexy vzestupně podle toho,

;    v jakém pořadí vznikají.

;(let \* ((x (lambda (x . y) (cons x (cdr y))))

;        (y (x 'a 'b)))

;  (map (lambda (y) (x y y))

;       y))

; 5. Napište, co počítá následující procedura (činnost procedury

;    popište tak, jako byste psali manuál)

;(define proc

;  (lambda (x)

;    (foldr (lambda (x y)

;             (if (list? x)

;                 `(,@(proc x) ,@y)

;                 `(,x ,@y)))

;           '()

;           x)))

; 6. Napište, co jsou procedury vyšších řádů. Dále vysvětlete,

;    co jsou monoidální operace a uveděte jejich příklady (alespoň 3)

; 7. Napište, na co se vyhodnotí následující výrazy

;; a

;(list cons 'a 10)

;; b

;(((lambda x) x) 'a 'b 'c)

;; c

;((let ((x -)) x) 10 20 30)

;; d

;((car (list map)) - '(1 2 3))

;; e

;(or 1 foo 2)

;; f

;(apply list '(a b) '(c d))

;; g

;(eval (let ((x 5)) (lambda y x)))

;; h

;((lambda (x) '(x '(,x ,,x))) 10)

;; i

;(cadar '((a b) (c d) (e f)))

;; j

;(let x ((y 10)) (list x y))