

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) $[_|Z]=[1,d,3]$.
 (b) $R \text{ is } 3+1, Z = 2+2, Z \text{ is } R$.
 (c) $\text{append}([_,3|_],[a],[1,3,B])$.
 (d) $Y \text{ is } 3+4, Z ::= Y+1$.
 (e) $X = 3*5, \backslash+ X = 15$.

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) $.(R,[1,4]) = [C,C,T|[]]$.
 (b) $f(h(B,A,1),[3,2|L]) = f(h(1,p+2,B),[K,2])$.

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

```
p([A|As],_,X):-
    p(As,A,X).
p([A,B|_],C,A):-
    2*A ::= B+C.
```

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 pontosvesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

- (a) $p([4,6],2,X)$.
 (b) $p([1,2,4,7],2,X)$.
 (c) $p([3,5,6],1,X)$.
 (d) $p([2,3,4,9,7,5,3],1,X)$.
 (e) $p([2.0,3.0,4.0,4.0,4.0,4.0,2.5,1.0],0.0,X)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

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

- (f) Írja le a p/2 eljárás jelentését deklaratív módon, azaz egészítse ki teljes kijelentő mondatra 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 az eredőjüket. Segédeljárást nem definiálhat! (8 pont)

```
% eredo_elem(+L, -A, -B, -E): Az L listában az A és B párok
% összevonhatók, és E az eredőjük.
% 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	real	: int -> real
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	Char.isAlpha	: char -> bool
rev	: 'a list -> 'a list	Char.isUpper	: 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) {a = #"a", b = (true, ord "c")} <> {b = (1 <> 0, ~1), a = 65}
- (b) (Char.isAlpha, 3*5.0, ~1) = (Char.isUpper, real 15, op-(0,2))
- (c) foldl map 0 [1,~2.0,3]

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

- (a) val x = List.filter Char.isUpper (rev (explode "aBcD"))
- (b) val x = 5*(if true andalso (1-2 > 1) orelse not(1-2 < 1) then 1 else 2)
- (c) val x = let val (a,b)=(1.0,(2.0,[3.0,4.0])) in a + #1(b) + hd(#2(b)) end

7. Tekintsük a következő függvénydefiníciókat! (8 pont)

```
fun kutya (a,b) =
  let fun macska a ys =
        if ( a > b )
        then rev ys
        else macska (10*a) (a::ys)
    in
      macska a []
    end
  fun eger (x,y) = foldl op+ 0 (kutya (x,y))
```

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

- (a) val xs = kutya(1,0)
 - (b) val xs = kutya(1,1)
 - (c) val xs = kutya(1,100)
 - (d) val xs = eger(1,100)
 - (e) val xs = eger(1,100000000)
8. Emelkedőnek nevezzük az olyan (x, y, z) hármasokat, amelyekre $x < y < z$. Írjon olyan függvényt emelkedok néven, amely egy $(int * int * int)$ list típusú listában 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) list -> int
   emelkedok xs = az xs-beli emelkedő hármasok száma *)
```

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

Deklaratív programozás, 1. zárthelyi, Budapest, 2007. április 18.

Munkaidő: 90 perc, összpontszám: 60

Prolog, „B” csoport (30 pont)

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) $U=3+3$, V is $U+1$, $W ::= 7$.
- (b) $[U|_] = [a, [b]]$.
- (c) $I = 4+1$, T is $7-2$, $I=T$.
- (d) $\text{append}(A, [_, B], [k, f, t])$.
- (e) $A*B=a*2*b*4$.

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

- (a) $[5, U|V] = .(X, .(X, []))$.
- (b) $r(s(3, C, B), [D|C], [B]) = r(A, [1, B], [4])$.

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

```
p([A|As], X, _) :-  
  p(As, X, A).  
p([A,B|_], A, C) :-  
  2*A ::= B+C.
```

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) $p([2, 3], X, 1)$.
- (b) $p([3, 5, 6], X, 1)$.
- (c) $p([1, 2, 4, 7], X, 1)$.
- (d) $p([4, 6, 8, 9, 7, 5, 3], X, 2)$.
- (e) $p([2.0, 2.5, 3.0, 3.0, 3.0, 3.0, 1.5, 0.0], X, 0.0)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

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

- (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 számpárokból álló listát. Ebben a listában két szomszédos elemet *rokonnak* hívunk, ha az első pár tagjainak összege megegyezik a második pár tagjainak összegével. Az ilyen párok X és Y tagjainak összegét *párösszegnek* nevezzük. Írjon olyan Prolog eljárást, amely felsorolja egy adott listában szereplő rokon elemeket és a párösszegüket. Segédeljárást nem definiálhat! (8 pont)

```
% rokonok(+L, -A, -B, -S): Az L listában az A és B számpárok  
% rokonok, és S a párösszegük.  
% 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
foldl	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	real	: int -> real
foldr	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	Char.isAlpha	: char -> bool
rev	: 'a list -> 'a list	Char.isLower	: 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) {b = (1 <> 0, ~1), a = 65} <> {a = #"a", b = (true, ord "c")}
- (b) (Char.isAlpha, ~1, 3*5.0) = (Char.isLower, op-(0,2), real 15)
- (c) foldr foldl 0 [1,~2.0,3]

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

- (a) val x = List.filter Char.isLower (rev (explode "aBcD"))
- (b) val x = 7*(if true andalso (1-2 > 1) orelse not(1-2 < 1) then 2 else 1)
- (c) val x = let val (b,a)=(2.0,[3.0,4.0]),1.0 in a + #1(b) + hd(#2(b)) end

7. Tekintsük a következő függvénydefiníciókat! (8 pont)

```
fun macska (a,b) =
  let fun eger a zs =
        if ( a > b )
          then rev zs
          else eger (a+1) (a::zs)
      in
        eger a []
      end
  fun kutya (x,y) = foldr op+ 0 (macska (x,y))
```

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

- (a) val xs = macska(1,1)
 - (b) val xs = macska(2,1)
 - (c) val xs = macska(1,5)
 - (d) val xs = kutya(1,5)
 - (e) val xs = kutya(123,125)
8. Süllyedőnek nevezzük az olyan (x, y, z) hármasokat, amelyekre $x > y > z$. Írjon olyan függvényt süllyedok néven, amely egy (int * int * int) list típusú listában található süllyedő hármasok számát adja eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkomentet. (8 pont)

```
(* süllyedok : (int * int * int) list -> int
   süllyedok zs = a zs-beli süllyedő hármasok száma *)
```

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