```
(*
Title   : SRLottn.Pas
Main    : SeriReda.Pas, SerieRW.Pas,
LastEdit: 2016-08-29 17.50 vers 1.30
Author  : helleforsdata, bosse engborg, 2004-02
        : Evalds Sport-Data HB, Bo Engborg, 1987-06
System  : Microsoft MsDos 6.22
          Borland Pascal 7.02,
          Stony Brook Pascal+ 6.13,
          Turbo Power Professional 5.23,
          Async Professional 2.00,
          B-Tree Filer 5.56,
System  : Microsoft Windows 3.11
          Borland Pascal 7.02,
          Data Entry Workshop 2.03,
          Win/Sys Librayr 1.02,
          B-Tree Filer 5.56,
System  : Microsoft Windows 95/98se/Me/2000/XP/2003
          Borland Delphi 7 personal,
          Turbo Power B-Tree Filer 5.57,
System  : Microsoft Windows 95/98se/Me/2000/XP/2003
          Borland Delphi 2005 architect,
          Turbo Power B-Tree Filer 5.57,
System  : Microsoft Windows 95/98se/Me/2000/XP/2003/Vista/7/8/10
          Windows 32/64
          Embarcadero Delphi XE5 professional
          Turbo Power B-Tree Filer 5.57c,
System  : Microsoft Windows 95/98se/Me/2000/XP/2003/Vista/7/8/10
          Windows 32/64
          Embarcadero Delphi 10.2 community edition
          Turbo Power B-Tree Filer 5.57c,
System  : Microsoft Windows 95/98se/Me/2000/XP/2003/Vista/7/8/10/11
          Windows 32/64
          Embarcadero Delphi 10.4.2 community edition
          Turbo Power B-Tree Filer 5.57d,
*)

{$I redaconfig.inc}

unit SRLottn;

interface

uses
 {$ifdef windows}
 {$ifdef win32orwin64}
  classes, contnrs,
 {$endif}
 {$else}
 {$endif}
  Esdtypes, ESmall, ETider, EMsgTxt,
 {$ifdef win32orwin64}
  Eerror32, ecountform,
 {$else}
  Eerror,
 {$endif}
  EFiler, Eixport,
 {$ifdef windows}
 {$else}
  Econsol, Escreen,
 {$endif}

  SRDefs, SRMisc, SRFilerK, SRFilerP, SRFilerG, SRMakeL,
 {$ifdef win32orwin64}
  srlottnform,
 {$else}
  srinputL,
 {$endif}
```

```
  SRLottX;

procedure makeGameOrder(Aowner: TComponent);

implementation

Type
  TSpeltidRec= record
    Inf: shortstring;
    nG: integer;
    G: Array[1..mxGamesPerTime] Of record
      hemma, borta: integer;
    end;
  end;
  TSpelTid= class
    public
    Speltid: TSpelTidRec;
    constructor Create;
  end;


(*-----------------------------------------------------------------------------*)
constructor TSpelTid.Create;
begin
  FillChar(Speltid,SizeOf(TSpelTidRec),#0);
  inherited;
end;
(*-----------------------------------------------------------------------------*)

(*-----------------------------------------------------------------------------*)
procedure GetSerieOrder(nL, ix: integer; Var lag1, lag2: integer);
Var
  omg, ll, nG: integer;
begin
  lag1:= 0; lag2:= 0;
  nG:= (nL* Pred(nL)) Div 2;
  omg:= (Pred(ix) Div nG)+ 1;
  ix:= (Pred(ix) Mod nG)+ 1;
  If Not(Odd(omg)) And (L^.skd1= 1) Then ix:= Succ(nG- ix);
  Case nL Of
   (* 0 eller 1 lag inga matcher *)
    0, 1: exit;
   (* 2 lag, 1 *)
    2: Case ix Of
         1: begin lag1:= 1; lag2:= 2; end;
       end;
   (* 3 lag, 3 *)
    3: Case ix Of
         1: begin lag1:= 1; lag2:= 2; end;
         2: begin lag1:= 1; lag2:= 3; end;
         3: begin lag1:= 2; lag2:= 3; end;
       end;
   (* 4 lag, 6 *)
    4: Case ix Of
         1: begin lag1:= 1; lag2:= 2; end;
         2: begin lag1:= 3; lag2:= 4; end;
         3: begin lag1:= 1; lag2:= 3; end;
         4: begin lag1:= 2; lag2:= 4; end;
         5: begin lag1:= 1; lag2:= 4; end;
         6: begin lag1:= 2; lag2:= 3; end;
       end;
   (* 5 lag, 10 *)
    5: Case ix Of
         1: begin lag1:= 1; lag2:= 2; end;
         2: begin lag1:= 3; lag2:= 4; end;
         3: begin lag1:= 5; lag2:= 1; end;
         4: begin lag1:= 2; lag2:= 3; end;
         5: begin lag1:= 4; lag2:= 5; end;
```

```
137           6: begin lag1:= 1; lag2:= 3; end;
138           7: begin lag1:= 5; lag2:= 2; end;
139           8: begin lag1:= 4; lag2:= 1; end;
140           9: begin lag1:= 3; lag2:= 5; end;
141          10: begin lag1:= 2; lag2:= 4; end;
142         end;
143      (* 6 lag, 15 *)
144       6: Case ix Of
145           1: begin lag1:= 1; lag2:= 6; end;
146           2: begin lag1:= 5; lag2:= 2; end;
147           3: begin lag1:= 3; lag2:= 4; end;
148           4: begin lag1:= 2; lag2:= 1; end;
149           5: begin lag1:= 6; lag2:= 3; end;
150           6: begin lag1:= 4; lag2:= 5; end;
151           7: begin lag1:= 3; lag2:= 2; end;
152           8: begin lag1:= 6; lag2:= 4; end;
153           9: begin lag1:= 1; lag2:= 5; end;
154          10: begin lag1:= 4; lag2:= 1; end;
155          11: begin lag1:= 5; lag2:= 3; end;
156          12: begin lag1:= 2; lag2:= 6; end;
157          13: begin lag1:= 1; lag2:= 3; end;
158          14: begin lag1:= 2; lag2:= 4; end;
159          15: begin lag1:= 6; lag2:= 5; end;
160         end;
161      (* 7 lag, 21 *)
162       7: Case ix Of
163           1: begin lag1:= 2; lag2:= 1; end;
164           2: begin lag1:= 3; lag2:= 4; end;
165           3: begin lag1:= 5; lag2:= 6; end;
166           4: begin lag1:= 7; lag2:= 1; end;
167           5: begin lag1:= 2; lag2:= 3; end;
168           6: begin lag1:= 4; lag2:= 5; end;
169           7: begin lag1:= 6; lag2:= 7; end;
170           8: begin lag1:= 1; lag2:= 4; end;
171           9: begin lag1:= 5; lag2:= 2; end;
172          10: begin lag1:= 3; lag2:= 7; end;
173          11: begin lag1:= 4; lag2:= 6; end;
174          12: begin lag1:= 1; lag2:= 5; end;
175          13: begin lag1:= 7; lag2:= 4; end;
176          14: begin lag1:= 3; lag2:= 1; end;
177          15: begin lag1:= 6; lag2:= 2; end;
178          16: begin lag1:= 5; lag2:= 3; end;
179          17: begin lag1:= 1; lag2:= 6; end;
180          18: begin lag1:= 2; lag2:= 7; end;
181          19: begin lag1:= 4; lag2:= 2; end;
182          20: begin lag1:= 6; lag2:= 3; end;
183          21: begin lag1:= 7; lag2:= 5; end;
184         end;
185      (* 8 lag, 28 *)
186       8: Case ix Of
187           1: begin lag1:= 1; lag2:= 2; end;
188           2: begin lag1:= 3; lag2:= 4; end;
189           3: begin lag1:= 5; lag2:= 6; end;
190           4: begin lag1:= 7; lag2:= 8; end;
191           5: begin lag1:= 4; lag2:= 1; end;
192           6: begin lag1:= 8; lag2:= 3; end;
193           7: begin lag1:= 5; lag2:= 2; end;
194           8: begin lag1:= 7; lag2:= 6; end;
195           9: begin lag1:= 1; lag2:= 5; end;
196          10: begin lag1:= 3; lag2:= 7; end;
197          11: begin lag1:= 2; lag2:= 4; end;
198          12: begin lag1:= 6; lag2:= 8; end;
199          13: begin lag1:= 5; lag2:= 3; end;
200          14: begin lag1:= 1; lag2:= 7; end;
201          15: begin lag1:= 2; lag2:= 8; end;
202          16: begin lag1:= 4; lag2:= 6; end;
203          17: begin lag1:= 8; lag2:= 5; end;
204          18: begin lag1:= 7; lag2:= 4; end;
```

```
205         19: begin lag1:= 3; lag2:= 2; end;
206         20: begin lag1:= 6; lag2:= 1; end;
207         21: begin lag1:= 4; lag2:= 8; end;
208         22: begin lag1:= 5; lag2:= 7; end;
209         23: begin lag1:= 2; lag2:= 6; end;
210         24: begin lag1:= 1; lag2:= 3; end;
211         25: begin lag1:= 4; lag2:= 5; end;
212         26: begin lag1:= 7; lag2:= 2; end;
213         27: begin lag1:= 8; lag2:= 1; end;
214         28: begin lag1:= 6; lag2:= 3; end;
215        end;
216    (* 9 lag, 36 *)
217     9: Case ix Of
218          1: begin lag1:= 2; lag2:= 7; end;
219          2: begin lag1:= 6; lag2:= 3; end;
220          3: begin lag1:= 5; lag2:= 9; end;
221          4: begin lag1:= 4; lag2:= 8; end;
222          5: begin lag1:= 6; lag2:= 1; end;
223          6: begin lag1:= 8; lag2:= 5; end;
224          7: begin lag1:= 7; lag2:= 4; end;
225          8: begin lag1:= 9; lag2:= 3; end;
226          9: begin lag1:= 8; lag2:= 2; end;
227         10: begin lag1:= 4; lag2:= 9; end;
228         11: begin lag1:= 1; lag2:= 7; end;
229         12: begin lag1:= 5; lag2:= 6; end;
230         13: begin lag1:= 3; lag2:= 1; end;
231         14: begin lag1:= 2; lag2:= 4; end;
232         15: begin lag1:= 9; lag2:= 8; end;
233         16: begin lag1:= 7; lag2:= 5; end;
234         17: begin lag1:= 6; lag2:= 2; end;
235         18: begin lag1:= 8; lag2:= 7; end;
236         19: begin lag1:= 5; lag2:= 3; end;
237         20: begin lag1:= 1; lag2:= 9; end;
238         21: begin lag1:= 4; lag2:= 5; end;
239         22: begin lag1:= 2; lag2:= 1; end;
240         23: begin lag1:= 9; lag2:= 6; end;
241         24: begin lag1:= 3; lag2:= 8; end;
242         25: begin lag1:= 7; lag2:= 9; end;
243         26: begin lag1:= 1; lag2:= 4; end;
244         27: begin lag1:= 6; lag2:= 8; end;
245         28: begin lag1:= 2; lag2:= 3; end;
246         29: begin lag1:= 8; lag2:= 1; end;
247         30: begin lag1:= 5; lag2:= 2; end;
248         31: begin lag1:= 4; lag2:= 6; end;
249         32: begin lag1:= 3; lag2:= 7; end;
250         33: begin lag1:= 9; lag2:= 2; end;
251         34: begin lag1:= 7; lag2:= 6; end;
252         35: begin lag1:= 1; lag2:= 5; end;
253         36: begin lag1:= 3; lag2:= 4; end;
254        end;
255    (* 10 lag, 45 *)
256    10: Case ix Of
257          1: begin lag1:= 1; lag2:= 10; end;
258          2: begin lag1:= 2; lag2:= 8; end;
259          3: begin lag1:= 3; lag2:= 9; end;
260          4: begin lag1:= 4; lag2:= 6; end;
261          5: begin lag1:= 5; lag2:= 7; end;
262          6: begin lag1:= 10; lag2:= 3; end;
263          7: begin lag1:= 8; lag2:= 4; end;
264          8: begin lag1:= 9; lag2:= 5; end;
265          9: begin lag1:= 7; lag2:= 1; end;
266         10: begin lag1:= 6; lag2:= 2; end;
267         11: begin lag1:= 5; lag2:= 10; end;
268         12: begin lag1:= 7; lag2:= 4; end;
