

```

1 -module('dp19a-gy2').
2 -compile(export_all).
3 -author('patai@iit.bme.hu, hanak@iit.bme.hu, kapolnai@iit.bme.hu').
4 -vsn('$LastChangedDate: 2019-11-05 09:17:30 +0100 (k, 05 nov 2019) $$').
5
6 %-----
7 %                               LISTA
8 %-----
9
10 % 1.
11 -spec seq(F::integer(), T::integer()) -> S::[integer()].
12 %% S = [F, F+1, ..., T].
13 seq(F, T) ->
14     seq(F, T, []).
15
16 -spec seq(F::integer(), T::integer(), L::[integer()]) -> S::[integer()].
17 %% seq(F, T, L) = F..T sorozat L elé füzve.
18 seq(F, T, L) when T<F -> L;
19 seq(F, T, L) ->
20     seq(F, T-1, [T|L]).
21
22 %-----
23
24 % 2.
25 -spec all_different(Xs::[any()]) -> B::boolean().
26 %% B igaz, ha az L listában csupa különböző értékű elem van.
27 all_different([]) ->
28     true;
29 all_different([H|T]) ->
30     not lists:member(H, T) andalso all_different(T).
31
32 -spec all_different2(Xs::[any()]) -> B::boolean().
33 all_different2(L) -> length(L) == length(lists:usort(L)).
34
35 %-----
36
37 % 3.
38 -spec van2eleme(L::list()) -> R::boolean().
39 %% R == length(L) >= 2.
40 van2eleme([_,_|_]) -> true;
41 van2eleme(_) -> false.
42
43 %-----
44
45 % 4.
46 -spec sublist(L0::[any()], S::integer(), N::integer()) -> L::[any()].
47 %% Az L0 lista S-edik elemétől kezdődő és N hosszú részlistája az L lista.
48 sublist(L, S, N) ->
49     [lists:nth(I, L) || I <- lists:seq(S, S+N-1)].
50
51 %-----
52
53 % 5.
54 -spec all(P::fun((T::any()) -> boolean()), L::[T::any()]) -> B::boolean().
55 all(Pred, L) ->
56     lists:foldl(fun(X,E) -> X andalso E end, true, lists:map(Pred, L)).
57
58 -spec all2(P::fun((T::any()) -> boolean()), L::[T::any()]) -> B::boolean().
59 all2(Pred, L) ->
60     lists:foldl(fun(X,E) -> Pred(X) andalso E end, true, L).
61
62 -spec all3(P::fun((T::any()) -> boolean()), L::[T::any()]) -> B::boolean().
63 %% B akkor és csak akkor igaz, ha Pred teljesül az L minden elemére.
64 all3(Pred, [Hd|Tail]) ->
65     case Pred(Hd) of
66     true -> all3(Pred, Tail);
67     false -> false
68     end;
69 all3(Pred, []) when is_function(Pred, 1) ->
70     true.

```

```

71
72 -spec all4(P::fun((T::any()) -> boolean()), L::[T::any()]) -> B::boolean().
73 all4(_Pred, []) ->
74     true;
75 all4(Pred, [H|T]) ->
76     Pred(H) andalso all4(Pred, T).
77
78 %-----
79
80 % 6.
81 -spec platok_hossza(Xs::[any()]) -> PHs::[{P::any(), H::integer()}].
82 %% Platónák nevezzük az egyenlő értékű elemek sorozatát egy listában.
83 %% PHs olyan {P, H} párok listája, amelyekben P a platót képező
84 %% érték, H pedig e plató hossza (= azonos értékű elemeinek száma).
85 platok_hossza([]) ->
86     [];
87 platok_hossza(Xs) ->
88     {[P|_Ps] = PPs, Ss} = plato(Xs),
89     [ {P, length(PPs)} | plato_hossza(Ss) ].
90
91 -spec plato(Xs::[any()]) -> PsMs::{Ps::[any()], Ms::[any()]}.
92 %% PsMs egy olyan {Ps, Ms} pár, amelyben Ps az Xs lista azonos értékű
93 %% elemekből álló, tovább már nem bővíthető prefixuma, Ms pedig az Xs
94 %% további elemeinek listája.
