Deklaratív programozás, pótzárthelyi, 2002. május 14. 17.15–18.45
Munkaidő: 90 perc, összpontszám: 60
Prolog, „A” 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 valtozó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőnek. (5 pont)

   (a) `[X|Y] = [a,b].`

   (b) `V is 3*U, U = 2+2.`

   (c) `A is 2+2, B = A+2.`

   (d) `P/Q = 48/12/3.`

   (e) `1+4 = Z, Z = 5.`

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) `f(X, X, [1]) = f(B+A, 4/2+C, C).`

   (b) `[g(P,a*b),Q|R] = [Q|.(g(6,S),[S])].`

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

```
q(a, b).
q(b, a).

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

   Állapítsa meg, hogy a feltett kérdésekre válaszul a rendszer milyen behelyettesítést ad az `X` változónak! Írja fel az összes megoldást a rendszer által előállított sorrendben, pontosvesszővel elválasztva! Ha nincs megoldás, írjon `{no}`-t!

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

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

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

   (d) `p([b,u,b,a], X).`

   (e) `p([b,e,z,a,b,a,l], X).`

   Tegyük fel, hogy a fenti eljárás `p(L, X)` hívásában az `L` argumentum egy csak `a` és `b` atomokat tartalmazó lista. Írja le általánosan, hogy milyen `X` értékeket eredményez majd ez a hívás, és a megoldásokat milyen sorrendben adja ki a Prolog rendszer. (7 pont)

4. Egy hegyi utat egy `L` számlistával írunk le, amelynek elemei az út egyes pontjainak tengerszint feletti magasságát adják meg. Feltételezzük, hogy ezen pontok között az út vagy csak emelkedik vagy csak lejt. Az emelkedő szakaszok szintkülönbségének az összegét az út emelkedésének nevezzük.

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

```
% emelkedes(+L, +E0, -E): Az L lista által leírt út emelkedése E-E0.
% L és E0 bemenő, E kimenő paraméter.

| ?- emelkedes([10,15,20], 0, E).                ----> E = 10 ? ; no
| ?- emelkedes([0,10,5,15], 0, E).               ----> E = 20 ? ; no
| ?- emelkedes([0,20,40,25,20,40,28,16], 0, E).  ----> E = 60 ? ; no
```

Deklaratív programozás, pótzárthelyi, 2002. május 14. 17.15–18.45
Munkaidő: 90 perc, összpontszám: 60
Standard ML, „A” csoport (30 pont)

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
map          : ('a -> 'b) -> 'a list -> 'b list
op::         : 'a * 'a list -> 'a list
op@          : 'a list * 'a list -> 'a list
List.drop    : 'a list * int -> 'a list
```

```
explode  : string -> char list
implode  : char list -> string
ord      : char -> int
chr      : int -> char
```

5. Mi az `f` **típusa** az alábbi (egymástól független) deklarációkban? (9 pont)

   (a) `fun f x y z = (x, y z)`

   (b) `fun f (x, y, z) = x (y z)`

   (c) `fun f x (y, z) = x (y, z)`

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

   (a) `val x = explode "csi" @ [#"p",#"a"]`

   (b) `val x = List.filter (fn true => true | _ => false) [1<2, 1.2=2.1, "a"<>"b"]`

   (c) `val x = map List.drop [([9,8,7,6,5],4), ([3,2],1)]`

7. Mit kap eredményül, ha az alábbi `f` függvényt a nyolc számjegyből álló, ÉÉÉÉHHNN alakban füzérként megadott **saját** születési dátumára alkalmazza (pl. f "19820431")? Válaszát indokolja! (6 pont)

```
local
   fun g (x::y::zs) =
         chr (let val z = ord x + ord y - ord #"0"
              in if z <= ord #"9" then z else z-10
              end) :: g zs
     | g _ = []
in
   fun f t = implode(g(explode t))
end
```

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

   Írjon olyan SML-függvényt, amely megfelel az alábbi fejkommentnek! Segédfüggvényt nem definiálhat! (9 pont)

```
(* emelkedo : int list * int -> int
   emelkedo (ns, n) = m, ha ns (n,m)-emelkedő; ~1, ha nem az
   PRE: ns minden eleme >= 0, n >= 0
*)
```

Példák:
```
emelkedo([5,6], 4) = 6
emelkedo([6,5], 4) = ~1
emelkedo([0,1,2,0,3], 0) = 3
emelkedo([0,1,0,0,2,0,3,4,0], 0) = 4
emelkedo([0,0,0,0], 9) = 9
emelkedo([], 123) = 123
```

Deklaratív programozás, pótzárthelyi, 2002. május 14. 17.15–18.45
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 valtozó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőnek. (5 pont)