269         13: begin lag1:= 1; lag2:= 6; end;
270         14: begin lag1:= 3; lag2:= 8; end;
271         15: begin lag1:= 2; lag2:= 9; end;
272         16: begin lag1:= 8; lag2:= 1; end;
```

```
273          17: begin lag1:= 6; lag2:= 3; end;
274          18: begin lag1:= 10; lag2:= 2; end;
275          19: begin lag1:= 9; lag2:= 7; end;
276          20: begin lag1:= 4; lag2:= 5; end;
277          21: begin lag1:= 1; lag2:= 3; end;
278          22: begin lag1:= 2; lag2:= 5; end;
279          23: begin lag1:= 6; lag2:= 9; end;
280          24: begin lag1:= 10; lag2:= 4; end;
281          25: begin lag1:= 7; lag2:= 8; end;
282          26: begin lag1:= 9; lag2:= 1; end;
283          27: begin lag1:= 5; lag2:= 6; end;
284          28: begin lag1:= 8; lag2:= 10; end;
285          29: begin lag1:= 3; lag2:= 7; end;
286          30: begin lag1:= 4; lag2:= 2; end;
287          31: begin lag1:= 1; lag2:= 4; end;
288          32: begin lag1:= 2; lag2:= 7; end;
289          33: begin lag1:= 3; lag2:= 5; end;
290          34: begin lag1:= 6; lag2:= 10; end;
291          35: begin lag1:= 8; lag2:= 9; end;
292          36: begin lag1:= 4; lag2:= 3; end;
293          37: begin lag1:= 7; lag2:= 6; end;
294          38: begin lag1:= 2; lag2:= 1; end;
295          39: begin lag1:= 10; lag2:= 9; end;
296          40: begin lag1:= 5; lag2:= 8; end;
297          41: begin lag1:= 1; lag2:= 5; end;
298          42: begin lag1:= 3; lag2:= 2; end;
299          43: begin lag1:= 9; lag2:= 4; end;
300          44: begin lag1:= 6; lag2:= 8; end;
301          45: begin lag1:= 7; lag2:= 10; end;
302        end;
303    (* 11 lag, 55 *)
304    11: Case ix Of
305          1: begin lag1:= 2; lag2:= 7; end;
306          2: begin lag1:= 3; lag2:=11; end;
307          3: begin lag1:= 9; lag2:= 5; end;
308          4: begin lag1:= 4; lag2:= 8; end;
309          5: begin lag1:=10; lag2:= 6; end;
310          6: begin lag1:= 8; lag2:= 1; end;
311          7: begin lag1:= 7; lag2:=10; end;
312          8: begin lag1:= 6; lag2:= 3; end;
313          9: begin lag1:=11; lag2:= 9; end;
314         10: begin lag1:= 5; lag2:= 4; end;
315         11: begin lag1:= 6; lag2:= 2; end;
316         12: begin lag1:= 1; lag2:= 5; end;
317         13: begin lag1:= 9; lag2:= 8; end;
318         14: begin lag1:=10; lag2:=11; end;
319         15: begin lag1:= 4; lag2:= 7; end;
320         16: begin lag1:= 5; lag2:=10; end;
321         17: begin lag1:=11; lag2:= 1; end;
322         18: begin lag1:= 3; lag2:= 7; end;
323         19: begin lag1:= 2; lag2:= 9; end;
324         20: begin lag1:= 8; lag2:= 6; end;
325         21: begin lag1:= 1; lag2:= 4; end;
326         22: begin lag1:=10; lag2:= 2; end;
327         23: begin lag1:= 7; lag2:= 8; end;
328         24: begin lag1:=11; lag2:= 6; end;
329         25: begin lag1:= 9; lag2:= 3; end;
330         26: begin lag1:= 7; lag2:= 5; end;
331         27: begin lag1:= 3; lag2:= 1; end;
332         28: begin lag1:= 8; lag2:=10; end;
333         29: begin lag1:= 2; lag2:=11; end;
334         30: begin lag1:= 4; lag2:= 9; end;
335         31: begin lag1:=10; lag2:= 3; end;
336         32: begin lag1:= 1; lag2:= 2; end;
337         33: begin lag1:= 6; lag2:= 9; end;
338         34: begin lag1:=11; lag2:= 4; end;
339         35: begin lag1:= 5; lag2:= 8; end;
340         36: begin lag1:= 9; lag2:= 7; end;
```

```
341          37: begin lag1:= 2; lag2:= 4; end;
342          38: begin lag1:= 6; lag2:= 1; end;
343          39: begin lag1:= 8; lag2:=11; end;
344          40: begin lag1:= 3; lag2:= 5; end;
345          41: begin lag1:= 4; lag2:=10; end;
346          42: begin lag1:= 5; lag2:= 2; end;
347          43: begin lag1:= 3; lag2:= 8; end;
348          44: begin lag1:= 9; lag2:= 1; end;
349          45: begin lag1:= 7; lag2:= 6; end;
350          46: begin lag1:=11; lag2:= 5; end;
351          47: begin lag1:=10; lag2:= 9; end;
352          48: begin lag1:= 2; lag2:= 3; end;
353          49: begin lag1:= 6; lag2:= 4; end;
354          50: begin lag1:= 1; lag2:= 7; end;
355          51: begin lag1:= 8; lag2:= 2; end;
356          52: begin lag1:= 4; lag2:= 3; end;
357          53: begin lag1:= 5; lag2:= 6; end;
358          54: begin lag1:= 7; lag2:=11; end;
359          55: begin lag1:= 1; lag2:=10; end;
360        end;
361    (* 12 lag, 66*)
362    12: Case ix Of
363           1: begin lag1:= 1; lag2:=10; end;
364           2: begin lag1:= 8; lag2:= 2; end;
365           3: begin lag1:= 3; lag2:=12; end;
366           4: begin lag1:= 9; lag2:=11; end;
367           5: begin lag1:= 4; lag2:= 7; end;
368           6: begin lag1:= 6; lag2:= 5; end;
369           7: begin lag1:=12; lag2:= 1; end;
370           8: begin lag1:=10; lag2:= 3; end;
371           9: begin lag1:= 2; lag2:= 4; end;
372          10: begin lag1:= 7; lag2:= 8; end;
373          11: begin lag1:=11; lag2:= 6; end;
374          12: begin lag1:= 5; lag2:= 9; end;
375          13: begin lag1:= 1; lag2:= 7; end;
376          14: begin lag1:= 8; lag2:=12; end;
377          15: begin lag1:= 4; lag2:=11; end;
378          16: begin lag1:= 2; lag2:= 6; end;
379          17: begin lag1:= 3; lag2:= 5; end;
380          18: begin lag1:= 9; lag2:=10; end;
381          19: begin lag1:= 5; lag2:= 1; end;
382          20: begin lag1:= 7; lag2:= 3; end;
383          21: begin lag1:= 6; lag2:= 9; end;
384          22: begin lag1:=10; lag2:= 2; end;
385          23: begin lag1:=12; lag2:= 4; end;
386          24: begin lag1:=11; lag2:= 8; end;
387          25: begin lag1:= 4; lag2:= 6; end;
388          26: begin lag1:= 9; lag2:=12; end;
389          27: begin lag1:= 8; lag2:=10; end;
390          28: begin lag1:= 2; lag2:=11; end;
391          29: begin lag1:= 1; lag2:= 3; end;
392          30: begin lag1:= 7; lag2:= 5; end;
393          31: begin lag1:=10; lag2:=12; end;
394          32: begin lag1:= 8; lag2:= 9; end;
395          33: begin lag1:= 5; lag2:= 2; end;
396          34: begin lag1:=11; lag2:= 7; end;
397          35: begin lag1:= 3; lag2:= 4; end;
398          36: begin lag1:= 6; lag2:= 1; end;
399          37: begin lag1:=10; lag2:= 5; end;
400          38: begin lag1:=12; lag2:= 2; end;
401          39: begin lag1:= 4; lag2:= 8; end;
402          40: begin lag1:= 9; lag2:= 3; end;
403          41: begin lag1:= 1; lag2:=11; end;
404          42: begin lag1:= 7; lag2:= 6; end;
405          43: begin lag1:= 2; lag2:= 1; end;
406          44: begin lag1:= 3; lag2:= 8; end;
407          45: begin lag1:= 5; lag2:= 4; end;
408          46: begin lag1:= 6; lag2:=12; end;
```

```
409         47: begin lag1:= 7; lag2:= 9; end;
410         48: begin lag1:=11; lag2:=10; end;
411         49: begin lag1:= 1; lag2:= 9; end;
412         50: begin lag1:= 2; lag2:= 7; end;
413         51: begin lag1:= 3; lag2:=11; end;
414         52: begin lag1:=10; lag2:= 4; end;
415         53: begin lag1:=12; lag2:= 5; end;
416         54: begin lag1:= 8; lag2:= 6; end;
417         55: begin lag1:= 4; lag2:= 1; end;
418         56: begin lag1:= 9; lag2:= 2; end;
419         57: begin lag1:= 6; lag2:= 3; end;
420         58: begin lag1:= 5; lag2:= 8; end;
421         59: begin lag1:= 7; lag2:=10; end;
422         60: begin lag1:=11; lag2:=12; end;
423         61: begin lag1:= 3; lag2:= 2; end;
424         62: begin lag1:= 4; lag2:= 9; end;
425         63: begin lag1:= 1; lag2:= 8; end;
426         64: begin lag1:= 5; lag2:=11; end;
427         65: begin lag1:=10; lag2:= 6; end;
428         66: begin lag1:=12; lag2:= 7; end;
429       end;
430   (* 13 lag, 78*)
431   13: Case ix Of
432         1: begin lag1:= 1; lag2:= 13; end;
433         2: begin lag1:= 2; lag2:= 12; end;
434         3: begin lag1:= 3; lag2:= 11; end;
435         4: begin lag1:= 4; lag2:= 10; end;
436         5: begin lag1:= 5; lag2:= 9; end;
437         6: begin lag1:= 6; lag2:= 8; end;
438
439         7: begin lag1:= 1; lag2:= 12; end;
440         8: begin lag1:=13; lag2:= 11; end;
441         9: begin lag1:= 2; lag2:= 10; end;
442        10: begin lag1:= 3; lag2:= 9; end;
443        11: begin lag1:= 4; lag2:= 8; end;
444        12: begin lag1:= 6; lag2:= 7; end;
445
446        13: begin lag1:= 1; lag2:= 11; end;
447        14: begin lag1:=12; lag2:= 10; end;
448        15: begin lag1:=13; lag2:= 9; end;
449        16: begin lag1:= 2; lag2:= 8; end;
450        17: begin lag1:= 4; lag2:= 7; end;
451        18: begin lag1:= 5; lag2:= 6; end;
452
453        19: begin lag1:= 1; lag2:= 10; end;
454        20: begin lag1:=11; lag2:= 9; end;
455        21: begin lag1:=12; lag2:= 8; end;
456        22: begin lag1:= 2; lag2:= 7; end;
457        23: begin lag1:= 3; lag2:= 6; end;
458        24: begin lag1:= 4; lag2:= 5; end;
459
460        25: begin lag1:= 1; lag2:= 9; end;
461        26: begin lag1:=10; lag2:= 8; end;
462        27: begin lag1:=12; lag2:= 7; end;
463        28: begin lag1:=13; lag2:= 6; end;
464        29: begin lag1:= 2; lag2:= 5; end;
465        30: begin lag1:= 3; lag2:= 4; end;
466
467        31: begin lag1:= 1; lag2:= 8; end;
468        32: begin lag1:=10; lag2:= 7; end;
469        33: begin lag1:=11; lag2:= 6; end;
470        34: begin lag1:=12; lag2:= 5; end;
471        35: begin lag1:=13; lag2:= 4; end;
472        36: begin lag1:= 2; lag2:= 3; end;
473
474        37: begin lag1:= 8; lag2:= 7; end;
475        38: begin lag1:= 9; lag2:= 6; end;
476        39: begin lag1:=10; lag2:= 5; end;
```

```
        40: begin lag1:=11; lag2:= 4; end;
        41: begin lag1:=12; lag2:= 3; end;
        42: begin lag1:=13; lag2:= 2; end;

        43: begin lag1:= 1; lag2:= 7; end;
