

# Deklaratív programozás feladatsor

Nagycsoport-2, 1. gyakorlati feladat és kérdéssor

## Prolog feladatok

### Bemelegítő feladatok – ZH jellegű feladatok

#### Kifejezés kiértékelés

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.

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

#### Alapstruktúra alakra hozás

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

- (a) [A,b\*c|D] = [2\*3,R,p].
- (b) [[A+C|B],f(A,B,C)] = [[1\*2+3,p,q],Z].

#### Nyomkövetés

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

```
p([K-V, _-Y|_], B, K-V) :-  
    abs(V-Y) =< B.  
p([_|L], B, Z) :-  
    p(L, B, 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([1-5], 3, X).
- (b) | ?- p([1-5,3-4], 1, X).
- (c) | ?- p([3-3,1-1,2-2], 4, X).
- (d) | ?- p([5-6,1-9,6-9,e-10], 2, X).
- (e) | ?- p([2-2,w-6,3-3,t-2,7-2,3-8,5-1,8-3], 2, X).

Tekintse a fenti p/3 eljárásra épülő alábbi p/2 eljárást, és egészítse ki teljes kijelentő mondattá a fejkommentet!

```
% p(L, Z): Az X-Y alakú párokból álló L listának Z egy olyan eleme,....  
p(L, Z) :- p(L, 0, Z).
```

## Rövid kérdések

Milyen jellegű programozási feladat megoldásához választaná a Prolog nyelvet?

Van-e, és ha igen, mi a különbség az alábbiak között Prologban?

`2*3`

`6`

Írjon az alábbi specifikációkra illeszkedő mintákat:

- egyelemű lista
- legalább kételemű lista
- X-et keressük
  - `functor(X, a, 3) .`
  - `functor(X, b, 4), arg(2, X, 7) .`
  - `functor(X, c, 2), arg(1, X, Y), functor(Y, ., 2) .`

Készítsen egyszerű LUT-ot (Look-up table) Prologban, amiből az egyes osztályzatokhoz tartozó megnevezés nyerhető ki (pl. 4 --> jó).

Tervezze meg Prologban a DP gyakorlatok csoportbeosztásának nyilvántartását! Tartsa nyilván az alábbiakat (tfh, hogy minden konzulensnek csak 1 „kiscsoportja” van, így a csoport a konzulens nevével azonosítható, továbbá minden csoportfelelősnek csak 1 „nagy csoportja” van):

- valamely hallgatónak ki a konzulense
- valamely „kiscsoportnak” ki a csoportfelelőse

Az előző nyilvántartáshoz írjon Prolog eljárásokat az alábbi kapcsolatok felderítésére:

- „A” hallgató csoportjának csoportfelelőse „B”
- „A” és „B” hallgató csoport társak
- „A” konzulenshez tartozik „B” hallgató. A konzulens ebben az esetben egyaránt lehet csoportfelelős vagy kiscsoport konzulense, így a >>hozzá tartozik<< lehet közvetett vagy közvetlen kapcsolat is!

## Kapcsolódó kisházik

Binomiális együtthatók meghatározása, <http://dp.iit.bme.hu/dp06a/dp06a-pl1-khf.html>

Ciklus szerinti forgatások, <http://dp.iit.bme.hu/dp06s/dp06s-pl2-khf.html>

## SML feladatok

### Bemelegítő feladatok – ZH jellegű feladatok

### Hibakeresés

Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két hiba van. Mik ezek?

(a) `[1*2, round 7.3, 3*7<>7*3.0]`

(b) `(#"a", 3=2, 12) = (ord 97, true, 5+7)`

(c) `map (fn (x,y) => x+y) [3.0, 4.0, real 5.0]`

## Kiértékelés eredménye

Mi a k értéke az alábbi, egymástól független deklarációk kiértékelése után?

- (a) `val (_::_:~:~:k::~_) = tl(tl(explode "vAlToGaTvA"))`  
(b) `val k = map (fn x => x mod 3) [1,7,3,5]`  
(c) `val (_::_:~:~:k) = List.filter Char.isUpper (explode "vAlToGaTvA")`

## Nyomkövetés

Nézzük a következő függvények definícióját!

```
fun comb (x::xs, y::ys) = (x, y) :: comb(ys, xs) | comb _ = []  
fun f zs = map (fn (a,b) => a+b) (comb(zs, tl zs))
```

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

- (a1) `val x = f [1,2,3,4,5]`  
(a2) `val x = f [~1]`  
(a3) `val x = f [~1,1]`  
(a4) `val x = f []`

Mutassa be a `comb([1,2,3,4], [2,3,4])` függvényalkalmazás egyszerűsítési lépéseit, mohó kiértékelést feltételezve!

## Rövid kérdések

Mennyi `ord("#"L)-ord("#"E)`? Kell-e ehhez ismerni az egyes alfabetikus karakterek kódját?

Van-e, és ha igen, mi a különbség az alábbiak között SML-ben?

```
2*3  
6
```

Írjon az alábbi specifikációkra illeszkedő mintákat:

- egyelemű lista
- legalább kételemű lista

Készítsen egyszerű LUT-ot (Look-up table) SML-ben, amiből az egyes osztályzatokhoz tartozó megnevezés nyerhető ki (pl. 5 --> „jeles”).

Írjon egy egész számlista elemeinek összegét meghatározó függvényt. Az üres számlista elemeinek összege legyen definíció szerint 0.

Deklaráljon kivételt Alma néven, amibe az alma „betegségét” szövegesen bele tudja csomagolni.

Tegyük fel, hogy **Kif1** egy olyan kifejezés, ami nemnegatív egész számot ad vissza (pl. valamely alma eladási ára), és az előző pontban deklarált kivétel egy példányát dobja, ha valami baj van az almával. Írjon **Kif1** „köré” olyan kivételkezelő blokkot, ami az egyes kivételekhez negatív egész számot rendel (pl. „kukacos” --> -1, „fonnyadt” --> -2, egyéb esetekben -3).

Az interaktív interpreteres környezetben teszteléshez használhatja **Kif1** helyett az alábbi:

```
(if 2=3 then 0 else raise Alma "kukacos")
```

## Kapcsolódó kisházik

Számábrázolás monoton számjegyekkel, <http://dp.iit.bme.hu/dp06a/dp06a-ml1-khf.html>

Altábla kinyerése, <http://dp.iit.bme.hu/dp06a/dp06a-ml2-khf.html>