1. Inkrementálás (PL)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
inc(+L,?I) pontosan akkor igaz, ha L és I egyforma hosszú, számokat tartalmazó
    listák, és I minden eleme eggyel nagyobb L azonos pozíciójú eleménél.
```

Pl:

```
| ?- inc([1,2,5.6],I).
I = [2,3,6.6] ? ;
no
```

2. Jobbrekurzív faktoriális (PL)

Egy szám faktoriálisát jobbrekurzív módon dinamikus programozással lehet meghatározni. Azaz n! = 1*2*3*...*(n-1)*n alapján ha egy k < n-re már ismert k! értéke, akkor n! = k!*(k+1)*(k+2)*...*(n-1)*n. k = 0 esetén pedig k! = 1, így innen lesz indítható a jobbrekurzív számítás.

Írjon az alábbi fejkommentnek megfelelõ programot!

```
fakt(+N0,+F0,+N,?F): ha N0 faktoriálisa F0, N faktoriálisa F.
```

PRE:

```
  0 <= N0 < N, N0 és N egészek
```

Pl:

```
| ?- fakt(5,120,7,F).
F = 5040 ? ;
no
```

Pl:

```
| ?- fakt(5,3,7,F).
F = 126 ? ;
no
```

3. Összegzés (PL)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
osszegek(+L,?O) pontosan akkor igaz, ha O az L lista szomszédos elempárjainak
    összegeit tartalmazó lista.
```

Pl:

```
| ?- osszegek([1,2,3,4],O).
O = [3,5,7] ? ;
no
```

Pl:

```
| ?- osszegek([4,2.5,6,1,2.14],O).
O = [6.5,8.5,7,3.14] ? ;
no
```

4. Eltolás (PL)

Egy C tömör listát ciklusnak nevezünk, ha nincs két azonos (egyesíthetõ) eleme. Például [z,e,n] és az üres lista ciklusok, de [z,e,n,e] nem az, mert az e elem ismétlõdik, és [z,e,n,E] sem az, ha az E behelyettesítetlen változó.

Egy X tömör érték C ciklus szerinti eltoltjának azt az értéket nevezzük, amely a C listában közvetlenül az X után áll, vagy ha nincs ilyen elem, akkor magát az X-et.

Írjon az alábbi fejkommentnek megfelelõ programot!

```
eltoltja(+X,+C,?Y) pontosan akkor igaz, ha az X tömör érték
    C ciklus szerinti eltoltja Y
```

PRE:

```
  X és C tömör
```

Pl:

```
| ?- eltoltja(e,[z,e,n], Y).
Y = n ? ;
no
```

Pl:

```
| ?- eltoltja(e,[t,r,u,e], Y).
Y = e ? ;
no
```

Pl:

```
| ?- eltoltja(e,[a,b,c], Y).
Y = e ? ;
no
```

5. Forgatás (PL)

Egy X tömör érték C ciklus szerinti forgatottjának azt az értéket nevezzük, amely a C listában közvetlenül az X után áll, vagy ha az X a C utolsó eleme, akkor a C elsõ elemét, vagy ha az X nem szerepel C-ben, akkor magát az X-et. Például az e forgatottja [t,r,u,e] szerint t, [z,e,n] szerint viszont n.

Írjon az alábbi fejkommentnek megfelelõ programot!

```
 forgatottja(+X,+C,?Y) pontosan akkor igaz, ha  az X tömör érték
        C ciklus szerinti forgatottja Y
```

PRE:

```
   X és C tömör
```

Pl:

```
| ?- forgatottja(e,[z,e,n], Y).
Y = n ? ;
no
```

Pl:

```
| ?- forgatottja(e,[t,r,u,e], Y).
Y = t ? ;
no
```

6. KHF1: Binomiális együtthatók (PL)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
 binoms(+N,?L): Az L lista az N-edrendû binomiális együtthatókat tartalmazza,
 azaz N+1 elemû és (0-tól számozva) I-edik elemének értéke "N alatt az I".
```

Pl:

```
| ?- binoms(4, L).
L = [1,4,6,4,1] ? ;
no
```

7. KHF2: Körbeforgatás (PL)

Egy L tömör lista C ciklus szerinti körbeforgatottjának azt a listát nevezzük, amely az L elemeinek a C szerinti forgatottjaiból áll (az eredeti sorrendben).

Írjon az alábbi fejkommentnek megfelelõ programot!

```
 korbe(+L,+C,?M) pontosan akkor igaz, ha  az L lista
     C ciklus szerinti körbeforgatottja az M lista.
```

PRE:

```
   L és C tömör
```

Pl:

```
| ?- korbe([t,r,u,e],[z,e,n], M).
M = [t,r,u,n] ? ;
no
```

Pl:

```
| ?- korbe([z,e,n],[t,r,u,e], M).
M = [z,t,n] ? ;
no
```

Pl:

```
| ?- korbe([p,r,o,1,2,3],[p,1,r,2], M).
M = [1,2,o,r,p,3] ? ;
no
```

8. Lista darabolás (SML)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
 cut : int * int -> 'a list -> 'a list
 cut (i,j) l = az l lista i. elemétõl kezdõdõ, j hosszú részlistája
     (az l lista feje az 1. eleme). Ha l (i+j-1)-nél kevesebb elemet
         tartalmaz, akkor a függvény ezt egy tetszõleges kivétellel jelezze!
```

Pl:

```
cut (2,3) [4,0,1,5,7] = [0,1,5]
```

9. Monoton sorozatok (SML)

Monotonnak nevezünk egy egész számokból álló sorozatot, amelyben ha az elsõ elem nem kisebb a másodiknál, akkor a második sem kisebb a harmadiknál, a harmadik sem a negyediknél s.í.t., ha pedig az elsõ elem nem nagyobb a másodiknál, akkor a második sem nagyobb a harmadiknál, a harmadik sem a negyediknél s.í.t.