        44: begin lag1:= 8; lag2:= 5; end;
        45: begin lag1:= 9; lag2:= 4; end;
        46: begin lag1:=10; lag2:= 3; end;
        47: begin lag1:=11; lag2:= 2; end;
        48: begin lag1:=12; lag2:=13; end;

        49: begin lag1:= 1; lag2:= 6; end;
        50: begin lag1:= 7; lag2:= 5; end;
        51: begin lag1:= 8; lag2:= 3; end;
        52: begin lag1:= 9; lag2:= 2; end;
        53: begin lag1:=10; lag2:=13; end;
        54: begin lag1:=11; lag2:=12; end;

        55: begin lag1:= 1; lag2:= 5; end;
        56: begin lag1:= 6; lag2:= 4; end;
        57: begin lag1:= 7; lag2:= 3; end;
        58: begin lag1:= 8; lag2:=13; end;
        59: begin lag1:= 9; lag2:=12; end;
        60: begin lag1:=10; lag2:=11; end;

        61: begin lag1:= 1; lag2:= 4; end;
        62: begin lag1:= 5; lag2:= 3; end;
        63: begin lag1:= 6; lag2:= 2; end;
        64: begin lag1:= 7; lag2:=13; end;
        65: begin lag1:= 8; lag2:=11; end;
        66: begin lag1:= 9; lag2:=10; end;

        67: begin lag1:= 1; lag2:= 3; end;
        68: begin lag1:= 4; lag2:= 2; end;
        69: begin lag1:= 5; lag2:=13; end;
        70: begin lag1:= 6; lag2:=12; end;
        71: begin lag1:= 7; lag2:=11; end;
        72: begin lag1:= 8; lag2:= 9; end;

        73: begin lag1:= 1; lag2:= 2; end;
        74: begin lag1:= 3; lag2:=13; end;
        75: begin lag1:= 4; lag2:=12; end;
        76: begin lag1:= 5; lag2:=11; end;
        77: begin lag1:= 6; lag2:=10; end;
        78: begin lag1:= 7; lag2:= 9; end;
      end;
    (* 14 lag, 91*)
    14: Case ix Of
          1: begin lag1:= 1; lag2:= 13; end;
          2: begin lag1:= 2; lag2:= 12; end;
          3: begin lag1:= 3; lag2:= 11; end;
          4: begin lag1:= 4; lag2:= 10; end;
          5: begin lag1:= 5; lag2:= 9; end;
          6: begin lag1:= 6; lag2:= 8; end;
          7: begin lag1:= 7; lag2:= 14; end;

          8: begin lag1:= 1; lag2:= 12; end;
          9: begin lag1:=13; lag2:= 11; end;
         10: begin lag1:= 2; lag2:= 10; end;
         11: begin lag1:= 3; lag2:= 9; end;
         12: begin lag1:= 4; lag2:= 8; end;
         13: begin lag1:= 5; lag2:= 14; end;
         14: begin lag1:= 6; lag2:= 7; end;

         15: begin lag1:= 1; lag2:= 11; end;
         16: begin lag1:=12; lag2:= 10; end;
         17: begin lag1:=13; lag2:= 9; end;
         18: begin lag1:= 2; lag2:= 8; end;
```

```
545         19: begin lag1:= 3; lag2:= 14; end;
546         20: begin lag1:= 4; lag2:= 7; end;
547         21: begin lag1:= 5; lag2:= 6; end;
548
549         22: begin lag1:= 1; lag2:= 10; end;
550         23: begin lag1:=11; lag2:= 9; end;
551         24: begin lag1:=12; lag2:= 8; end;
552         25: begin lag1:=13; lag2:= 14; end;
553         26: begin lag1:= 2; lag2:= 7; end;
554         27: begin lag1:= 3; lag2:= 6; end;
555         28: begin lag1:= 4; lag2:= 5; end;
556
557         29: begin lag1:= 1; lag2:= 9; end;
558         30: begin lag1:=10; lag2:= 8; end;
559         31: begin lag1:=11; lag2:= 14; end;
560         32: begin lag1:=12; lag2:= 7; end;
561         33: begin lag1:=13; lag2:= 6; end;
562         34: begin lag1:= 2; lag2:= 5; end;
563         35: begin lag1:= 3; lag2:= 4; end;
564
565         36: begin lag1:= 1; lag2:= 8; end;
566         37: begin lag1:= 9; lag2:= 14; end;
567         38: begin lag1:=10; lag2:= 7; end;
568         39: begin lag1:=11; lag2:= 6; end;
569         40: begin lag1:=12; lag2:= 5; end;
570         41: begin lag1:=13; lag2:= 4; end;
571         42: begin lag1:= 2; lag2:= 3; end;
572
573         43: begin lag1:= 1; lag2:= 14; end;
574         44: begin lag1:= 8; lag2:= 7; end;
575         45: begin lag1:= 9; lag2:= 6; end;
576         46: begin lag1:=10; lag2:= 5; end;
577         47: begin lag1:=11; lag2:= 4; end;
578         48: begin lag1:=12; lag2:= 3; end;
579         49: begin lag1:=13; lag2:= 2; end;
580
581         50: begin lag1:= 1; lag2:= 7; end;
582         51: begin lag1:= 14; lag2:= 6; end;
583         52: begin lag1:= 8; lag2:= 5; end;
584         53: begin lag1:= 9; lag2:= 4; end;
585         54: begin lag1:=10; lag2:= 3; end;
586         55: begin lag1:=11; lag2:= 2; end;
587         56: begin lag1:=12; lag2:=13; end;
588
589         57: begin lag1:= 1; lag2:= 6; end;
590         58: begin lag1:= 7; lag2:= 5; end;
591         59: begin lag1:= 14; lag2:= 4; end;
592         60: begin lag1:= 8; lag2:= 3; end;
593         61: begin lag1:= 9; lag2:= 2; end;
594         62: begin lag1:=10; lag2:=13; end;
595         63: begin lag1:=11; lag2:=12; end;
596
597         64: begin lag1:= 1; lag2:= 5; end;
598         65: begin lag1:= 6; lag2:= 4; end;
599         66: begin lag1:= 7; lag2:= 3; end;
600         67: begin lag1:= 14; lag2:= 2; end;
601         68: begin lag1:= 8; lag2:=13; end;
602         69: begin lag1:= 9; lag2:=12; end;
603         70: begin lag1:=10; lag2:=11; end;
604
605         71: begin lag1:= 1; lag2:= 4; end;
606         72: begin lag1:= 5; lag2:= 3; end;
607         73: begin lag1:= 6; lag2:= 2; end;
608         74: begin lag1:= 7; lag2:=13; end;
609         75: begin lag1:= 14; lag2:=12; end;
610         76: begin lag1:= 8; lag2:=11; end;
611         77: begin lag1:= 9; lag2:=10; end;
612
```

```
         78: begin lag1:= 1; lag2:= 3; end;
         79: begin lag1:= 4; lag2:= 2; end;
         80: begin lag1:= 5; lag2:=13; end;
         81: begin lag1:= 6; lag2:=12; end;
         82: begin lag1:= 7; lag2:=11; end;
         83: begin lag1:= 14; lag2:=10; end;
         84: begin lag1:= 8; lag2:= 9; end;

         85: begin lag1:= 1; lag2:= 2; end;
         86: begin lag1:= 3; lag2:=13; end;
         87: begin lag1:= 4; lag2:=12; end;
         88: begin lag1:= 5; lag2:=11; end;
         89: begin lag1:= 6; lag2:=10; end;
         90: begin lag1:= 7; lag2:= 9; end;
         91: begin lag1:= 14; lag2:= 8; end;
        end;
    (* 15 lag, 105*)
    15: Case ix Of
          1: begin lag1:= 1; lag2:=15; end;
          2: begin lag1:= 2; lag2:=14; end;
          3: begin lag1:= 3; lag2:=13; end;
          4: begin lag1:= 4; lag2:=12; end;
          5: begin lag1:= 5; lag2:=11; end;
          6: begin lag1:= 6; lag2:=10; end;
          7: begin lag1:= 7; lag2:= 9; end;

          8: begin lag1:= 1; lag2:=14; end;
          9: begin lag1:=15; lag2:=13; end;
         10: begin lag1:= 2; lag2:=12; end;
         11: begin lag1:= 3; lag2:=11; end;
         12: begin lag1:= 4; lag2:=10; end;
         13: begin lag1:= 5; lag2:= 9; end;
         14: begin lag1:= 7; lag2:= 8; end;

         15: begin lag1:= 1; lag2:=13; end;
         16: begin lag1:=14; lag2:=12; end;
         17: begin lag1:=15; lag2:=11; end;
         18: begin lag1:= 2; lag2:=10; end;
         19: begin lag1:= 3; lag2:= 9; end;
         20: begin lag1:= 5; lag2:= 8; end;
         21: begin lag1:= 6; lag2:= 7; end;

         22: begin lag1:= 1; lag2:=12; end;
         23: begin lag1:=13; lag2:=11; end;
         24: begin lag1:=14; lag2:=10; end;
         25: begin lag1:=15; lag2:= 9; end;
         26: begin lag1:= 3; lag2:= 8; end;
         27: begin lag1:= 4; lag2:= 7; end;
         28: begin lag1:= 5; lag2:= 6; end;

         29: begin lag1:= 1; lag2:=11; end;
         30: begin lag1:=12; lag2:=10; end;
         31: begin lag1:=13; lag2:= 9; end;
         32: begin lag1:=15; lag2:= 8; end;
         33: begin lag1:= 2; lag2:= 7; end;
         34: begin lag1:= 3; lag2:= 6; end;
         35: begin lag1:= 4; lag2:= 5; end;

         36: begin lag1:= 1; lag2:=10; end;
         37: begin lag1:=11; lag2:= 9; end;
         38: begin lag1:=13; lag2:= 8; end;
         39: begin lag1:=14; lag2:= 7; end;
         40: begin lag1:=15; lag2:= 6; end;
         41: begin lag1:= 2; lag2:= 5; end;
         42: begin lag1:= 3; lag2:= 4; end;

         43: begin lag1:= 1; lag2:= 9; end;
         44: begin lag1:=11; lag2:= 8; end;
```

```
681          45: begin lag1:=12; lag2:= 7; end;
682          46: begin lag1:=13; lag2:= 6; end;
683          47: begin lag1:=14; lag2:= 5; end;
684          48: begin lag1:=15; lag2:= 4; end;
685          49: begin lag1:= 2; lag2:= 3; end;
686
687          50: begin lag1:= 9; lag2:= 8; end;
688          51: begin lag1:=10; lag2:= 7; end;
689          52: begin lag1:=11; lag2:= 6; end;
690          53: begin lag1:=12; lag2:= 5; end;
691          54: begin lag1:=13; lag2:= 4; end;
692          55: begin lag1:=14; lag2:= 3; end;
693          56: begin lag1:=15; lag2:= 2; end;
694
695          57: begin lag1:= 1; lag2:= 8; end;
696          58: begin lag1:= 9; lag2:= 6; end;
697          59: begin lag1:=10; lag2:= 5; end;
698          60: begin lag1:=11; lag2:= 4; end;
699          61: begin lag1:=12; lag2:= 3; end;
700          62: begin lag1:=13; lag2:= 2; end;
701          63: begin lag1:=14; lag2:=15; end;
702
703          64: begin lag1:= 1; lag2:= 7; end;
704          65: begin lag1:= 8; lag2:= 6; end;
705          66: begin lag1:= 9; lag2:= 4; end;
706          67: begin lag1:=10; lag2:= 3; end;
707          68: begin lag1:=11; lag2:= 2; end;
708          69: begin lag1:=12; lag2:=15; end;
709          70: begin lag1:=13; lag2:=14; end;
710
711          71: begin lag1:= 1; lag2:= 6; end;
712          72: begin lag1:= 7; lag2:= 5; end;
713          73: begin lag1:= 8; lag2:= 4; end;
714          74: begin lag1:= 9; lag2:= 2; end;
715          75: begin lag1:=10; lag2:=15; end;
716          76: begin lag1:=11; lag2:=14; end;
717          77: begin lag1:=12; lag2:=13; end;
718
719          78: begin lag1:= 1; lag2:= 5; end;
720          79: begin lag1:= 6; lag2:= 4; end;
721          80: begin lag1:= 7; lag2:= 3; end;
