

A Prolog eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Segédeljárást nem definiálhat.

Az egyes részfeladatokra azok szám- és betűjelével (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)  $X = 1+2$ ,  $X = 3$ .

(c)  $X = 4$ ,  $\backslash+ 2+2 = X$ .

(d)  $3+4$  is  $X$ .

(e)  $X*Y = 2*2*3$ .

2. Írja fel az alábbi kifejezések alapstruktúra-alakját (azaz szintaktikus édesítőszerek nélküli formáját), vagy rajzolja fel a hozzájuk tartozó fastruktúrákat! (6 pont)

(a)  $3+6-z/x$

(b)  $[f(3,4), X|L]$

(c)  $[g(1,2,(a-b)*3), []]$

3. Milyen változó-behelyettesítéseket eredményeznek az alábbi egyesítések? (9 pont)

(a)  $f(a*b*c, [1,2]) = f(A*B, [C|D])$ .

(b)  $[.(1,E), F, F] = [[X], Y*Z, X*2]$ .

(c)  $f(G/H, [3/G-b|J]) = f(P, [P-Q, Q])$ .

4. Tekintse a `p/1` eljárást és az azt hívó alábbi célsorozatot!

```
p(B) :-
    member(A, B),
    ( A =< 0'6 -> put_code(A)
    ; put_code(0'.)
    ),
    fail.
p(_).

| ?- write(ÉÉHHNN), nl, p("ÉÉHHNN"), nl.
```

Mit ír ki a fenti célsorozat, ha abban az `ÉÉHHNN` karaktersorozat helyén az ön születési dátuma (pl. 811231) szerepel? Fogalmazza meg általánosan, hogy a `p/1` által kiírt szöveg hogyan függ a paraméterfüzérben szereplő dátumtól! (Futtatáskor a `lists` könyvtár be van töltve.) (5 pont)

5. Írjon olyan Prolog eljárást, amely megfelel az alábbi fejkommentnek! (5 pont)

```
% kozep(+L, -A): A egy olyan eleme az L listának, amely a bal- és
% jobb-szomszédjának számtani közepe.
```

```
| ?- kozep([-1,0,1,2,3,5,7,8], K).
K = 0 ? ; K = 1 ? ; K = 2 ? ; K = 5 ?
```

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Segédfüggvényt nem definiálhat.

Az egyes részfeladatokra azok szám- és betűjelével (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa:

```
explode      : string -> char list
List.filter  : ('a -> bool) -> 'a list -> 'a list
foldl       : ('a * 'b -> 'b) -> 'b -> 'a list -> 'b
implode     : char list -> string
map         : ('a -> 'b) -> 'a list -> 'b list
op+        : int * int -> int
ord         : char -> int
rev        : 'a list -> 'a list
```

6. Mi az `f` típusa az alábbi (egymástól független) deklarációkban? (8 pont)

- (a) `fun f x y z = x z y`
- (b) `fun f (x, y) = y (x+1, 2.0)`
- (c) `fun f x y = y (x y)`

7. Mi az `x` értéke és típusa az alábbi (egymástól független) deklarációkban? (3 pont)

- (a) `val x = implode(["a"] @ ["b", "c"])`
- (b) `val x = List.filter (fn x => x > 1.5) [1.0, 2.0, 3.0]`
- (c) `val x = map (fn i => i+1) [1,2,3,4]`

8. Mit kap eredményül, ha az alábbi `f` függvényt a hat számjegyből álló, (ééhhnn alakban füzéreként megadott születési dátumára alkalmazza (pl. `f "811231"`)? Válaszát indokolja! (6 pont)

```
fun f x =
  let
    fun g (x::xs) ys = x :: g ys xs
      | g _ _ = []
    val y = explode x
  in
    implode (g y (rev y))
  end
```

9. Írja meg az alábbi `f` függvény egy ekvivalens változatát a `foldl`, `map`, `op+` és `ord` függvények alkalmazásával! (6 pont)

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

10. Írjon olyan SML-függvényt `front` néven, amely az alább deklarált adattípus adatkonstruktoraival létrehozott adatszerkezet „elejét” listává alakítja az adatok sorrendjének megőrzésével! Az adatszerkezet „eleje”: a `<<` adatkonstruktorral az adatszerkezetbe illesztett értékek sorozata. (7 pont)

```
infixr 5 <<; infix 5 >>;
datatype 'a que = Nil | << of 'a * 'a que | >> of 'a que * 'a
```

Példák:

```
front Nil = [];
front("b" << Nil) = ["b"];
front(Nil >> "b") = [];
front((1 << 2 << ((3 << Nil) >> 4 >> 5)) >> 6) = [1, 2, 3];
```

A Prolog eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Segédeljárást nem definiálhat.