   (a) `[u,v,w] = [X,Y|Z].`

   (b) `4+2+3 = X+Y.`

   (c) `X is X+1.`

   (d) `X is 3+4, X = 3+4.`

   (e) `a+A = B+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], Y, 2*X] = [.(1,E), F, F].`

   (b) `f(U*V, [3*2+b|W]) = f(S, [S+T, T]).`

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

```
s([], 0).
s([X|_], X) :- X > 5.
s([_,_|L], X) :- s(L, X).
```

   Állapítsa meg, hogy a feltett kérdésekre válaszul a rendszer milyen behelyettesítést ad az `X` változónak! Írja fel az összes megoldást a rendszer által előállított sorrendben, pontosvesszővel elválasztva! Ha nincs megoldás, írjon `{no}`-t!

```
| ?- s([8], X).
| ?- s([7,8], X).
| ?- s([7,8,9], X).
| ?- s([1,6,8,2,7,9,3], X).
| ?- s([2,3,4,5,6,7,8,9], X).
```

   Tegyük fel, hogy a fenti eljárás `s(L, X)` hívásában `L` egy páros elemszámú lista. Írja le általánosan, hogy ebben az esetben milyen `X` értékeket eredményez majd ez a hívás, és a megoldásokat milyen sorrendben adja ki a Prolog rendszer. (7 pont)

4. Egy repülő útvonalát egy `L` számlistával írjuk le, amelynek elemei a repülő rendszeres időközökben mért tengerszint feletti magasságát adják meg, méterben. Zuhanásnak nevezzük azt, ha két egymás utáni adat közötti különbség legalább 10 méteres süllyedést mutat.

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

```
% zuhanasok(+L, +Z0, -Z): Az L lista által leírt repülő-útvonalon Z-Z0
% számú esetben fordul elő zuhanás. L és Z0 bemenő, Z kimenő paraméter.

| ?- zuhanasok([20,15,10], 0, Z).              ----> Z = 0 ? ; no
| ?- zuhanasok([20,10], 0, Z).                 ----> Z = 1 ? ; no
| ?- zuhanasok([0,20,40,25,20,40,28,16], 0, Z). ----> Z = 3 ? ; no
```

Deklaratív programozás, pótzárthelyi, 2002. május 14. 17.15–18.45
Munkaidő: 90 perc, összpontszám: 60
Standard ML, „B” csoport (30 pont)

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
map          : ('a -> 'b) -> 'a list -> 'b list
op::         : 'a * 'a list -> 'a list
op@          : 'a list * 'a list -> 'a list
List.take    : 'a list * int -> 'a list
```

```
explode  : string -> char list
implode  : char list -> string
ord      : char -> int
chr      : int -> char
```

5. Mi az `f` **típusa** az alábbi (egymástól független) deklarációkban? (9 pont)

   (a) `fun f x y z = (x y, z)`

   (b) `fun f (x, y, z) = (x y) z`

   (c) `fun f (x, y) z = x (y, z)`

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

   (a) `val x = [#"p",#"a"] @ explode "csi"`

   (b)
   ```
   val x = List.filter (fn true => false | _ => true)
                                         [3.0>4.0, 4.7>2.4, ~3.0<=0.4]
   ```

   (c) `val x = map List.take [([9,8,7,6,5],4), ([3,2],1)]`

7. Mit kap eredményül, ha az alábbi `f` függvényt a nyolc számjegyből álló, ÉÉÉÉHHNN alakban füzérként megadott **saját** születési dátumára alkalmazza (pl. f "19820431")? Válaszát indokolja! (6 pont)

```
local
   fun h (u::v::ws) =
         chr (let val w = (ord u + ord v) div 2
              in if w >= ord #"5" then w else 2*w - ord #"0"
              end) :: h ws
     | h _ = []
in
   fun f t = implode(h(explode t))
end
```

8. Egy nemnegatív egész számokból álló listát (n, m)-lejtőnek hívunk, ha a 0 elemektől eltekintve kizárólag az $n, n-1, \ldots, m+1$ (esetleg üres) számsorozatot tartalmazza, ebben a sorrendben. Speciálisan: egy üres vagy csak 0-kat tartalmazó listát (n, n)-lejtőnek tekintünk, tetszőleges n esetén.

   Írjon olyan SML-függvényt, amely megfelel az alábbi fejkommentnek! Segédfüggvényt nem definiálhat! (9 pont)

```
(* lejto : int list * int -> int
   lejto (ns, n) = m, ha ns (n,m)-lejtő; ~1, ha nem az
   PRE: ns minden eleme >= 0, n > 0
*)
```

Példák:
```
lejto([6,5], 6) = 4
lejto([5,6], 5) = ~1
lejto([0,3,2,0,1], 3) = 0
lejto([0,5,4,0,0,3,0,2,0], 5) = 1
lejto([0,0,0,0], 9) = 9
lejto([], 123) = 123
```