722          81: begin lag1:= 8; lag2:= 2; end;
723          82: begin lag1:= 9; lag2:=14; end;
724          83: begin lag1:=10; lag2:=13; end;
725          84: begin lag1:=11; lag2:=12; end;
726
727          85: begin lag1:= 1; lag2:= 4; end;
728          86: begin lag1:= 5; lag2:= 3; end;
729          87: begin lag1:= 6; lag2:= 2; end;
730          88: begin lag1:= 7; lag2:=15; end;
731          89: begin lag1:= 8; lag2:=14; end;
732          90: begin lag1:= 9; lag2:=12; end;
733          91: begin lag1:=10; lag2:=11; end;
734
735          92: begin lag1:= 1; lag2:= 3; end;
736          93: begin lag1:= 4; lag2:= 2; end;
737          94: begin lag1:= 5; lag2:=15; end;
738          95: begin lag1:= 6; lag2:=14; end;
739          96: begin lag1:= 7; lag2:=13; end;
740          97: begin lag1:= 8; lag2:=12; end;
741          98: begin lag1:= 9; lag2:=10; end;
742
743          99: begin lag1:= 1; lag2:= 2; end;
744         100: begin lag1:= 3; lag2:=15; end;
745         101: begin lag1:= 4; lag2:=14; end;
746         102: begin lag1:= 5; lag2:=13; end;
747         103: begin lag1:= 6; lag2:=12; end;
748         104: begin lag1:= 7; lag2:=11; end;
```

```
749         105: begin lag1:= 8; lag2:=10; end;
750         end;
751     (* 16 lag, 120*)
752     16: Case ix Of
753           1: begin lag1:= 1; lag2:=15; end;
754           2: begin lag1:= 2; lag2:=14; end;
755           3: begin lag1:= 3; lag2:=13; end;
756           4: begin lag1:= 4; lag2:=12; end;
757           5: begin lag1:= 5; lag2:=11; end;
758           6: begin lag1:= 6; lag2:=10; end;
759           7: begin lag1:= 7; lag2:= 9; end;
760           8: begin lag1:= 8; lag2:= 16; end;
761
762           9: begin lag1:= 1; lag2:=14; end;
763          10: begin lag1:=15; lag2:=13; end;
764          11: begin lag1:= 2; lag2:=12; end;
765          12: begin lag1:= 3; lag2:=11; end;
766          13: begin lag1:= 4; lag2:=10; end;
767          14: begin lag1:= 5; lag2:= 9; end;
768          15: begin lag1:= 6; lag2:= 16; end;
769          16: begin lag1:= 7; lag2:= 8; end;
770
771          17: begin lag1:= 1; lag2:=13; end;
772          18: begin lag1:=14; lag2:=12; end;
773          19: begin lag1:=15; lag2:=11; end;
774          20: begin lag1:= 2; lag2:=10; end;
775          21: begin lag1:= 3; lag2:= 9; end;
776          22: begin lag1:= 4; lag2:= 16; end;
777          23: begin lag1:= 5; lag2:= 8; end;
778          24: begin lag1:= 6; lag2:= 7; end;
779
780          25: begin lag1:= 1; lag2:=12; end;
781          26: begin lag1:=13; lag2:=11; end;
782          27: begin lag1:=14; lag2:=10; end;
783          28: begin lag1:=15; lag2:= 9; end;
784          29: begin lag1:= 2; lag2:= 16; end;
785          30: begin lag1:= 3; lag2:= 8; end;
786          31: begin lag1:= 4; lag2:= 7; end;
787          32: begin lag1:= 5; lag2:= 6; end;
788
789          33: begin lag1:= 1; lag2:=11; end;
790          34: begin lag1:=12; lag2:=10; end;
791          35: begin lag1:=13; lag2:= 9; end;
792          36: begin lag1:=14; lag2:= 16; end;
793          37: begin lag1:=15; lag2:= 8; end;
794          38: begin lag1:= 2; lag2:= 7; end;
795          39: begin lag1:= 3; lag2:= 6; end;
796          40: begin lag1:= 4; lag2:= 5; end;
797
798          41: begin lag1:= 1; lag2:=10; end;
799          42: begin lag1:=11; lag2:= 9; end;
800          43: begin lag1:=12; lag2:= 16; end;
801          44: begin lag1:=13; lag2:= 8; end;
802          45: begin lag1:=14; lag2:= 7; end;
803          46: begin lag1:=15; lag2:= 6; end;
804          47: begin lag1:= 2; lag2:= 5; end;
805          48: begin lag1:= 3; lag2:= 4; end;
806
807          49: begin lag1:= 1; lag2:= 9; end;
808          50: begin lag1:=10; lag2:= 16; end;
809          51: begin lag1:=11; lag2:= 8; end;
810          52: begin lag1:=12; lag2:= 7; end;
811          53: begin lag1:=13; lag2:= 6; end;
812          54: begin lag1:=14; lag2:= 5; end;
813          55: begin lag1:=15; lag2:= 4; end;
814          56: begin lag1:= 2; lag2:= 3; end;
815
816          57: begin lag1:= 1; lag2:= 16; end;
```

```
817          58: begin lag1:= 9; lag2:= 8; end;
818          59: begin lag1:=10; lag2:= 7; end;
819          60: begin lag1:=11; lag2:= 6; end;
820          61: begin lag1:=12; lag2:= 5; end;
821          62: begin lag1:=13; lag2:= 4; end;
822          63: begin lag1:=14; lag2:= 3; end;
823          64: begin lag1:=15; lag2:= 2; end;
824
825          65: begin lag1:= 1; lag2:= 8; end;
826          66: begin lag1:= 16; lag2:= 7; end;
827          67: begin lag1:= 9; lag2:= 6; end;
828          68: begin lag1:=10; lag2:= 5; end;
829          69: begin lag1:=11; lag2:= 4; end;
830          70: begin lag1:=12; lag2:= 3; end;
831          71: begin lag1:=13; lag2:= 2; end;
832          72: begin lag1:=14; lag2:=15; end;
833
834          73: begin lag1:= 1; lag2:= 7; end;
835          74: begin lag1:= 8; lag2:= 6; end;
836          75: begin lag1:= 16; lag2:= 5; end;
837          76: begin lag1:= 9; lag2:= 4; end;
838          77: begin lag1:=10; lag2:= 3; end;
839          78: begin lag1:=11; lag2:= 2; end;
840          79: begin lag1:=12; lag2:=15; end;
841          80: begin lag1:=13; lag2:=14; end;
842
843          81: begin lag1:= 1; lag2:= 6; end;
844          82: begin lag1:= 7; lag2:= 5; end;
845          83: begin lag1:= 8; lag2:= 4; end;
846          84: begin lag1:= 16; lag2:= 3; end;
847          85: begin lag1:= 9; lag2:= 2; end;
848          86: begin lag1:=10; lag2:=15; end;
849          87: begin lag1:=11; lag2:=14; end;
850          88: begin lag1:=12; lag2:=13; end;
851
852          89: begin lag1:= 1; lag2:= 5; end;
853          90: begin lag1:= 6; lag2:= 4; end;
854          91: begin lag1:= 7; lag2:= 3; end;
855          92: begin lag1:= 8; lag2:= 2; end;
856          93: begin lag1:= 16; lag2:=15; end;
857          94: begin lag1:= 9; lag2:=14; end;
858          95: begin lag1:=10; lag2:=13; end;
859          96: begin lag1:=11; lag2:=12; end;
860
861          97: begin lag1:= 1; lag2:= 4; end;
862          98: begin lag1:= 5; lag2:= 3; end;
863          99: begin lag1:= 6; lag2:= 2; end;
864         100: begin lag1:= 7; lag2:=15; end;
865         101: begin lag1:= 8; lag2:=14; end;
866         102: begin lag1:= 16; lag2:=13; end;
867         103: begin lag1:= 9; lag2:=12; end;
868         104: begin lag1:=10; lag2:=11; end;
869
870         105: begin lag1:= 1; lag2:= 3; end;
871         106: begin lag1:= 4; lag2:= 2; end;
872         107: begin lag1:= 5; lag2:=15; end;
873         108: begin lag1:= 6; lag2:=14; end;
874         109: begin lag1:= 7; lag2:=13; end;
875         110: begin lag1:= 8; lag2:=12; end;
876         111: begin lag1:= 16; lag2:=11; end;
877         112: begin lag1:= 9; lag2:=10; end;
878
879         113: begin lag1:= 1; lag2:= 2; end;
880         114: begin lag1:= 3; lag2:=15; end;
881         115: begin lag1:= 4; lag2:=14; end;
882         116: begin lag1:= 5; lag2:=13; end;
883         117: begin lag1:= 6; lag2:=12; end;
884         118: begin lag1:= 7; lag2:=11; end;
```

```
885         119: begin lag1:= 8; lag2:=10; end;
886         120: begin lag1:= 16; lag2:= 9; end;
887         end;
888     end;
889     If Not(Odd(omg)) And (L^.skd2= 1) Then begin
890       ll:= lag1; lag1:= lag2; lag2:= ll;
891     end;
892   end;
893   (*---------------------------------------------------------------------------*)
894   procedure Hatten(fa, n: integer; Var nRes: HattenType);
895   Var
896     i, k: integer;
897     aRes: HattenType;
898   begin
899     If n= 0 Then exit;
900     RandoMize;
901     For i:= 1 To n Do aRes[i]:= i;
902     For i:= n DownTo 1 Do begin
903       k:= Random(i)+ 1;
904       nRes[i]:= aRes[k]; aRes[k]:= aRes[i];
905     end;
906     For i:= 1 To n Do nRes[i]:= nRes[i]+ fa-1;
907   end;
908   (*---------------------------------------------------------------------------*)
909   procedure SetKlasser(j4, j5: integer; VAR n: integer);
910   Var
911     found: boolean;
912     i, j: integer;
913     dummyi: integer;
914   begin
915     FillChar(KlassSet^,SizeOf(KlassSet^),#0);
916     n:= 0; i:= j4- 1;
917     While (i< j5) Do begin
918       Inc(i); j:= 0;
919       Repeat
920         Inc(j);
921         found:= (SRs^[i].kKlass= KlassSet^[j].namn);
922       Until (j>= n) Or found;
923       If Not(found) Then begin
924         Inc(n);
925         If (n>= mxGames) Then exit;
926         KlassSet^[n].namn:= SRs^[i].kKlass;
927       end;
928     end;
929     For i:= 1 To n Do begin
930       KlassSet^[i].antal:= 0;
931       For j:= j4 To j5 Do begin
932         If (KlassSet^[i].namn= SRs^[j].kKlass) Then
933           Inc(KlassSet^[i].antal);
934       end;
935     end;
936     For i:= 1 To n- 1 Do begin
937       For j:= i+ 1 To n Do begin
938         If Not(KlassSet^[i].antal>= KlassSet^[j].antal) Then begin
939           dummyi:= KlassSet^[i].antal;
940           KlassSet^[i].antal:= KlassSet^[j].antal;
941           KlassSet^[j].antal:= dummyi;
942           dummyi:= KlassSet^[i].namn;
943           KlassSet^[i].namn:= KlassSet^[j].namn;
944           KlassSet^[j].namn:= dummyi;
945         end;
946       end;
947     end;
948   end;
949   (*---------------------------------------------------------------------------*)
950   procedure Lottning(j3, j4, j5, nKlasser: integer);
951   Var
952     hsort: HattenType;
```

```
  y2, r2, s2, jj, p2, z2, inv, zz2, c, x, k, q, t, i2: integer;
  j, j2, l2: real;
  aaa: boolean;
  dumSRs: SRType;
  (*-------------------------------------------------------------------------*)
  procedure Nolla;
  begin
