

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 értelmezőknek. (5 pont)

- (a) $.(X, Y) = [a, b]$.
 (b) $X ::= 2+3*4, X = 14$.
 (c) $2+4+6 = X+Y$.
 (d) $X \text{ is } 3*2, X+3 ::= 9$.
 (e) $A*B = 1*2+3$.

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

- (a) $[a*b-X, c+X] = [Y-3|Z]$.
 (b) $g(V*W, [1*2+3|Z]) = g(K, [K+L, L])$.

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

```
p([Y,Z|_], X, Z) :- Z ::= X+Y.
p([X|L], _, Z) :-
    p(L, X, Z).
```

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

- (a) $| \text{?- } p([5], 5, X)$.
 (b) $| \text{?- } p([5,6], 1, X)$.
 (c) $| \text{?- } p([1,1,2,4], 0, X)$.
 (d) $| \text{?- } p([1,2,3,5,1,6], 1, X)$.
 (e) $| \text{?- } p([1,4,5,1,2,3,3,6], 0, X)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

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

- (f) Írja le a p/2 eljárás jelentését deklaratív módon, azaz egészítse ki teljes kijelentő mondatná a fenti fejkommentet! Írja le azt is, hogy az eljárás milyen sorrendben állítja elő a megoldásokat! (8 pont)

4. Tekintsünk egy X-Y párokból álló listát. Ebben a listában két szomszédos elemet *összevonhatónak* hívunk, ha az első pár második tagja és a második pár első tagja megegyezik. Az ilyen X-Y és Y-Z elemek *eredőjének* hívjuk az X-Z párt. Írjon olyan Prolog eljárást, amely felsorolja egy adott listában szereplő összevonható elemeket és azok eredőjét. Segédeljárást nem definiálhat! (8 pont)

```
% eredo_elem(+L, -A, -B, -E): A párokból álló L listában A és B két
% összevonható szomszédos elem, amelyek eredője E.
% L bemenő, A, B és E kimenő paraméterek.
```

```
| ?- eredo_elem([a-e,a-c,c-b,c-d], A, B, E).
A = a-c, B = c-b, E = a-b ? ;
no
| ?- eredo_elem([a-b,a-c,c-d,d-e,b-a,a-c], A, B, E).
A = a-c, B = c-d, E = a-d ? ;
A = c-d, B = d-e, E = c-e ? ;
A = b-a, B = a-c, E = b-c ? ;
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, relációk és op^{\wedge} kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldl	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	rev	: 'a list -> 'a list
op@	: 'a list * 'a list -> 'a list	Char.isAlpha	: char -> bool
op::	: 'a * 'a list -> 'a list	ord	: char -> int
map	: ('a -> 'b) -> 'a list -> 'b list	chr	: int -> char

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **statikus szemantikai hiba** van. Melyek ezek? (7 pont)

- (a) [op>("#a", "b"), (1, 2) <> (1, 2, 3), true = false]
- (b) (2*3 = 3+3, chr 95, ~9) = (6*1, "b", 0-5-4)
- (c) foldl ~1 [4, 2, 6, 4, 1, 2] op/

6. Mi a **q** értéke az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val (_::_::q::_) = explode "hús" @ rev ["t", "é", "v"]
- (b) val (_::_::q) = List.filter Char.isAlpha (explode "4a3r2ald")
- (c) val q = map (fn (a, b) => a > b) [(7, 2*3), (1, 2), (ord #"A", ord #"Z")]

7. Tekintsük a következő függvénydefiníciókat! (8 pont)

```
(* sf : 'a list * 'a list -> 'a list *)
fun sf (ms, ns) = foldl op:: ns ms
(* f1 : ('a * 'a) list -> 'a list *)
(* f2 : ('a * 'b) list * 'a list * 'b list -> 'a list * 'b list *)
fun f1 mns = let fun f2 ((i,j)::ijs, ms, ns) = f2(ijs, i::ms, j::ns)
                | f2 ([], ms, ns) = (ms, ns)
            in
                sf(f2(mns, [], []))
            end
```

Mi az **x** értéke az alábbi, egymástól független deklarációk kiértékelése után?

- (a) val x = sf ([0, 2, 4], [1, 3, 5])
- (b) val x = sf ([1.0, 2.0, 3.0], [0.0, 0.0])
- (c) val x = f1 [(0,1), (2,3), (4,5)]
- (d) val x = f1 [("a","A"), ("b","B"), ("c","C")]
- (e) val x = foldl op^ "" (f1 [("n",""), ("o",""), ("p","")])

8. Tekintsük az alábbi adattípus-deklarációt:

```
datatype 'a D = X of 'a D list | Y of 'a
```

Emelkedőnek nevezzük az olyan (x, y, z) hármasokat, amelyekre $x < y < z$. Írjon olyan függvényt emelkedők néven, amely egy $(int * int * int)$ D típusú adatstruktúrában található emelkedő hármasok számát adja eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.

(8 pont)

```
(* emelkedok : (int * int * int) D -> int
   emelkedok t = a t-beli emelkedő hármasok száma *)
```

