

```

#####
* 7. 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)) )
  )
)
#####
* 7. Vypište poslední prvek P seznamu S.
(defun posledni (S)
  (cond
    ((null S) nil)
    ((null (rest S)) (first S))
    (T (posledni (rest S)) )
  )
)
#####
* 8. Vložte prvek P na začátek seznamu S.
% Použití: (vloz (quote (2 3 4 5)) 1)
(defun vloz (S P)
  (cond
    ((null S) (list P) ) % list vytvoří nový S
    (T (cons P S) ) % cons spojení P a S
  )
)
#####
* 9. Vložte prvek P na konec seznamu S.
(defun vlozK (S P)
  (cond
    ((null S) (cons P nil) )
    (T (append S (cons P nil)) )
  )
)
#####
* 10. Smazte 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)))
  )
)

```

1

```

#####
* equal pro porovnání
#####
* 9. Vypište počet prvků N lineárního seznamu S.
(defun delka (S)
  (cond
    ((null S) 0)
    (T (+ 1 (delka (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)) )
  )
)

```

2

```

)
}

(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.
  * (Není korektní.)

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

(defun soucetAll (S)

```

[illegible]

```

    ( (null (rest S))
      ( (=< (first S) (first (rest S))) (rostrouci (rest S)) )
    )
  )
  (defun klasajici (s)
    (cond
      ((null (rest S))
        (T)
      )
      (>= (first S) (first (rest S))) (klasajici (rest S)) )
    )
  )
  %17. Seřadte seznam od nejmenšího prvku po největší.
  %18. Seřadte seznam od největšího prvku po nejmenší.
  %19. Sjednotte 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
      )
      (T
        (obsahuje (rest K) P)
      )
    )
  )
  %20. Sjednotte 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 pruník 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
        (cons (first K) (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
      )
      (T
        (obsahuje (rest K) P)
      )
    )
  )
  %22. Vytvořte pruník 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) )
    )
  )
}
%23. Vytvořte sjednocení 2 množin K a L, přičemž množina K
% vznikla pruníkem 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
    (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)
      null)
    ((null L)
      null)
  )
)

```

7

```

    (member (first K) L)
      (cons (first K) (prunik2 (rest K) L) )
    )
  )
}
%25. Vypočítejte faktoriál z čísla F a vraťte N.

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

```

8