    jj:= jj-p2; s2:= j3; r2:= j3; y2:= y2+ i2; i2:= 0;
  end;
  (*-------------------------------------------------------------------------*)
  procedure Slumptal;
  Var
    ndx: integer;
    dummyi: integer;
  begin
    Repeat
      k:= Random(j5- j4+ 1)+ j4;
    Until (SRs^[k].kKlass = KlassSet^[c].namn) And Not(SRs^[k].ee> 10000);
    l2:= l2+ j2; q:= Round(l2)+ j3;
    Repeat
      Repeat
        Inc(r2);
      Until Not(SRs^[r2].ee> 10000);
      Inc(s2);
    Until Not(s2<> q);
    dummyi:= SRs^[r2].ee; SRs^[r2].ee:= SRs^[k].ee;
    SRs^[k].ee:= dummyi; SRs^[r2].ee:= SRs^[r2].ee+ 10000;
    dummyi:= SRs^[r2].lag1; SRs^[r2].lag1:= SRs^[k].lag1;
    SRs^[k].lag1:= dummyi;
    dummyi:= SRs^[r2].lag2; SRs^[r2].lag2:= SRs^[k].lag2;
    SRs^[k].lag2:= dummyi;
    dummyi:= SRs^[r2].kKlass; SRs^[r2].kKlass:= SRs^[k].kKlass;
    SRs^[k].kKlass:= dummyi;
  end;
  (*-------------------------------------------------------------------------*)
  procedure Algoritm;
  Var
                zz2, x: integer;
  begin
    l2:= (1- j2); c:= y2;
    If (i2> 1) Then begin
      Hatten(1,i2,hsort);
      t:= (p2 Div i2);
      For zz2:= 1 To t Do begin
        For x:= 1 To i2 Do begin
          c:= y2+ hsort[x]-1;
          SlumpTal;
        end;
      end;
      Nolla;
    end else begin
      For zz2:= 1 To p2 Do SlumpTal;
      Nolla;
    end;
  end;
  (*-------------------------------------------------------------------------*)
begin
  y2:= 1; jj:= j5- j3; i2:= 0;
  inv:= ((j5- j3) Div KlassSet^[1].antal);
  While (y2<= nKlasser) Do begin
    r2:= j3; s2:= j3; Inc(i2);
    p2:= KlassSet^[y2].antal; z2:= y2;
    aaa:= true;
    While (z2< nKlasser) And aaa Do begin
      Inc(z2);
      If (KlassSet^[y2].antal <> KlassSet^[z2].antal) Then begin
        j:= jj; j2:= (j / p2);
        Algoritm; aaa:= false;
```

```
1021        end else begin
1022          p2:= p2+ KlassSet^[z2].antal;
1023          Inc(i2);
1024          aaa:= true;
1025        end;
1026      end;
1027      If aaa Then begin
1028        If (y2= nKlasser) And (p2= 1) Then begin
1029          j2:= 1;
1030          Algoritm;
1031        end else begin
1032          j:= jj; j2:= (j / p2);
1033          Algoritm;
1034        end;
1035      end;
1036    end;
1037    x:= Random(inv)+ 1;
1038    If (x> 1) And ((j5- j3)> 1) Then begin
1039      For k:= 1 To Pred(x) Do begin
1040        dumSRs:= SRs^[j5];
1041        For c:= j5 DownTo j4+ 1 Do SRs^[c]:= SRs^[c-1];
1042        SRs^[j4]:= dumSRs;
1043      end;
1044    end;
1045  end;
1046  (*-------------------------------------------------------------------------*)
1047  procedure MakeGameOrder(Aowner: TComponent);
1048  Const
1049    changetid: shortstring= 'Ändrtid';
1050    changeboss: shortstring= 'Ändraledare';
1051  Type
1052    Tgcheck= Array[0..mxGames] Of record
1053      LL1, LL2: integer;
1054    end;
1055  Var
1056    xF: text;
1057    lRec, mxNr: longint;
1058    ndx, tii, cc, ii, io, ix, kk, l1, l2, ww, yy: integer;
1059    antalklasser: integer;
1060    checkdatum, iss, fe: boolean;
1061    gsave, gcheck: ^Tgcheck;
1062    ss: shortstring;
1063    STlist: TObjectList;
1064    ST: TSpelTid;
1065    (*-----------------------------------------------------------------------*)
1066    procedure AddSRndx(fr, tow: integer);
1067    Var
1068      gRec, i: longint;
1069      x, y, yyy, k, z, jj, k0, k1, f: integer;
1070      doit, iss, fe, fileend: boolean;
1071      pp, ss: allstrng;
1072      tkey1, tkey2, ldate: Strng16;
1073      inF: text;
1074      tlen, tl, ndx: integer;
1075    begin
1076      If checkdatum Then begin
1077        Assign(inF,L^.MaskName); Reset(inF);
1078        checkdatum:= (IOresult= 0);
1079      end;
1080      FillChar(gcheck^,SizeOf(Tgcheck),#0);
1081      If (L^.notonsameday> 0) Then
1082        tlen:= 10
1083      else tlen:= 16;
1084      pp:= '';
1085      yy:= 0; z:= 0; ldate:= 'xyz';
1086      FixtSelectGame(ClearRtn,'',gRec,iss,fe); fileend:= false;
1087      y:= mxGamesPerTime; pp:= ''; yyy:= 0; ndx:= -1;
1088      For i:= fr To tow Do begin
```

```
1089        Inc(y);
1090        If (y>= L^.GamesPerTime) Then begin
1091          If (pp <> '') Then begin
1092            ST.Speltid.nG:= yyy; yyy:= 0;
1093          end;
1094          If (ndx>= 0) Then
1095            STlist.Items[ndx]:= ST;
1096          pp:= ''; Inc(yy);
1097          FixtSelectGame(NextRtn,'',gRec,iss,fe);
1098          If Not(iss) Or fileend Then begin
1099            ZoGame; fileend:= true;
1100            Game^.DateTid:= changetid+ NumbStr(yy,3);
1101            Game^.Chief:= changeboss+ NumbStr(yy,3);
1102          end;
1103          pp:= Game^.DateTid+ ','+ Game^.Chief;
1104          ST:= TSpelTid.Create;
1105          ST.Speltid.Inf:= pp;
1106          STlist.Add(ST);
1107          ndx:= STlist.IndexOf(ST);
1108          pp:= '';
1109          If (Pos(changetid,Game^.DateTid) <> 0) Then
1110            tl:= Length(Game^.DateTid)
1111          else tl:= tlen;
1112          If (ldate <> Copy(Game^.DateTid,1,tl)) Then begin
1113            FillChar(gcheck^,SizeOf(Tgcheck),#0);
1114            z:= 0;
1115            ldate:= Copy(Game^.DateTid,1,tl);
1116          end;
1117          y:= 0;
1118        end;
1119        With SRs^[i] Do begin
1120          Inc(L^.K[kKlass].nC);
1121          x:= L^.K[kKlass].nC;
1122          GetSerieOrder(L^.K[kKlass].nL,x,l1,l2);
1123          lag1:= SerieLnr^[kKlass][l1];
1124          lag2:= SerieLnr^[kKlass][l2];
1125          If true Then begin
1126            k1:= ww; k0:= 0; iss:= false;
1127            Repeat
1128              jj:= z; doit:= false;
1129              If true Then begin
1130                While (jj> 0) And Not(doit) Do begin
1131                  With gcheck^[jj] Do
1132                    doit:= (lag1= LL1) Or
1133                           (lag2= LL2) Or
1134                           (lag1= LL2) Or
1135                           (lag2= LL1);
1136                  Dec(jj);
1137                end;
1138              end;
1139              If checkdatum And Not(doit) Then begin
1140                If (Pos(changetid,ldate) = 0) Then begin
1141                  Reset(inF); checkdatum:= (IOresult= 0);
1142                  doit:= false;
1143                  While Not(EOF(inF)) And Not(doit) Do begin
1144                    ReadLn(inF,ss);
1145                    tkey1:= GetIxportKey(ss,',',1);
1146                    If (Length(tkey1)<= 5) Then begin
1147                      Val(tkey1,jj,f);
1148                      doit:= (lag1= jj) Or (lag2= jj);
1149                      If doit Then begin
1150                        tkey1:= GetIxportKey(ss,',',2);
1151                        f:= Length(tkey1);
1152                        tkey2:= EPad(Game^.DateTid,f);
1153                        For jj:= 1 To f Do
1154                          If (tkey1[jj]= '?') Then tkey2[jj]:= '?';
1155                        doit:= (Pos(tkey1,tkey2) <> 0);
1156                      end;
```

```
                  end;
                end;
              end;
            end;
            If doit Then begin
              If Not(iss) Then begin
                Inc(ww);
                With gsave^[ww] Do begin
                  LL1:= lag1;
                  LL2:= lag2;
                end;
              end;
              If true Then begin
                Repeat
                  Inc(k0);
                  If (k0<= k1) Then begin
                    With gsave^[k0] Do begin
                      lag1:= LL1;
                      lag2:= LL2;
                    end;
                  end;
                Until (k0> k1) Or ((lag1+ lag2)> 0);
                doit:= (k0<= k1);
                iss:= true;
              end;
            end else begin
              If iss Then begin
                If (k0<= k1) Then begin
                  With gsave^[k0] Do begin
                    LL1:= 0;
                    LL2:= 0;
                  end;
                end;
              end;
            end;
          Until Not(doit);
          If (k0> k1) Then begin
            lag1:= 0; lag2:= 0;
          end else begin
            Inc(z);
            With gcheck^[z] Do begin
              LL1:= lag1;
              LL2:= lag2;
            end;
          end;
        end;
        WriteCountW(8,'Utmatning: "'+ NumbStr(l1,4)+ '"'+
                      '-"'+ NumbStr(l2,4)+ '"'+
                      RightNumbStr(fr,4)+ '-['+
                      RightNumbStr(i,4)+ ']-'+
                      RightNumbStr(tow,4));
        If ((lag1+ lag2) <> 0) Then begin
          pp:= pp+ NumbStr(lag1,0)+ '-'+ NumbStr(lag2,0)+ ',';
          ST.Speltid.G[Succ(y)].hemma:= lag1;
          ST.Speltid.G[Succ(y)].borta:= lag2;
          Inc(yyy);
        end else Dec(y);
      end;
    end;
    If (ww> 0) Then begin
      i:= 0;
      Repeat
        Inc(y);
        If (y>= L^.GamesPerTime) Then begin
          If (pp <> '') Then begin
            ST.SpelTid.nG:= yyy; yyy:= 0;
          end;
          If (ndx>= 0) Then
```

```
1225              STlist.Items[ndx]:= ST;
1226            pp:= ''; Inc(yy);
1227            FixtSelectGame(NextRtn,'',gRec,iss,fe);
1228            If Not(iss) Or fileend Then begin
1229              ZoGame; fileend:= true;
1230              Game^.DateTid:= changetid+ NumbStr(yy,3);
1231              Game^.Chief:= changeboss+ NumbStr(yy,3);
1232            end;
1233            pp:= Game^.DateTid+ ','+ Game^.Chief;
1234            ST:= TSpelTid.Create;
1235            ST.Speltid.Inf:= pp;
1236            STlist.Add(ST);
1237            ndx:= STlist.IndexOf(ST);
1238            pp:= '';
