

Deklaratív programozás, mintazárthelyi, 2008. november 3.
Munkaidő: 90 perc, összpontszám: 60
Erlang (30 pont)

Az Erlang függvények megírását kérő feladatokban a jegyzetben szereplő összes Erlang függvény (akár beépített, akár a jegyzetben definiált) szabadon használható, a modulneveket nem kötelező kiírni (pl. lists:map helyett írható egyszerűen map). Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények argumentumai és visszatérési értékei:

concat(ListakListaJa) -> Lista	map(Fun, ListaBe) -> ListaKi
foldl(Fun, KezdetiErtek, Lista) -> Eredmeny	seq(KezdetiErtek, VegErtek) -> Lista
filter(Fun, ListaBe) -> ListaKi	length(Lista) -> Hossz
append(Lista1, Lista2) -> Lista12	reverse(Lista) -> FordLista

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

- (a) `X = concat(filter(fun([_|_]) -> true; (_) -> false end, [2,[3],[6,1],[5]]))`.
- (b) `X = 5 * (case {a,4,6} of {A,B} -> B; C -> {_,D,E} = C, E; {a,F,G} -> F end)`.
- (c) `X = [{X, Y} || X <- seq(1, 10), Y <- seq(1, 10), X * X == Y]`.

6. Nézzük a következő függvények definícióját! (9 pont)

<code>g([X, Y Z]) -></code> <code>if</code> <code>X < Y -></code> <code>[X g([Y Z])];</code> <code>true -></code> <code>[Y g([X Z])]</code> <code>end;</code> <code>g(Z) -></code> <code>Z.</code>	<code>h(Z) -></code> <code>h(Z, length(Z) - 1).</code> <code>h(Z, 0) -></code> <code>Z;</code> <code>h(Z, C) -></code> <code>h(g(Z), C - 1).</code>
---	---

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

- (a) `X = g([])`.
- (b) `X = g("a")`.
- (c) `X = g("alma")`.
- (d) `X = h("alma")`.
- (e) `X = h("korte")`.

7. Készítsen az alábbival megegyező funkciójú **jobbrekurzív** függvényt, és mutasson egy példát a hívására! (7 pont)

```
%% rev([any()]) -> [any()]
rev([A | Z]) ->
    rev(Z) ++ [A];
rev([]) ->
    [].
```

Hívása: `rev("alma")`.

8. Emelkedőnek nevezzük az olyan $\{X, Y, Z\}$ hármesokat, amelyekre $X < Y < Z$. Írjon olyan függvényt emelkedők néven, amely egy `{integer(), integer(), integer()}` típusú listában található emelkedő hármesok számát adja eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet. (8 pont)

```
%% emelkedok([integer(), integer(), integer()]) -> integer()
%% A bemeneti listában szereplő emelkedő hármesok 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.`