

A Prolog-eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 2.b) hivatkozzon!

1. Döntse el, mi lesz az alábbi Prolog kérdések eredménye (hiba, meghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőnek. (5 pont)

(a)  $P \text{ is } [1,2], P = [1,2].$

(b)  $U/V = a+b/2.$

(c)  $A = B+2, B \text{ is } 3*2.$

(d)  $X \text{ is } 2+2, X ::= 2*2.$

(e)  $[G|H] = [a,b].$

2. Írja fel az alábbi egyenlőségek bal- és jobboldalának alapstruktúra alakját, vagy rajzolja fel a fastruktúrájukat! Adja meg, milyen változó-behelyettesítéseket eredményeznek ezek az egyesítések! (9 pont)

(a)  $[X, X|Y] = [1+B, A+2*3, A].$

(b)  $.(f(K,a+b), [N|M]) = [N, f(1/2,L), K].$

3. Tegyük fel, hogy az alábbi programot betöltöttük a SICStus Prolog rendszerbe.

```
p([], X, X).
p([X|L], X, X).
p([X|L], _Y, Z) :-
    p(L, X, Z).
```

Állapítsa meg, hogy a feltett kérdésre válaszul a rendszer milyen behelyettesítést ad az X változónak! Sorolja fel az összes megoldást, a rendszer által előállított sorrendben és írja le ezeket pontosvesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

(a)  $p([], a, X).$

(b)  $p([a], a, X).$

(c)  $p([a], b, X).$

(d)  $p([a,b,b,c,d,e], a, X).$

(e)  $p([a,b,b,b,c], b, X).$

Tekintse a fenti p/3 eljárásra épülő alábbi p/2 eljárást:

```
% p(L, Y): Az L listának Y egy olyan eleme,....
p([X|T], Y) :- p(T, X, Y).
```

- (f) Egészítse ki teljes kijelentő mondatná a fenti fejkommentet!

(7 pont)

4. Egy nem negatív egész számokból álló listát  $n - m$  emelkedőnek hívunk, ha a 0 elemektől eltekintve az  $n + 1, n + 2, \dots, m$  (esetleg üres) számsorozatot tartalmazza, ebben a sorrendben. (Speciálisan: egy üres, vagy csak 0 számokat tartalmazó listát  $n - n$  emelkedőnek tekintünk, tetszőleges  $n$  esetén.)

Írjon olyan Prolog eljárást, amely megfelel az alábbi fejkommentnek! Segédeljárást nem használhat!

(9 pont)

```
% emelkedo(+L, +N, ?M): L egy N-M emelkedő.
```

```
% (L és N bemenő, M kimenő paraméter.)
```

```
| ?- emelkedo([5,6], 4, M).          ----> M = 6 ? ; no
| ?- emelkedo([6,5], 4, M).          ----> no
| ?- emelkedo([0,1,2,0,3], 0, M).     ----> M = 3 ? ; no
| ?- emelkedo([0,1,0,0,2,0,3,4,0], 0, M). ----> M = 4 ? ; no
| ?- emelkedo([0,0,0,0], 9, M).       ----> M = 9 ? ; no
| ?- emelkedo([], 123, M).            ----> M = 123 ? ; no
```

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	implode	: char list -> string
map	: ('a -> 'b) -> 'a list -> 'b list	ord	: char -> int
op::	: 'a * 'a list -> 'a list	str	: char -> string
op^	: string * string -> string		

6. Mi az **f típusa** az alábbi (egymástól független) deklarációkban? (6 pont)

- (a) `fun f x y = (x, y)`
- (b) `fun f (x, y) = x y`
- (c) `fun f x y = y x`

7. Mi az **x értéke** és **típusa** az alábbi (egymástól független) deklarációkban? (6 pont)

- (a) `val x = "csi" ^ implode [#"p",#"a"]`
- (b) `val x = List.filter (fn x => x > 0.0) [1.3, ~2.2, 3.0, 0.0]`
- (c) `val x = map op- [(5,2), (6,~3), (~4,~7), (1,1)]`

8. Mit kap eredményül, ha az alábbi **f** függvényt a hat számjegyből álló, **ÉÉHHNN** alakban fűzérként megadott **SAJÁT** születési dátumára alkalmazza (pl. `f "820431"`)? Válaszát indokolja! (6 pont)

```
fun f t = let
    fun g (x::y::zs) = if x < y then y::g(x::zs) else x::g(y::zs)
      | g zs = zs : char list
  in
    implode(g(explode t))
  end
```

9. Írja meg az alábbi **f** függvény egy ekvivalens változatát a megadott könyvtári függvényekkel! (6 pont)

```
fun f [] = ""
  | f (x::xs) = str x ^ f xs
```

Az alkalmazandó függvények: `foldr`, `map`, `op^`, `str`. Használja fel mindet! Segédfüggvényt **NEM** definiálhat (`fn` jelöléssel sem)!

10. Adott egy előjeles egész számokat tartalmazó lista. Írjon függvényt `posNeg` néven egy olyan lista előállítására, amelyben az argumentumlista nemnegatív elemei az eredménylista elején, negatív elemei pedig az eredménylista végén vannak, tetszőleges sorrendben. (6 pont)

```
(* posNeg : int list -> int list
   posNeg xs = olyan lista, amelynek az elején az xs nemnegatív, a végén
               pedig az xs negatív elemei vannak, tetszőleges sorrendben
*)
```

Példák: `posNeg [] = []`  
`posNeg [~2, ~3, ~1] = [~1, ~3, ~2]`  
`posNeg [2, 0, 1] = [0, 2, 1]`  
`posNeg [~2, 2, ~1, ~3, 1] = [1, 2, ~2, ~3, ~1]`

A Prolog-eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 2.b) hivatkozzon!