1239            If (Pos(changetid,Game^.DateTid) <> 0) Then
1240              tl:= Length(Game^.DateTid)
1241            else tl:= tlen;
1242            If (ldate <> Copy(Game^.DateTid,1,tl)) Then begin
1243              FillChar(gcheck^,SizeOf(Tgcheck),#0);
1244              z:= 0; ldate:= Copy(Game^.DateTid,1,tl);
1245            end;
1246            y:= 0;
1247          end;
1248          Inc(i);
1249          With gsave^[i] Do begin
1250            l1:= LL1;
1251            l2:= LL2;
1252          end;
1253          doit:= false;
1254          If ((l1+ l2) <> 0) Then begin
1255            jj:= z;
1256            While (jj> 0) And Not(doit) Do begin
1257              With gcheck^[jj] Do
1258                doit:= (l1= LL1) Or
1259                       (l2= LL2) Or
1260                       (l1= LL2) Or
1261                       (l2= LL1);
1262              Dec(jj);
1263            end;
1264            If checkdatum And Not(doit) Then begin
1265              If (Pos(changetid,ldate) = 0) Then begin
1266                Reset(inF); checkdatum:= (IOresult= 0);
1267                While Not(EOF(inF)) And Not(doit) Do begin
1268                  ReadLn(inF,ss);
1269                  tkey1:= GetIxportKey(ss,',',1);
1270                  If (Length(tkey1)<= 5) Then begin
1271                    Val(tkey1,jj,f);
1272                    doit:= (l1= jj) Or (l2= jj);
1273                    If doit Then begin
1274                      tkey1:= GetIxportKey(ss,',',2);
1275                      f:= Length(tkey1);
1276                      tkey2:= EPad(Game^.DateTid,f);
1277                      For jj:= 1 To f Do
1278                        If (tkey1[jj]= '?') Then tkey2[jj]:= '?';
1279                      doit:= (Pos(tkey1,tkey2) <> 0);
1280                    end;
1281                  end;
1282                end;
1283              end;
1284            end;
1285            If Not(doit) Then begin
1286              Inc(z);
1287              With gcheck^[z] Do begin
1288                LL1:= l1;
1289                LL2:= l2;
1290              end;
1291            end else Dec(i);
1292            WriteCountW(8,'Utmatning: "'+ NumbStr(l1,4)+ '"'+
```

```
                        '-"'+ NumbStr(l2,4)+ '"'+
                        RightNumbStr(fr,4)+ '-['+
                        RightNumbStr(i,4)+ ']-'+
                        RightNumbStr(tow,4));
          If Not(doit) Then begin
            pp:= pp+ NumbStr(l1,0)+ '-'+ NumbStr(l2,0)+ ',';
            ST.Speltid.G[Succ(y)].hemma:= l1;
            ST.Speltid.G[Succ(y)].borta:= l2;
            Inc(yyy);
          end;
        end else Dec(y);
      Until (i>= ww);
    end;
    If (pp <> '') Then begin
      ST.SpelTid.nG:= yyy; yyy:= 0;
      pp:= '';
    end;
    If (ndx>= 0) Then
      STlist.Items[ndx]:= ST;
  end;
  (*---------------------------------------------------------------------------*)
  procedure TestaLottn;
  Var
    tlen, i, j, n, x, xx, u, y, p1, p2, a, bb, b, c: integer;
    found, done: boolean;
    ST1: TSpelTid;
    GG: TSpelTidRec;
    pp, ss, tt: allstrng;
  begin
    If (L^.notonsameday> 0) Then
      tlen:= 10
    else tlen:= 16;
    n:= Pred(STlist.Count);
    if (n>= 0) then begin
      i:= 0;
      Repeat
        ST:= TSpelTid(STlist.Items[i]);
        pp:= ST.Speltid.Inf;
        if (Pos(changetid,pp) <> 0) then begin
          x:= ST.Speltid.nG;
          Repeat
            if (x> 0) then begin
              p1:= ST.Speltid.G[x].hemma;
              p2:= ST.Speltid.G[x].borta;
              j:= 0; done:= false;
              Repeat
                ST1:= TSpelTid(STlist.Items[j]);
                ss:= Copy(ST1.Speltid.Inf,1,tlen);
                if (Pos(changetid,ss) = 0) then begin
                  a:= j;
                  Repeat
                    ST1:= TSpelTid(STlist.Items[a]);
                    tt:= Copy(ST1.Speltid.Inf,1,tlen);
                    found:= (ss= tt);
                    Inc(a);
                  Until Not(found) Or (a> n);
                  b:= Pred(a);
                  Dec(b);
                  a:= j; found:= true; xx:= L^.GamesPerTime;
                  bb:= 0;
                  Repeat
                    ST1:= TSpelTid(STlist.Items[a]);
                    u:= ST1.Speltid.nG;
                    if (u< L^.GamesPerTime) then begin
                      xx:= u;
                      If (bb = 0) Then bb:= a;
                    end;
                    y:= 0;
```

```
                    Repeat
                      Inc(y);
                      found:= found And (ST1.Speltid.G[y].hemma<> p1);
                      found:= found And (ST1.Speltid.G[y].hemma<> p2);
                      found:= found And (ST1.Speltid.G[y].borta<> p1);
                      found:= found And (ST1.Speltid.G[y].borta<> p2);
                    Until (y>= u) Or Not(found);
                    Inc(a);
                  Until Not(found) Or (a> b);
                  if found Then begin
                    if (xx< L^.GamesPerTime) then begin
                      ST1:= TSpelTid(STlist.Items[bb]);
                      u:= ST1.Speltid.nG;
                      Inc(u);
                      ST1.Speltid.G[u].hemma:= p1;
                      ST1.Speltid.G[u].borta:= p2;
                      ST1.Speltid.nG:= u;
                      STList.Items[bb]:= ST1;
                      ST.Speltid.G[x].hemma:= 0;
                      ST.Speltid.G[x].borta:= 0;
                      STList.Items[i]:= ST;
                      done:= true;
                    end;
                  end;
                  j:= b;
                end;
                Inc(j);
              Until (j> n) Or done;
            end;
            Dec(x);
          Until (x<= 0);
          x:= ST.Speltid.nG;
          FillChar(GG,SizeOf(TSpelTidRec),#0);
          a:= 0;
          for j:= 1 To x Do begin
            p1:= ST.Speltid.G[j].hemma;
            p2:= ST.Speltid.G[j].borta;
            if ((p1+ p2) > 0) then begin
              Inc(a);
              GG.G[a].hemma:= p1;
              GG.G[a].borta:= p2;
            end;
          end;
          ST.Speltid.G:= GG.G;
          ST.Speltid.nG:= a;
          STlist.Items[i]:= ST;
        end;
        Inc(i);
      Until (i> n);
    end;
  end;
  (*-------------------------------------------------------------------------*)
function TestaLottnB: boolean;
  Var
    tlen, i, j, n, x, u, y, p1, p2, a, b, c: integer;
    doit, found, done: boolean;
    ST1: TSpelTid;
    pp, ss, tt: allstrng;
  begin
    doit:= false;
    If (L^.notonsameday> 0) Then
      tlen:= 10
    else tlen:= 16;
    n:= Pred(STlist.Count);
    if (n>= 0) then begin
      i:= 0;
      Repeat
        ST:= TSpelTid(STlist.Items[i]);
```

```
1429          pp:= ST.Speltid.Inf;
1430          if (Pos(changetid,pp) <> 0) then begin
1431            doit:= true;
1432            x:= ST.Speltid.nG;
1433            Repeat
1434              if (x> 0) then begin
1435                p1:= ST.Speltid.G[x].hemma;
1436                p2:= ST.Speltid.G[x].borta;
1437                j:= 0; done:= false;
1438                Repeat
1439                  ST1:= TSpelTid(STlist.Items[j]);
1440                  ss:= Copy(ST1.Speltid.Inf,1,tlen);
1441                  if (Pos(changetid,ss) = 0) then begin
1442                    a:= j;
1443                    Repeat
1444                      ST1:= TSpelTid(STlist.Items[a]);
1445                      tt:= Copy(ST1.Speltid.Inf,1,tlen);
1446                      found:= (ss= tt);
1447                      Inc(a);
1448                    Until Not(found) Or (a> n);
1449                    b:= Pred(a);
1450                    Dec(b);
1451                    a:= j; found:= true;
1452                    Repeat
1453                      ST1:= TSpelTid(STlist.Items[a]);
1454                      u:= ST1.Speltid.nG;
1455                      y:= 0;
1456                      Repeat
1457                        Inc(y);
1458                        found:= found And (ST1.Speltid.G[y].hemma<> p1);
1459                        found:= found And (ST1.Speltid.G[y].hemma<> p2);
1460                        found:= found And (ST1.Speltid.G[y].borta<> p1);
1461                        found:= found And (ST1.Speltid.G[y].borta<> p2);
1462                      Until (y>= u) Or Not(found);
1463                      Inc(a);
1464                    Until Not(found) Or (a> b);
1465                    if found Then begin
1466                      ST1:= TSpelTid(STlist.Items[j]);
1467                      u:= ST1.Speltid.nG;
1468                      ST.Speltid.G[x].hemma:= ST1.Speltid.G[u].hemma;
1469                      ST.Speltid.G[x].borta:= ST1.Speltid.G[u].borta;
1470                      ST1.Speltid.G[u].hemma:= p1;
1471                      ST1.Speltid.G[u].borta:= p2;
1472                      STList.Items[j]:= ST1;
1473                      STList.Items[i]:= ST;
1474                      done:= true;
1475                    end;
1476                    j:= b;
1477                  end;
1478                  Inc(j);
1479                Until (j> n) Or done;
1480              end;
1481              Dec(x);
1482            Until (x<= 0);
1483          end;
1484          Inc(i);
1485        Until (i> n);
1486      end;
1487      TestaLottnB:= doit;
1488    end;
1489    (*-------------------------------------------------------------------------*)
1490    procedure UtmataTillFil(yy: integer);
1491    Var
1492      i, j, n: integer;
1493      pp, ss: allstrng;
1494      fr, tow: integer;
1495      inF: text;
1496    begin
```

```