Az egyes részfeladatokra azok szám- és betűjelével (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 is 3+4, X = 3+4.`

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

(c) `X is X+1.`

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

(e) `X = a, \+ a = X.`

2. Írja fel az alábbi kifejezések alapstruktúra-alakját (azaz szintaktikus édesítőszerek nélküli formáját), vagy rajzolja fel a hozzájuk tartozó fastruktúrákat! (6 pont)

(a) `1*2*3/x`

(b) `h(3,4,[1,X|L])`

(c) `[u/(1-a),[],h(1,2)]`

3. Milyen változó-behelyettesítéseket eredményeznek az alábbi egyesítések? (9 pont)

(a) `[A+B, C|D] = [1+2+3, 4+5].`

(b) `[E+b*c, F+G|E] = .(X, [X, 1]).`

(c) `h([H, G], H*G) = h([Q/1|R], P/Q*3).`

4. Tekintse a `p/1` eljárást és az azt hívó alábbi célsorozatot!

```
p([]).
```

```
p([A|B]) :-
```

```
    member(A, "56789"), !,
```

```
    p(B),
```

```
    put_code(A).
```

```
p([_ | B]) :-
```

```
    p(B),
```

```
    put_code(0'x).
```

```
| ?- write(ÉÉHHNN), nl, p("ÉÉHHNN"), nl.
```

Mit ír ki a fenti célsorozat, ha abban az `ÉÉHHNN` karaktersorozat helyén az ön születési dátuma (pl. 811231) szerepel? Fogalmazza meg általánosan, hogy a `p/1` által kiírt szöveg hogyan függ a paraméterfüzérben szereplő dátumtól! (Futtatáskor a `lists` könyvtár be van töltve.) (5 pont)

5. Írjon olyan Prolog eljárást, amely megfelel az alábbi fejkomentnek! (5 pont)

```
% nullpar(+L, -NL): NL egy olyan kételemű (folytonos) részlistája L-nek,  
% amelynek összege 0.
```

```
| ?- nullpar([0,1,-1,-2,2,-2,-1], NP).
```

```
NP = [1,-1] ? ; NP = [-2,2] ? ; NP = [2,-2] ? ; no
```

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Segédfüggvényt nem definiálhat.

Az egyes részfeladatokra azok szám- és betűjelével (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa:

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

6. Mi az `f` típusa az alábbi (egymástól független) deklarációkban? (8 pont)

- (a) `fun f x y z = z y x`
- (b) `fun f (x, y) = x(1.0, y div 2)`
- (c) `fun f x y = (y, (x y))`

7. Mi az `x` értéke és típusa az alábbi (egymástól független) deklarációkban? (3 pont)

- (a) `val x = implode("#z" :: #y" :: [#x])`
- (b) `val x = List.filter (fn x => x <= 2) [1, 2, 3]`
- (c) `val x = map (fn i => i div 2) [3,4,6,9]`

8. Mit kap eredményül, ha az alábbi `f` függvényt a hat számjegyből álló, (ééhhnn alakban füzéreként megadott születési dátumára alkalmazza (pl. `f "801128"`)? Válaszát indokolja! (6 pont)

```
fun f x =
  let
    fun g (x::xs) ys = str x ^ g ys xs
      | g _ _ = ""
    val y = explode x
  in
    g (rev y) y
  end
```

9. Írja meg az alábbi `f` függvény egy ekvivalens változatát a `foldr`, `map`, `op^` és `str` függvények alkalmazásával! (6 pont)

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

10. Írjon olyan SML-függvényt `revRear` néven, amely az alább deklarált adattípus adatkonstruktoraival létrehozott adatszerkezet „végét” listává alakítja az adatok sorrendjének megfordításával! Az adatszerkezet „vége”: `a >>` adatkonstruktorral az adatszerkezetbe illesztett értékek sorozata. (7 pont)

```
infixr 5 <<; infix 5 >>;
datatype 'a que = Nil | << of 'a * 'a que | >> of 'a que * 'a
```

Példák:

```
revRear Nil = [];
revRear("b" << Nil) = [];
revRear(Nil >> "b") = ["b"];
revRear((1 << 2 << ((3 << Nil) >> 4 >> 5)) >> 6) = [6, 5, 4];
```