95 plato([X|[X|_Ys] = XYs]) ->
96     {Ps, Ms} = plato(XYs),
97     {[X|Ps], Ms};
98 plato([X|[_Y|_Ys] = YYs]) ->
99     {X}, YYs;
100 plato(Xs) -> % itt már csak egy elemű vagy üres lehet az Xs lista
101     {Xs, []}.
102
103 %-----
104
105 % 7.
106 -spec duplak(Xs::[any()]) -> Ys::[any()].
107 %% Ys az Xs azon elemeinek listája, melyek azonosak az őket követő elemmel.
108 duplak([]) -> [];
109 duplak(L) ->
110     [ E || {E,E} <- lists:zip(lists:sublist(L, length(L)-1), tl(L)) ].
111
112 %-----
113
114 % 8.
115 -spec kozepe(M::list([any()])) -> M1::list([any()]).
116 %% M1 az M n*n-es négyzetes mátrix olyan (n/2)*(n/2) méretű részmátrixa,
117 %% mely az n/4+1. sor n/4+1. oszlopának elemétől kezdődik.
118 kozepe(M) ->
119     N = length(M),
120     N4 = N div 4,
121     N2 = N div 2,
122     [ lists:sublist(R, N4 + 1, N2) || R <- lists:sublist(M, N4 + 1, N2) ].
123
124 -spec kozepe2(M::list([any()])) -> M1::list([any()]).
125 kozepe2(M) ->
126     N = length(M),
127     Seq = lists:seq(N div 4 + 1, N div 4 + N div 2),
128     [ [ lists:nth(C, lists:nth(R, M)) || C <- Seq ]
129     || R <- Seq
130     ].
131
132 %-----
133
134 % 9.
135 -spec laposkozepe(M::list([any()])) -> L::[any()].
136 %% Az L lista az M mátrix közepe elemeinek listája.
137 laposkozepe(M) ->
138     lists:flatten(kozepe(M)).
139

```

```

140 -spec laposkozepe2(M::list([any()])) -> L::[any()].
141 laposkozepe2(M) ->
142   N = length(M),
143   Seq = lists:seq(N div 4 + 1, N div 4 + N div 2),
144   [ lists:nth(C, lists:nth(R, M))
145     || R <- Seq,
146     C <- Seq
147   ].
148
149 %-----
150
151 % 10.
152 -spec pivot(M::list([any()]),R::integer(),C::integer()) -> M1::list([any()]).
153 %% M1 az M mátrix R-edik sorának és C-edik oszlopának elhagyásával áll elő.
154 pivot(M, R, C) ->
155   N = length(M),
156   [ [ lists:nth(Co, lists:nth(Ro, M)) || Co <- lists:seq(1, N), Co /= C ]
157     || Ro <- lists:seq(1, N), Ro /= R
158   ].
159
160 %-----
161
162 % 11.
163 -spec transpose(M::list([any()])) -> MT::list([any()]).
164 %% M transzponáltja MT.
165 transpose([]) ->
166   [];
167 transpose([[[] | Matrix]]) ->
168   transpose(Matrix);
169 transpose(Matrix) ->
170   [[H || [H|_] <- Matrix] | transpose([T || [_|T] <- Matrix])].
171
172 -spec transpose2(M::list([any()])) -> MT::list([any()]).
173 transpose2([]) ->
174   [];
175 transpose2([[[] | Matrix]]) ->
176   transpose2(Matrix);
177 transpose2(Matrix) ->
178   [lists:map(fun hd/1, Matrix) | transpose2(lists:map(fun tl/1, Matrix))].
179
180 %-----
181
182 % +1.
183 -spec osszetett(K::integer()) -> L::[integer()].
184 %% L tartalmazza az összetett számokat 4..K*K között, ismétlődés lehet.
185 osszetett(K) ->
186   % lists:usort
187   ([ J || I <- lists:seq(2,K), J <- lists:seq(I*2, K*K, I)]).
188
189 %-----
190
191 % +2.
192 -spec primek(K::integer()) -> L::[integer()].
193 %% L tartalmazza a prímszámokat 2..K*K között.
