Úloha 1

- definujte funkce:
  - **(druhy** L**)** – vrací druhý prvek L
    (druhy '(1 2 3 4)) → 2
  - **(posledni** L**)** – vrací poslední prvek L
    (posledni '(1 2 3 4)) → 4
  - **(bezkonce** L**)** – vrací seznam kromě posledního prvku
    (bezkonce '(1 2 3 4)) → (1 2 3)

Řešení 1

```
(define (druhy L)
    (car (cdr L)))

(define (posledni L)
    (car (reverse L)))

(define (bezkonce L)
    (reverse (cdr (reverse L))))
```

Úloha 2

- definujte funkci **(bezdruheho L)**, která vrátí seznam bez druhého prvku
- např. (bezdruheho '(1 2 3 4 5)) → (1 3 4 5)
- nápověda: použijte cons

Řešení 2

```
(define (bezdruheho L)
    (cons (car L)
          (cdr (cdr L))))
```

Úloha 3

- vytvořte funkci (**prohod** L), která dostane dvouprvkový seznam a prohodí pořadí jeho prvků bez použití funkce reverse
- např. (prohod '(1 2)) → (2 1)
- zvažte různé alternativy, jak funkci implementovat (list, cons, append)

Řešení 3

```
(define (prohod L)
    (list (car (cdr L))
          (car L)))

(define (prohod2 L)
    (cons (car (cdr L))
          (list (car L))))

(define (prohod3 L)
    (append (cdr L)
            (list (car L))))
```

Úloha 4

- vytvořte booleovskou funkci (**mod23?** N), která testuje, zda je N dělitelné zároveň 2 i 3
  - např. (mod23? 12) → #true, (mod23? 15) → #false
- vytvořte číselnou funkci (**omezit** mini x maxi), která vrací x omezené na interval od mini do maxi
  - (omezit 0 19 100) → 19
  - (omezit 0 -5 100) → 0
  - (omezit 0 250 100) → 100

Řešení 4

```
(define (mod23? N)
    (and (even? N)
         (= 0 (modulo N 3))))

(define (omezit mini x maxi)
    (cond
        [(< x mini) mini]
        [(> x maxi) maxi]
        [else x]))

(define (omezit2 mini x maxi)
    (min maxi (max mini x)))
```

Úloha 5

- vytvořte funkci (**suda?** L), která testuje, zda jsou všechna čísla v seznamu L sudá

Řešení 5

```
(define (suda? L)
    (cond
        [(empty? L) #t]
        [(odd? (car L)) #f]
        [else (suda? (cdr L))]))
```

Úloha 6

- vytvořte funkci (**secti** L), která dostane seznam čísel a vrátí jejich součet
- např. (secti '(25 31 6 10)) → 72

Úloha 6

- vytvořte funkci (**secti** L), která dostane seznam čísel a vrátí jejich součet
- např. (secti '(25 31 6 10)) → 72
- nápověda:
  - součet prázdného seznamu je 0
  - jinak je součet = první prvek + součet zbytku

Řešení 6

```
(define (secti L)
    (if (empty? L) 0
        (+ (car L)
            (secti (cdr L)))))

(define (secti2 L)
    (cond
        [(empty? L) 0]
        [else (+ (car L)
                 (secti2 (cdr L)))]))
```