

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 1. Vypište předposlední prvek P seznamu S.
% Použití: (predposledni (quote(1 2 3 4 5)))

(defun predposledni (S)
  (cond
    ((null S) nil)
    ((null (rest S)) nil)
    ((null (rest (rest S))) (first S))
    (T (predposledni (rest S)) )
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 2. Vypište poslední prvek P seznamu S.

(defun posledni (S)
  (cond
    ((null S) nil)
    ((null (rest S)) (first S))
    (T (posledni (rest S)) )
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 3. Vložte prvek P na začátek seznamu S.
% Použití: (vlozZ (quote(2 3 4 5)) 1)

(defun vlozZ (S P)
  (cond
    ((null S) (list P) ) % list vytvoří nový S
    (T (cons P S) ) % cons spojení P a S
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 4. Vložte prvek P na konec seznamu S.

(defun vlozK (S P)
  (cond
    ((null S) (cons P nil) )
    (T (append S (cons P nil)))
  )
)

% append spojení více seznamů S

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 5. Smažte prvek P ze seznamu S.

(defun smaz (S P)
  (cond
    ((null S) S )
    ((equal (first S) P) (rest S) )
    (T (cons (first S) (smaz (rest S) P)))
  )
)

```

```

% equal pro porovnání

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 6. Vypište počet prvků N lineárního seznamu S.

(defun delkaL (S)
  (cond
    ((null S) 0 )
    (T (+ 1 (delkaL (rest S)) ) )
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 7. Vypište počet atomických prvků N seznamu S.

(defun delkaA (S)
  (cond
    ((null S) 0 )
    ((atom (first S)) (+ 1 (delkaA (rest S)) ) )
    (T (+ (delkaA (first S)) (delkaA (rest S)) ) )
  )
)

% atom, je P atomem?

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 8. Vypište nejmenší prvek P seznamu S.

(defun minimum (S)
  (cond
    ((null S) nil)
    ((null (rest S)) (first S) )
    (<= (first S) (minimum (rest S))) (first S) )
    (T (minimum (rest S)))
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 9. Vypište největší prvek P seznamu S.

(defun maximum (S)
  (cond
    ((null S) nil)
    ((null (rest S)) (first S) )
    (>= (first S) (maximum (rest S))) (first S) )
    (T (maximum (rest S)))
  )
)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% 10. Vypište součet nejmenšího a největšího prvku seznamu S.

(defun soucetMinMax (S)
  (cond
    ((null S) nil)
    ((null (rest S)) nil )
    (T (+ (minimum S) (maximum S)) )
  )
)

```

```

)
)
(defun minimum (S)
  (cond
    ( (null (rest S)) (first S) )
    ( (<= (first S) (minimum (rest S))) (first S) )
    (T (minimum (rest S))) )
  )
)

(defun maximum (S)
  (cond
    ( (null (rest S)) (first S) )
    ( (>= (first S) (maximum (rest S))) (first S) )
    (T (maximum (rest S))) )
  )
)

*****
% 11. Vypište součet nejmenšího a druhého nejmenšího prvku P
% seznamu S.
% (Není korektní.)

(defun soucetMinMin (S)
  (cond
    ( (null S) nil )
    ( (null (rest S)) nil )
    (T (+ (minimum S) (minimum (smaz S (minimum S)))) )
  )
)

(defun minimum (S)
  (cond
    ( (null (rest S)) (first S) )
    ( (<= (first S) (minimum (rest S))) (first S) )
    (T (minimum (rest S))) )
  )
)

(defun smaz (S P)
  (cond
    ( (null S) S )
    ( (equal (first S) P) (rest S) )
    (T (cons (first S) (smaz (rest S) P))) )
  )
)

% equal pro porovnání

*****
% 12. Vypište součet největšího a druhého největšího prvku P
% seznamu S.

*****
% 13. Vypište součet všech prvků P v seznamu S.

(defun soucetAll (S)

```

```

(cond
  ( (null S) nil )
  ( (null (rest S)) (first S) )
  (T (+ (first S) (soucetAll (rest S)))) )
)

*****
% 14. Vypište index nejmenšího prvku P seznamu S.

(defun indexMin (S)
  (cond
    ( (null S) nil )
    ( (equal (first S) (minimum S)) 1 )
    (T (+ 1 (indexMin (rest S)))) )
  )
)

(defun minimum (S)
  (cond
    ( (null (rest S)) (first S) )
    ( (<= (first S) (minimum (rest S))) (first S) )
    (T (minimum (rest S))) )
  )
)

*****
% 15. Vypište index největšího prvku P seznamu S.

(defun indexMax (S)
  (cond
    ( (null S) nil )
    ( (equal (first S) (maximum S)) 1 )
    (T (+ 1 (indexMax (rest S)))) )
  )
)

(defun maximum (S)
  (cond
    ( (null (rest S)) (first S) )
    ( (>= (first S) (maximum (rest S))) (first S) )
    (T (maximum (rest S))) )
  )
)