1. Döntse el, mi lesz az alábbi Prolog kérdések eredménye (hiba, megghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőknek. (5 pont)

- (a)  $a*b+2 = C+D$ .
- (b)  $[X,Y|Z] = [a,b,c]$ .
- (c)  $I \text{ is } 3*4, J ::= I+1$ .
- (d)  $U = 1, U \text{ is } U+1$ .
- (e)  $S \text{ is } 2*4, T = S+2$ .

2. Írja fel az alábbi egyenlőségek bal- és jobboldalának alapstruktúra alakját, vagy rajzolja fel a fastruktúrájukat! Adja meg, milyen változó-behelyettesítéseket eredményeznek ezek az egyesítések! (9 pont)

- (a)  $.(X, [4/2+Y, Y]) = [U+1, X|V]$ .
- (b)  $f(.(a+b, U), [1, V, V*c]) = f(W, [S|W])$ .

3. Tegyük fel, hogy az alábbi programot betöltöttük a SICStus Prolog rendszerbe.

```
p([X|_], [X|_], X).
p([_|L1], [_|L2], X) :-
    p(L1, L2, X).
```

Állapítsa meg, hogy a feltett kérdésre válaszul a rendszer milyen behelyettesítést ad az X változónak! Sorolja fel az összes megoldást, a rendszer által előállított sorrendben és írja le ezeket pontosvesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

- (a)  $p([], [a], X)$ .
- (b)  $p([a], [a], X)$ .
- (c)  $p([b,d], [c,d], X)$ .
- (d)  $p([a,b,d], [a,c,d], X)$ .
- (e)  $p([a,b,b,b,c,d], [b,c,b,a,c,d], X)$ .

Tekintse a fenti p/3 eljárásra épülő alábbi p/2 eljárást:

```
% p(L, Y): Az L listának Y egy olyan eleme,....
p(L, Y) :- p(L, [_|L], Y).
```

- (f) Egészítse ki teljes kijelentő mondatná a fenti fejkommentet! (7 pont)

4. Egy nem negatív egész számokból álló listát  $n/m$  lejtőnek hívunk, ha a 0 elemektől eltekintve az  $n, n-1, \dots, m$  (esetleg üres) számsorozatot tartalmazza, ebben a sorrendben. (Speciálisan: egy üres, vagy csak 0 számokat tartalmazó listát  $n/m$  lejtőnek tekintünk, ahol  $m = n + 1$ , tetszőleges  $n$  esetén.)

Írjon olyan Prolog eljárást, amely megfelel az alábbi fejkommentnek! Segédeljárást nem használhat! (9 pont)

```
% lejto(+L, +N, -M): L egy N/M lejtő
% (L és N bemenő, M kimenő paraméter.)
```

```
| ?- lejto([6,5], 6, M).      ----> M = 5 ? ; no
| ?- lejto([5,6], 5, M).      ----> no
| ?- lejto([0,3,2,0,1], 3, M). ----> M = 1 ? ; no
| ?- lejto([0,5,4,0,0,3,0,2,0], 5, M). ----> M = 2 ? ; no
| ?- lejto([0,0,0,0], 9, M).  ----> M = 10 ? ; no
| ?- lejto([], 123, M).      ----> M = 124 ? ; no
```

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	implode	: char list -> string
map	: ('a -> 'b) -> 'a list -> 'b list	ord	: char -> int
op::	: 'a * 'a list -> 'a list	str	: char -> string
op^	: string * string -> string		

6. Mi az **f típusa** az alábbi (egymástól független) deklarációkban? (6 pont)

- (a) fun f (y, x) = (x, y)
- (b) fun f (x, y) = y x
- (c) fun f x y = x y

7. Mi az **x értéke** és **típusa** az alábbi (egymástól független) deklarációkban? (6 pont)

- (a) val x = implode("#"p" :: #"a" :: explode "csi")
- (b) val x = map op^ [("al", "ma"), ("kar", "fa"), ("al", "fa")]
- (c) val x = List.filter (fn x => x) [3.0>4.0, 4.7>2.4, ~3.0<=0.4, 0.7>=6.0]

8. Mit kap eredményül, ha az alábbi **f** függvényt a hat számjegyből álló, **ÉÉHHNN** alakban füzérként megadott **SAJÁT** születési dátumára alkalmazza (pl. f "820431")? Válaszát indokolja! (6 pont)

```
fun f t = let
    fun h (u::v::ws) = if v >= u then u::h(v::ws) else v::h(u::ws)
      | h ws = ws : char list
  in
    implode(h(explode t))
  end
```

9. Írja meg az alábbi **f** függvény egy ekvivalens változatát a megadott könyvtári függvényekkel! (6 pont)

```
fun f [] = 0
  | f (x::xs) = ord x + f xs
```

Az alkalmazandó függvények: foldr, map, op+, ord. Használja fel mindet! Segédfüggvényt NEM definiálhat (fn jelöléssel sem)!

10. Adott egy valós számokat tartalmazó lista. Írjon függvényt **intRat** néven egy olyan lista előállítására, amelyben az argumentumlista egész értékű elemei az eredménylista elején, nem egész értékű elemei pedig az eredménylista végén vannak, tetszőleges sorrendben. (6 pont)

```
(* intRat : real list -> real list
   intRat xs = olyan lista, amelynek az elején az xs egész értékű, a végén
               pedig a nem egész értékű elemei vannak, tetszőleges sorrendben
*)
```

Példák: intRat [] = []  
 intRat [2.0, 3.0, 1.0] = [1.0, 3.0, 2.0]  
 intRat [2.2, 3.1, 1.44] = [2.2, 1.44, 3.1]  
 intRat [2.0, 2.2, 1.44, 3.1, 1.0] = [1.0, 2.0, 2.2, 3.1, 1.44]