Írjon az alábbi fejkommentnek megfelelõ programot!

```
 monotonsorozat : int list -> bool
 monotonsorozat l = igaz, ha l monoton
```

Pl:

```
monotonsorozat [4,4,5,8,11] = true
```

Pl:

```
monotonsorozat [14,7,2] = true
```

Pl:

```
monotonsorozat [0,3,3,8] = true
```

Pl:

```
monotonsorozat [2,6,4,33] = false
```

10. Számjegyek (SML)

Egy szám átváltásánál adott számrendszerbe az alábbi séma szerint lehet eljárni: az egyes helyiérték meghatározása a magasabb helyiértékeken ábrázolandó maradék meghatározása ha a maradék nem 0, a maradékot rekurzívan át kell váltani

Írjon az alábbi fejkommentnek megfelelõ programot!

```
szamjegy : int -> int -> int * int
szamjegy n x = (egyes, magasabb), ahol egyes az x szám n-es számrendszerben
               vett alakjának egyes helyiértéke, magasabb pedig a magasabb
               helyiértéken ábrázolandó része
```

PRE:

```
n > 1
```

PI:

```
szamjegy 2 11 = (1, 5)
```

PI:

```
szajmegy 2 5 = (1, 2)
```

PI:

```
szamjegy 2 2 = (0, 1)
```

PI:

```
szamjegy 2 1 = (1, 0)
```

11. Fordított számrendszerváltás (SML)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
szamrendszer : int -> int -> int list
szamrendszer n x = az x szám n-es számrendszerben vett jegyeinek listája,
                   növekvõ sorrendben, egy lezáró 0-val.
```

PRE:

```
n > 1
```

PI:

```
szamrendszer 2 11 = [1, 1, 0, 1, 0]
```

PI:

```
szamrendszer 15 201 = [6, 13, 0]
```

12. Számrendszerváltás (SML)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
szamrendszer : int -> int -> int list
szamrendszer n x = az x szám n-es számrendszerben vett jegyeinek listája.
```

PRE:

```
n > 1
```

PI:

```
szamrendszer 2 11 = [1, 0, 1, 1]
```

PI:

```
szamrendszer 15 201 = [13, 6]
```

13. KHF1: Monoton számrendszer (SML)

Írjon az alábbi fejkommentnek megfelelõ programot!

```
monoton : int -> int
monoton a = b, ahol 'b' >= 2 a legkisebb olyan természetes szám, amelyre a
'b' alapú számrendszerben felírt 'a' szám monoton számsorozatot ad eredményül
```

PI:

```
monoton 4 = 2
```

PI:

```
monoton 11 = 4
```

PI:

```
monoton 145 = 8
```

PI:

```
monoton 293 = 7
```

14. KHF2: Altábla (SML)

Adott egy m oszlopból és n sorból álló, téglalap alakú tábla, amelynek a mezõire az (x,y) koordinátájukkal hivatkozunk (1<=x<=n, 1<=y<=m). E tábla altáblájának nevezzük azt a j oszlopból és i sorból álló részét (1<=i<=n, 1<=j<=m), amelyet a bal felsõ mezõjének - ún. referenciapontjának - a koordinátája azonosít. A tábla és az altáblák bal felsõ mezõjének (relatív) koordinátája (1,1).

Ha egy i*j méretû altábla referenciapontjának a koordinátája (x,y), akkor teljesülniük kell a következõ egyenlõtlenségeknek: 1<=x+i-1<=n, 1<=y+j-1<=m.

Írjon az alábbi fejkommentnek megfelelõ programot!

```
altabla : 'a list list -> (int * int) -> (int * int) -> 'a list list
altabla tss (x,y) (i,j) = a 'tss' táblának az az 'i'*'j' méretû altáblája,
    amelynek referenciapontja a 'tss' tábla '(x,y)' koordinátájú mezõje;
    ha a táblának nincs '(x,y)' referenciapontú, 'i'*'j' méretû altáblája,
    ezt a függvény az 'Altabla' kivétellel jelezze
```

PI:

```
val tss = [["a","b","c","d","e","f"],
           ["0","1","2","3","4","5"],
           ["N","M","O","P","Q","R"],
           [".",",",";",":","<",">"],
```

```
          ["q","w","e","r","t","y"]];
  altabla tss (2,3) (3,2);
> val it = [["2","3"],["0","P"],[";",":"]] : string list list
PI:
  altabla tss (2,3) (5,2);
! Uncaught exception:
! Altabla
PI:
  altabla tss (6,3) (3,2);
! Uncaught exception:
! Altabla
```