1497      pp:= '; lottning av spelfördelning: bosse engborg, hellefors 2016-08-29';
1498      Writeln(xF,pp);
1499      pp:= '; '+ Info^.Arr+ ' '+ nDate(yymmdd)+ ' '+ Info^.Tvl;
1500      Writeln(xF,pp);
1501      pp:= '';
1502      Writeln(xF,'*Speldag');
1503      fr:= 0; tow:= Pred(STlist.Count);
1504      pp:= '';
1505      For i:= fr To tow Do begin
1506        ST:= TSpelTid(STlist.Items[i]);
1507        pp:= ST.Speltid.Inf;
1508        Writeln(xF,pp);
1509        pp:= '*!';
1510        n:= ST.Speltid.nG;
1511        j:= 1;
1512        while (j<= n) do begin
1513          pp:= pp+ NumbStr(ST.Speltid.G[j].hemma,0)+ '-';
1514          pp:= pp+ NumbStr(ST.Speltid.G[j].borta,0)+ ',';
1515          Inc(j);
1516        end;
1517        If (n> 0) Then Writeln(xF,pp);
1518      end;
1519      Writeln(xF,'*Lottad');
1520      Writeln(xF,'; Antal speltider '+ RightNumbStr(Succ(tow),3));
1521      Writeln(xF,'; Antal spel/tid '+ RightNumbStr(L^.GamesPerTime,3));
1522      Writeln(xF,'; Antal matcher '+ RightNumbStr(yy,3));
1523      If (L^.notonsameday> 0) Then
1524        Writeln(xF,'; endast en match per datum och lag');
1525      If (L^.dotheXX> 0) Then
1526        Writeln(xF,'; försöker fixa ändrtider');
1527      If (L^.dotheXX> 1) And (L^.dotheXX> 0) Then
1528        Writeln(xF,'; byter i '+ NumbStr(L^.dotheXX,0)+ ' steg och försöker fixa ändrtider');
1529      If checkdatum Then begin
1530        Assign(inF,L^.MaskName); Reset(inF);
1531        checkdatum:= (IOresult= 0);
1532      end;
1533      If checkdatum Then begin
1534        Writeln(xF,'; lagens ej spelbara speltillfällen');
1535        Reset(inF);
1536        While Not(EOF(inF)) Do begin
1537          ReadLn(inF,ss);
1538          Writeln(xF,'; '+ ss);
1539        end;
1540        Close(inF);
1541      end;
1542      Writeln(xF,'');
1543    end;
1544    (*-------------------------------------------------------------------------*)
1545  begin
1546    If Not(HeapMemOk(SizeOf(Tgcheck)* 2)) Then exit;
1547    If Not(SetLottRec(Aowner)) Then exit;
1548    New(gcheck); New(gsave);
1549    FillChar(gsave^,SizeOf(Tgcheck),#0); ww:= 0;
1550    checkdatum:= (L^.Maskname <> '') And EExistFile(L^.Maskname);
1551    Assign(xF,L^.Flottname); Rewrite(xF); io:= IOresult;
1552    If (io <> 0) Then begin
1553      MyMsg:= 'Kan inte öppna filen "'+ L^.Flottname+ '" för skrivning';
1554      If ErrorMsg(1) Then exit;
1555    end;
1556    STList:= TObjectList.Create(true);
1557    tii:= pushTimer;
1558    RandoMize;
1559    ZoMask; MaskList:= Ml; MaskList.ClsRec:= mxClass+1;
1560    MakeCountW(Aowner,8,'Spelfördelar till "'+ L^.FlottName+ '"');
1561    FillChar(SRs^,SizeOf(SRs^),#0);
1562    FillChar(SerieLnr^,SizeOf(SerieLnrType),#0);
1563    cc:= 0; ix:= 0;
1564    Repeat
```

```
1565          Inc(cc);
1566      WriteCountW(2,'Klasser antal:'+ RightNumbStr(cc,2)+
1567                    ' ['+ RightNumbStr(L^.nKlasser,2)+ ']');
1568      With L^.K[cc] Do begin
1569        Ml.SortO:= SSnr; Ml.ClsRec:= rec;
1570        SRMakeList(Aowner,Ml,mxNr);
1571        If (SRrecs^[0]> 1) Then begin
1572          If (SRrecs^[0]<= mxInClass) Then begin
1573            kk:= 0;
1574            Repeat
1575              Inc(kk); lRec:= SRrecs^[kk];
1576              FixtSelectSR(RcRtn,'',lRec,iss,fe);
1577              SerieLnr^[cc][kk]:= SR^.iSnr;
1578              WriteCountW(3,'"'+ SR^.SNr+ '" Antal lag i klassen:'+
1579                            RightNumbStr(kk,4)+
1580                            ' ['+ RightNumbStr(SRrecs^[0],4)+ ']');
1581            Until (kk>= SRrecs^[0]);
1582            kk:= 0;
1583            Repeat
1584              Inc(kk); Inc(ix);
1585              WriteCountW(4,'Matcher i klassen antal:'+
1586                            RightNumbStr(kk,4)+
1587                            ' ['+ RightNumbStr(nG,4)+ ']');
1588              WriteCountW(5,'Matcher totalt antal:'+
1589                            RightNumbStr(ix,4)+
1590                            ' ['+ RightNumbStr(L^.nGames,4)+ ']');
1591              GetSerieOrder(nL,kk,l1,l2);
1592              WriteCountW(6,'Matchmöte: "'+ NumbStr(SerieLnr^[cc][l1],4)+
1593                            '"-"'+ NumbStr(SerieLnr^[cc][l2],4));
1594              With SRs^[ix] Do begin
1595                kKlass:= cc;
1596                lag1:= SerieLnr^[cc][l1];
1597                lag2:= SerieLnr^[cc][l2];
1598              end;
1599            Until (kk>= nG);
1600          end else begin
1601            lRec:= SRrecs^[1];
1602            FixtSelectSR(RcRtn,'',lRec,iss,fe);
1603            ErrorMsgX(201,'antalet i klassen "'+
1604                      Klass^.Bet+ '" överskrider maxantalet '+
1605                      RightNumbStr(mxInclass,3));
1606          end;
1607        end;
1608      end;
1609    Until (cc>= L^.nKlasser);
1610    mxNr:= ix;
1611    For ii:= 1 To mxNr Do SRs^[ii].ee:= ii;
1612    If (mxNr> 0) Then begin
1613      If (mxNr< 2) Then begin
1614        AddSRndx(1,mxNr);
1615          end else begin
1616                        SetKlasser(1,mxNr,antalklasser);
1617                   Lottning(0,1,mxNr,antalklasser);
1618                        AddSRndx(1,mxNr);
1619      end;
1620      if (L^.DoTheXX> 0) Then begin
1621        TestaLottn;
1622        if (L^.dotheXX> 1) then begin
1623          for ii:= 2 To L^.dotheXX Do begin
1624            if TestaLottnB then
1625              TestaLottn;
1626          end;
1627        end;
1628      end;
1629      UtMataTillFil(mxNr);
1630    end;
1631    RestoreCountW;
1632    ReadClockDate;
```

```
1633    STlist.Free;
1634    Close(xF); io:= IOresult;
1635    MyMsg:= 'Lottningsarbete tid (mmm:ss) '+ popTimer(mmmss);
1636    If ErrorMsg(1) Then;
1637    If (io <> 0) Then begin
1638      MyMsg:= 'Fel vid stänging av filen "'+ L^.Flottname+ '"';
1639      If ErrorMsg(1) Then;
1640    end;
1641    Dispose(gsave); Dispose(gcheck);
1642    SRrecen[1]:= 0;
1643  end;
1644  (*-------------------------------------------------------------------------*)
1645  procedure MakeGameOrderX(Aowner: TComponent);
1646  Const
1647   {$ifdef windows}
1648    changetid: shortstring= 'Ändrtid';
1649    changeboss: shortstring= 'Ändraledare';
1650   {$else}
1651    changetid: string= 'Žndrtid';
1652    changeboss: string= 'Žndraledare';
1653   {$endif}
1654  Type
1655    Tgcheck= Array[0..mxGames] Of record
1656      LL1, LL2: integer;
1657    end;
1658  Var
1659    xF: text;
1660    lRec, mxNr: longint;
1661    tii, cc, ii, io, ix, kk, l1, l2, ww: integer;
1662    antalklasser: integer;
1663    checkdatum, iss, fe: boolean;
1664    gsave, gcheck: ^Tgcheck;
1665    ss: shortstring;
1666    (*-----------------------------------------------------------------------*)
1667    procedure AddSRndx(fr, tow: integer);
1668    Var
1669      gRec, i: longint;
1670      x, yy, y, k, z, jj, k0, k1, f: integer;
1671      doit, iss, fe, fileend: boolean;
1672      pp, ss: allstrng;
1673      tkey1, tkey2, ldate: Strng16;
1674      inF: text;
1675      tlen, tl: integer;
1676    begin
1677      If checkdatum Then begin
1678        Assign(inF,L^.MaskName); Reset(inF);
1679        checkdatum:= (IOresult= 0);
1680      end;
1681      FillChar(gcheck^,SizeOf(Tgcheck),#0);
1682      If (L^.notonsameday> 0) Then
1683        tlen:= 10
1684      else tlen:= 16;
1685      pp:= '; '+ Info^.Arr+ ' '+ nDate(yymmdd)+ ' '+ Info^.Tvl;
1686     {$ifdef windows}
1687      pp:= AnsiToOEMstr(pp);
1688     {$endif}
1689      Writeln(xF,pp);
1690      pp:= '';
1691      Writeln(xF,'*Speldag');
1692      yy:= 0; z:= 0; ldate:= 'xyz';
1693      FixtSelectGame(ClearRtn,'',gRec,iss,fe); fileend:= false;
1694      y:= mxGamesPerTime; pp:= '';
1695      For i:= fr To tow Do begin
1696        Inc(y);
1697        If (y>= L^.GamesPerTime) Then begin
1698         {$ifdef windows}
1699          pp:= AnsiToOEMstr(pp);
1700         {$endif}
```

```
1701          If (pp <> '') Then Writeln(xF,'*!'+ pp);
1702          pp:= ''; Inc(yy);
1703          FixtSelectGame(NextRtn,'',gRec,iss,fe);
1704          If Not(iss) Or fileend Then begin
1705            ZoGame; fileend:= true;
1706            Game^.DateTid:= changetid+ NumbStr(yy,3);
1707            Game^.Chief:= changeboss+ NumbStr(yy,3);
1708          end;
1709          pp:= Game^.DateTid+ ','+ Game^.Chief;
1710         {$ifdef windows}
1711          pp:= AnsiToOEMstr(pp);
1712         {$endif}
1713          Writeln(xF,pp);
1714          pp:= '';
1715          If (Pos(changetid,Game^.DateTid) <> 0) Then
1716            tl:= Length(Game^.DateTid)
1717          else tl:= tlen;
1718          If (ldate <> Copy(Game^.DateTid,1,tl)) Then begin
1719            FillChar(gcheck^,SizeOf(Tgcheck),#0);
1720            z:= 0;
1721            ldate:= Copy(Game^.DateTid,1,tl);
1722          end;
1723          y:= 0;
1724        end;
1725        With SRs^[i] Do begin
1726          Inc(L^.K[kKlass].nC);
1727          x:= L^.K[kKlass].nC;
1728          GetSerieOrder(L^.K[kKlass].nL,x,l1,l2);
1729          lag1:= SerieLnr^[kKlass][l1];
1730          lag2:= SerieLnr^[kKlass][l2];
1731          If true Then begin
1732            k1:= ww; k0:= 0; iss:= false;
1733            Repeat
1734              jj:= z; doit:= false;
1735              If true Then begin
1736                While (jj> 0) And Not(doit) Do begin
1737                  With gcheck^[jj] Do
1738                    doit:= (lag1= LL1) Or
1739                           (lag2= LL2) Or
1740                           (lag1= LL2) Or
1741                           (lag2= LL1);
1742                  Dec(jj);
1743                end;
1744              end;