```
Példák: emelkedok(X[]) = 0;
         emelkedok(Y(3,4,4)) = 0;
         emelkedok(Y(3,4,5)) = 1;
         emelkedok(X[X[],X[],Y(3,4,5)]) = 1;
         emelkedok(X[X[Y(2,5,8),Y(6,9,9)],X[X[Y(8,8,9)],Y(~2,3,8)],X[],Y(3,4,5)]) = 3;
```

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 értelmezőknek. (5 pont)

- (a) $a-b-c = X-Y$.
 (b) $[A,B,C] = [1,2]$.
 (c) $X = 1+2, \setminus + X = 3$.
 (d) $X = a-U, X = V-2*3$.
 (e) $X > 2, X \text{ is } 1+2$.

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) $[X, X|Y] = [c+k, L+a*b, L]$.
 (b) $f(P-Q, [P, Q]) = f(2*v-5, [v*u|S])$.

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

```
p([Y,Z|_], X, Y) :- Y == Z-X.
p([X|L], _, Z) :-
    p(L, X, Z).
```

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

- (a) $| \text{?- } p([5], 5, X)$.
 (b) $| \text{?- } p([5,6], 1, X)$.
 (c) $| \text{?- } p([2,3,5,6], 1, X)$.
 (d) $| \text{?- } p([1,2,3,5,1,6], 1, X)$.
 (e) $| \text{?- } p([1,4,5,1,2,3,3,6], 0, X)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

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

- (f) Írja le a p/2 eljárás jelentését deklaratív módon, azaz egészítse ki teljes kijelentő mondatná a fenti fejkommentet! Írja le azt is, hogy az eljárás milyen sorrendben állítja elő a megoldásokat! (8 pont)

4. Tekintsünk egy X-Y párokból álló számlistát. Ebben a listában két szomszédos elemet *rokonnak* hívunk, ha az első pár tagjainak összege megegyezik a második pár tagjainak összegével. Írjon olyan Prolog eljárást, amely felsorolja egy adott listában szereplő rokon elemeket és azok közös összegét. Segédeljárást nem definiálhat! (8 pont)

```
% rokonok(+L, -A, -B, -S): A számpárokból álló L listában A és B két
% rokon szomszédos elem, amelyek közös összege S.
% L bemenő, A, B és S kimenő paraméterek.
```

```
| ?- rokonok([1-2,2-3,4-1,2-2], A, B, S).
A = 2-3, B = 4-1, S = 5 ? ;
no
| ?- rokonok([1-2,2-3,1-4,3-2,1-8,1-7,4-4], A, B, S).
A = 2-3, B = 1-4, S = 5 ? ;
A = 1-4, B = 3-2, S = 5 ? ;
A = 1-7, B = 4-4, S = 8 ? ;
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, relációk és op^{\wedge} kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	Char.isLower	: char -> bool
map	: ('a -> 'b) -> 'a list -> 'b list	ord	: char -> int
op@	: 'a list * 'a list -> 'a list	tl	: 'a list -> 'a list
op::	: 'a * 'a list -> 'a list		

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két **statikus szemantikai hiba** van. Melyek ezek? (7 pont)

- (a) `[2 < 2.3, "a" > #"b", 7 mod 3 = 0]`
- (b) `(64.0, 1*4, ~12) = (ord #"b", 2+2 = 4 div 1, 0-7-7)`
- (c) `foldr (op div) ~10 [1.4, 0.6, 3.4, 4.7, 2.1, 1.9]`

6. Mi a **p értéke** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) `val (_::_:~:~:~:~:~:~:~) = tl [#"v", #"a", #"t"] @ explode "ta"`
- (b) `val (_::_:~:~:~:~:~:~:~) = List.filter Char.isLower (explode "1d2a3r4a")`
- (c) `val p = map (fn (a, b) => a < b) [(4+2, 7), (2, 1), (ord #"9", ord #"8")]`

7. Tekintsük a következő függvénydefiníciókat! (8 pont)

```
(* sg : 'a list * 'a list -> 'a list *)
fun sg (ms, ns) = foldr op:: ns ms
(* g1 : ('a * 'a) list -> 'a list *)
(* g2 : 'a list * 'b list * ('a * 'b) list -> 'b list * 'a list *)
fun g1 mns = let fun g2 (ms, ns, (i,j)::ijs) =
                g2(j::ms, i::ns, ijs)
                | g2 (ms, ns, []) = (ns, ms)
            in
                sg(g2([], [], mns))
            end
```

Mi az **x értéke** az alábbi, egymástól független deklarációk kiértékelése után?

- (a) `val x = sg ([0, 2, 4], [1, 3, 5])`
- (b) `val x = sg ([1.0, 2.0, 3.0], [0.0, 0.0])`
- (c) `val x = g1 [(0,1), (2,3), (4,5)]`
- (d) `val x = g1 [("a","A"), ("b","B"), ("c","C")]`
- (e) `val x = foldr op^ "" (g1 [("n",""), ("o",""), ("p","")])`

8. Tekintsük az alábbi adattípus-deklarációt:

```
datatype 'a H = U of 'a H list | V of 'a
```

Süllyedőnek nevezzük az olyan (x, y, z) hármasokat, amelyekre $x > y > z$. Írjon olyan függvényt süllyedok néven, amely egy $(int * int * int)$ H típusú adatstruktúrában található süllyedő hármasok számát adja eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.

(8 pont)

```
(* süllyedok : (int * int * int) H -> int
   süllyedok t = a t-beli süllyedő hármasok száma *)
```

```
Példák: süllyedok(U[]) = 0;
         süllyedok(V(3,3,4)) = 0;
         süllyedok(V(5,4,3)) = 1;
         süllyedok(U[U[],U[],V(5,4,3)]) = 1;
         süllyedok(U[U[V(8,5,2),V(6,6,9),U[U[V(8,7,7)],V(8,3,~2)],U[],V(5,4,3)]) = 3;
```