194 primek(K) ->
195   Os = osszetett(K),
196   [ X || X <- lists:seq(2,K*K), not lists:member(X,Os)].
197
198 %-----
199
200 % +3.
201 -spec zip(Xs::[any()], Ys::[any()]) -> XYs::[{any(), any()}].
202 %% XYs olyan párok listája, amelyek első eleme az Xs, második eleme
203 %% az Ys lista azonos pozíciójú eleme.
204 zip([], []) ->
205   [];
206 zip([H1|T1], [H2|T2]) ->
207   [{H1,H2}|zip(T1,T2)].
208

```

```

209 -spec unzip(XYs::[{any(), any()}]) -> {Xs::[any()], Ys::[any()]}.
210 %% XYs olyan párok listája, amelyek első eleme az Xs, második eleme
211 %% az Ys lista azonos pozíciójú eleme.
212 unzip([]) ->
213   {[], []};
214 unzip([H1|H2|T]) ->
215   {T1,T2} = unzip(T),
216   {H1|T1}, {H2|T2}.
217
218 %-----
219
220 test(lista) ->
221   M=[a,b,e,f],
222     [c,d,g,h],
223     [i,j,m,n],
224     [k,l,o,p]],
225   M0 = [[a,b],
226         [c,d],
227         [e,f]],
228   M1 = [[a,e,i,m],
229         [b,f,j,n],
230         [c,g,k,o],
231         [d,h,l,p]],
232   io:format("~p~n",
233   [[
234     seq(10,13) == [10,11,12,13],
235     % flatten([1,[2,3],[[4]],5,[6]]) == [1,2,3,4,5,6],
236     all_different([1,2,3,1]) == false,
237     all_different([1,2,3]) == true,
238     all_different2([1,2,3,1]) == false,
239     all_different2([1,2,3]) == true,
240     van2eleme([]) == false,
241     van2eleme([a]) == false,
242     van2eleme([a,b]) == true,
243     van2eleme([a,b,c]) == true,
244     duplak([1,2,3]) == [],
245     duplak([1,1,2,3,3,3]) == [1,3,3],
246     all(fun is_atom/1, [a,b,c]) and not all(fun is_atom/1, [a,b,1]),
247     all2(fun is_atom/1, [a,b,c]) and not all(fun is_atom/1, [a,b,1]),
248     all3(fun is_atom/1, [a,b,c]) and not all(fun is_atom/1, [a,b,1]),
249     all4(fun is_atom/1, [a,b,c]) and not all(fun is_atom/1, [a,b,1]),
250     platok_hossza([a,a,a,b,b,c,c,c,d,e,f,f,g,h]) ==
251     [{a,3},{b,2},{c,4},{d,1},{e,1},{f,3},{g,1},{h,1}],
252     platok_hossza([a,a,a,b,b,d,f,f,h]) == [{a,3},{b,2},{d,1},{f,2},{h,1}],
253     sublist([1,2,3,4],2,2) == [2,3],
254     sublist([1,2,3,4],2,3) == [2,3,4],
255     kozepe(M) == [[d,g],[j,m]],
256     kozepe2(M) == [[d,g],[j,m]],
257     laposkozepe(M) == [d,g,j,m],
258     laposkozepe2(M) == [d,g,j,m],
259     pivot(M,2,3) == [[a,b,f],[i,j,n],[k,l,p]],
260     osszetett(5) == [4,6,8,10,12,14,16,18,20,22,24,6,9,
261                    12,15,18,21,24,8,12,16,20,24,10,15,20,25],
262     primek(5) == [2,3,5,7,11,13,17,19,23],
263     zip([1,2,3], [a,b,c]) == [{1,a},{2,b},{3,c}],
264     unzip([1,a],[2,b],[3,c]) == [1,2,3], [a,b,c]],
265     transpose(M0) == [[a,c,e], [b,d,f]],
266     transpose2(M0) == [[a,c,e], [b,d,f]],
267     transpose(M1) == [[a,b,c,d], [e,f,g,h], [i,j,k,l], [m,n,o,p]],
268     transpose2(M1) == [[a,b,c,d], [e,f,g,h], [i,j,k,l], [m,n,o,p]]
269   ]]).

```