*****
% 16. Zjistěte, zda je seznam S monotónní, tzn. jestli je
% rostoucí, popř. klesající.

(defun monotonnost (S)
  (cond
    ( (null S) nil )
    (T (or (rostouci S) (klesajici S))) )
  )
)

(defun rostouci (S)
  (cond

```

```

    ( (null (rest S)) T )
    ( (<= (first S) (first (rest S))) (rostouci (rest S)) )
  )
)

(defun klesajici (S)
  (cond
    ( (null (rest S)) T )
    ( (>= (first S) (first (rest S))) (klesajici (rest S)) )
  )
)

#####
% 17. Seřadte seznam od nejmenšího prvku po největší.

#####
% 18. Seřadte seznam od největšího prvku po nejmenší.

#####
% 19. Sjednoďte 2 množiny K, L a vraťte množinu M.

(defun sjednoceni2 (K L)
  (cond
    ((null K) L)
    ((null L) K)
    ((member (first K) L) (sjednoceni2 (rest K) L))
    (T (cons (first K) (sjednoceni2 (rest K) L) ) )
  )
)

% Nebo:

(defun sjednoceni2 (K L)
  (cond
    ((null K) L)
    ((obsahuje L (first K)) (sjednoceni2 (rest K) L))
    (T (cons (first K) (sjednoceni2 (rest K) L) ) )
  )
)

(defun obsahuje (K P)
  (cond
    ((null K) nil)
    ((= (first K) P) T)
    (T (obsahuje (rest K) P))
  )
)

#####
% 20. Sjednoďte 3 množiny J, K, L a vraťte množinu M.

(defun sjednoceni3 (K L M)
  (cond
    (T (sjednoceni2 M (sjednoceni2 K L)))
  )
)

```

```

(defun sjednoceni2 (K L)
  (cond
    ((null K) L)
    ((null L) K)
    ((member (first K) L) (sjednoceni2 (rest K) L))
    (T (cons (first K) (sjednoceni2 (rest K) L) ) )
  )
)

#####
% 21. Vytvořte průnik 2 množin K, L a vraťte množinu M.

(defun prunik2 (K L)
  (cond
    ((null K) nil)
    ((null L) nil)
    ((member (first K) L) (cons (first K) (prunik2 (rest K) L) ) )
    (T (prunik2 (rest K) L) )
  )
)

% Nebo:

(defun prunik2 (K L)
  (cond
    ((null K) nil)
    ((null L) nil)
    ((obsahuje L (first K)) (cons (first K) (prunik2 (rest K) L) ) )
    (T (prunik2 (rest K) L) )
  )
)

(defun obsahuje (K P)
  (cond
    ((null K) nil)
    ((= (first K) P) T)
    (T (obsahuje (rest K) P))
  )
)

#####
% 22. Vytvořte průnik 3 množin J, K, L a vraťte množinu M.

(defun prunik3 (K L M)
  (cond
    (T (prunik2 M (prunik2 K L) ) )
  )
)

(defun prunik2 (K L)
  (cond
    ((null K) nil)
    ((null L) nil)
  )
)

```

```

(member (first K) L)
      (cons (first K) (prunik2 (rest K) L) ) )
(T
  )
)

```

 * 23. Vytvořte sjednocení 2 množin K a L, přičemž množina K
 * vznikla prunikem množiny I a J. Vraťte množinu M.

```

(defun sjednoceni (I J L)
  (cond
    (T (sjednoceni2 (prunik2 I J) L) )
  )
)

```

```

(defun sjednoceni2 (K L)
  (cond
    ((null K) L)
    ((null L) K)
    ((member (first K) L) (sjednoceni2 (rest K) L))
    (T (cons (first K) (sjednoceni2 (rest K) L) ) )
  )
)

```

```

(defun prunik2 (I J)
  (cond
    ((null I) nil)
    ((null J) nil)
    ((member (first I) J)
      (cons (first I) (prunik2 (rest I) J) ) )
    (T
      (cons (first I) (prunik2 (rest I) J) )
    )
  )
)

```

 * 24. Vytvořte prunik 2 množin K a L, přičemž množina K
 * vznikla sjednocením množiny I a J. Vraťte množinu M.

```

(defun prunik (I J L)
  (cond
    (T (prunik2 (sjednoceni2 I J) L) )
  )
)

```

```

(defun sjednoceni2 (I J)
  (cond
    ((null I) J)
    ((null J) I)
    ((member (first I) J) (sjednoceni2 (rest I) J))
    (T (cons (first I) (sjednoceni2 (rest I) J) ) )
  )
)

```

```

(defun prunik2 (K L)
  (cond
    ((null K) nil)
    ((null L) nil)
  )
)

```

```

(member (first K) L)
      (cons (first K) (prunik2 (rest K) L) ) )
(T
  )
)

```

 * 25. Vypočítejte faktoriál z čísla F a vraťte N.

```

(defun faktorial (F)
  (cond
    ((< F 0) nil)
    ((= F 0) 1)
    (T (* F (faktorial (- F 1))))
  )
)

```