Paradigmata programování 2 – test1

1. Popište, jak probíhá vyhodnocení seznamu $(E_1\, E_2\, \ldots\, E_n)$, v případě, že se E_1 vyhodnotí na makro. (3 body)

2. Vyhodnoťte následující symbolický výraz a zakreslete hierarchii prostředí, která při tomto vyhodnocování vznikne. (4 bodů)

```
(let* ((x (+ 2 5))
       (y '(1 ,x 2 ,(+ x 2))))
  (let z ((i y))
    (if (null? i) x
        (begin
          (set! x (+ x (car i)))
          (z (cdr i))))))
```

3. Do následujícího kódu, který je implementací seznamu v objektově orientovaném stylu, dopište lokální proceduru `delete` pro odstranění prvku na zadaném indexu. (8 bodů)

```
(define make-list
  (lambda ()
    (let ((data '()))

      (define empty?
        (lambda ()
          (null? data)))

      (define insert
        (lambda (item index)
          (if (= index 0)
              (set! data (mcons item data))
              (let iter ((pos data)
                         (i index))
                (if (null? pos) (error "Index out of range. "))
                (if (> i 1) (iter (mcdr pos) (- i 1))
                    (set-mcdr! pos (mcons item (mcdr pos))))))))

      (define delete
        ...

      (lambda (message . args)
        (cond ((equal? message 'empty?) (empty?))
              ((equal? message 'insert) (insert (car args) (cadr args)))
              ((equal? message 'delete) (delete (car args)))
              (else (error "Unknown message!")))))))
```

4. Vytvořte makro `swap`, které prohodí hodnoty navázané na dva symboly. (5 bodů)

   Příklady použití:

```
(define x 5)
(define y #t)
(swap x y)
x
y
(swap x -5)
```

   Výstupy příkladů:

```
#t
5
Makro pracuje jen se symboly!
```