1745              If checkdatum And Not(doit) Then begin
1746                If (Pos(changetid,ldate) = 0) Then begin
1747                  Reset(inF); checkdatum:= (IOresult= 0);
1748                  doit:= false;
1749                  While Not(EOF(inF)) And Not(doit) Do begin
1750                    ReadLn(inF,ss);
1751                    tkey1:= GetIxportKey(ss,',',1);
1752                    If (Length(tkey1)<= 5) Then begin
1753                      Val(tkey1,jj,f);
1754                      doit:= (lag1= jj) Or (lag2= jj);
1755                      If doit Then begin
1756                        tkey1:= GetIxportKey(ss,',',2);
1757                        f:= Length(tkey1);
1758                        tkey2:= EPad(Game^.DateTid,f);
1759                        For jj:= 1 To f Do
1760                          If (tkey1[jj]= '?') Then tkey2[jj]:= '?';
1761                        doit:= (Pos(tkey1,tkey2) <> 0);
1762                      end;
1763                    end;
1764                  end;
1765                end;
1766              end;
1767              If doit Then begin
1768                If Not(iss) Then begin
```

```
1769                Inc(ww);
1770                With gsave^[ww] Do begin
1771                  LL1:= lag1;
1772                  LL2:= lag2;
1773                end;
1774              end;
1775              If true Then begin
1776                Repeat
1777                  Inc(k0);
1778                  If (k0<= k1) Then begin
1779                    With gsave^[k0] Do begin
1780                      lag1:= LL1;
1781                      lag2:= LL2;
1782                    end;
1783                  end;
1784                Until (k0> k1) Or ((lag1+ lag2)> 0);
1785                doit:= (k0<= k1);
1786                iss:= true;
1787              end;
1788            end else begin
1789              If iss Then begin
1790                If (k0<= k1) Then begin
1791                  With gsave^[k0] Do begin
1792                    LL1:= 0;
1793                    LL2:= 0;
1794                  end;
1795                end;
1796              end;
1797            end;
1798          Until Not(doit);
1799          If (k0> k1) Then begin
1800            lag1:= 0; lag2:= 0;
1801          end else begin
1802            Inc(z);
1803            With gcheck^[z] Do begin
1804              LL1:= lag1;
1805              LL2:= lag2;
1806            end;
1807          end;
1808        end;
1809        WriteCountW(8,'Utmatning: "'+ NumbStr(l1,4)+ '"'+
1810                      '-"'+ NumbStr(l2,4)+ '"'+
1811                      RightNumbStr(fr,4)+ '-['+
1812                      RightNumbStr(i,4)+ ']-'+
1813                      RightNumbStr(tow,4));
1814        If (Length(pp)> 60) Then begin
1815         {$ifdef windows}
1816          pp:= AnsiToOEMstr(pp);
1817         {$endif}
1818          Writeln(xF,'*!'+ pp);
1819          pp:= '';
1820        end;
1821        If ((lag1+ lag2) <> 0) Then
1822          pp:= pp+ NumbStr(lag1,0)+ '-'+ NumbStr(lag2,0)+ ','
1823        else Dec(y);
1824      end;
1825    end;
1826    If (ww> 0) Then begin
1827      i:= 0;
1828      Repeat
1829        Inc(y);
1830        If (y>= L^.GamesPerTime) Then begin
1831         {$ifdef windows}
1832          pp:= AnsiToOEMstr(pp);
1833         {$endif}
1834          If (pp <> '') Then Writeln(xF,'*!'+ pp);
1835          pp:= ''; Inc(yy);
1836          FixtSelectGame(NextRtn,'',gRec,iss,fe);
```

```
1837            If Not(iss) Or fileend Then begin
1838              ZoGame; fileend:= true;
1839              Game^.DateTid:= changetid+ NumbStr(yy,3);
1840              Game^.Chief:= changeboss+ NumbStr(yy,3);
1841            end;
1842            pp:= Game^.DateTid+ ','+ Game^.Chief;
1843           {$ifdef windows}
1844            pp:= AnsiToOEMstr(pp);
1845           {$endif}
1846            Writeln(xF,pp);
1847            pp:= '';
1848            If (Pos(changetid,Game^.DateTid) <> 0) Then
1849              tl:= Length(Game^.DateTid)
1850            else tl:= tlen;
1851            If (ldate <> Copy(Game^.DateTid,1,tl)) Then begin
1852              FillChar(gcheck^,SizeOf(Tgcheck),#0);
1853              z:= 0; ldate:= Copy(Game^.DateTid,1,tl);
1854            end;
1855            y:= 0;
1856          end;
1857          Inc(i);
1858          With gsave^[i] Do begin
1859            l1:= LL1;
1860            l2:= LL2;
1861          end;
1862          doit:= false;
1863          If ((l1+ l2) <> 0) Then begin
1864            jj:= z;
1865            While (jj> 0) And Not(doit) Do begin
1866              With gcheck^[jj] Do
1867                doit:= (l1= LL1) Or
1868                       (l2= LL2) Or
1869                       (l1= LL2) Or
1870                       (l2= LL1);
1871              Dec(jj);
1872            end;
1873            If checkdatum And Not(doit) Then begin
1874              If (Pos(changetid,ldate) = 0) Then begin
1875                Reset(inF); checkdatum:= (IOresult= 0);
1876                While Not(EOF(inF)) And Not(doit) Do begin
1877                  ReadLn(inF,ss);
1878                  tkey1:= GetIxportKey(ss,',',1);
1879                  If (Length(tkey1)<= 5) Then begin
1880                    Val(tkey1,jj,f);
1881                    doit:= (l1= jj) Or (l2= jj);
1882                    If doit Then begin
1883                      tkey1:= GetIxportKey(ss,',',2);
1884                      f:= Length(tkey1);
1885                      tkey2:= EPad(Game^.DateTid,f);
1886                      For jj:= 1 To f Do
1887                        If (tkey1[jj]= '?') Then tkey2[jj]:= '?';
1888                      doit:= (Pos(tkey1,tkey2) <> 0);
1889                    end;
1890                  end;
1891                end;
1892              end;
1893            end;
1894            If Not(doit) Then begin
1895              Inc(z);
1896              With gcheck^[z] Do begin
1897                LL1:= l1;
1898                LL2:= l2;
1899              end;
1900            end else Dec(i);
1901            WriteCountW(8,'Utmatning: "'+ NumbStr(l1,4)+ '"'+
1902                          '-"'+ NumbStr(l2,4)+ '"'+
1903                          RightNumbStr(fr,4)+ '-['+
1904                          RightNumbStr(i,4)+ ']-'+
```

```
                          RightNumbStr(tow,4));
           If (Length(pp)> 60) Then begin
            {$ifdef windows}
             pp:= AnsiToOEMstr(pp);
            {$endif}
             Writeln(xF,'*!'+ pp);
             pp:= '';
           end;
           If Not(doit) Then
             pp:= pp+ NumbStr(l1,0)+ '-'+ NumbStr(l2,0)+ ','
         end else Dec(y);
       Until (i>= ww);
     end;
     If (pp <> '') Then begin
      {$ifdef windows}
       pp:= AnsiToOEMstr(pp);
      {$endif}
       Writeln(xF,'*!'+ pp);
       pp:= '';
     end;
     Writeln(xF,'*Lottad');
     Writeln(xF,'; Antal speltider '+ RightNumbStr(yy,3));
     Writeln(xF,'; Antal spel/tid '+ RightNumbStr(L^.GamesPerTime,3));
     Writeln(xF,'; Antal matcher '+ RightNumbStr(tow,3));
     If (L^.notonsameday> 0) Then
       Writeln(xF,'; endast en match per datum och lag');
     If checkdatum Then begin
       Writeln(xF,'; lagens ej spelbara speltillfällen');
       Reset(inF);
       While Not(EOF(inF)) Do begin
         ReadLn(inF,ss);
         Writeln(xF,'; '+ ss);
       end;
       Close(inF);
     end;
   end;
   (*---------------------------------------------------------------------------*)
begin
   If Not(HeapMemOk(SizeOf(Tgcheck)* 2)) Then exit;
   If Not(SetLottRec(Aowner)) Then exit;
   New(gcheck); New(gsave);
   FillChar(gsave^,SizeOf(Tgcheck),#0); ww:= 0;
   checkdatum:= (L^.Maskname <> '') And EExistFile(L^.Maskname);
   Assign(xF,L^.Flottname); Rewrite(xF); io:= IOresult;
   If (io <> 0) Then begin
     MyMsg:= 'Kan inte öppna filen "'+ L^.Flottname+ '" för skrivning';
     If ErrorMsg(1) Then exit;
   end;
   tii:= pushTimer;
   RandoMize;
   ZoMask; MaskList:= Ml; MaskList.ClsRec:= mxClass+1;
  {$ifdef windows}
   MakeCountW(Aowner,8,'Spelfördelar till "'+ L^.FlottName+ '"');
  {$else}
   MakeCountW(8,'Spelf”rdelar till "'+ L^.FlottName+ '"');
  {$endif}
   FillChar(SRs^,SizeOf(SRs^),#0);
   FillChar(SerieLnr^,SizeOf(SerieLnrType),#0);
   cc:= 0; ix:= 0;
   Repeat
           Inc(cc);
     WriteCountW(2,'Klasser antal:'+ RightNumbStr(cc,2)+
                   ' ['+ RightNumbStr(L^.nKlasser,2)+ ']');
     With L^.K[cc] Do begin
       Ml.SortO:= SSnr; Ml.ClsRec:= rec;
       SRMakeList(Aowner,Ml,mxNr);
       If (SRrecs^[0]> 1) Then begin
         If (SRrecs^[0]<= mxInClass) Then begin
```

```
1973            kk:= 0;
1974            Repeat
1975              Inc(kk); lRec:= SRrecs^[kk];
1976              FixtSelectSR(RcRtn,'',lRec,iss,fe);
1977              SerieLnr^[cc][kk]:= SR^.iSnr;
1978              WriteCountW(3,'"'+ SR^.SNr+ '" Antal lag i klassen:'+
1979                          RightNumbStr(kk,4)+
1980                          ' ['+ RightNumbStr(SRrecs^[0],4)+ ']');
1981            Until (kk>= SRrecs^[0]);
1982            kk:= 0;
1983            Repeat
1984              Inc(kk); Inc(ix);
1985              WriteCountW(4,'Matcher i klassen antal:'+
1986                          RightNumbStr(kk,4)+
1987                          ' ['+ RightNumbStr(nG,4)+ ']');
1988              WriteCountW(5,'Matcher totalt antal:'+
1989                          RightNumbStr(ix,4)+
1990                          ' ['+ RightNumbStr(L^.nGames,4)+ ']');
1991              GetSerieOrder(nL,kk,l1,l2);
1992              WriteCountW(6,'Matchmöte: "'+ NumbStr(SerieLnr^[cc][l1],4)+
1993                          '"-"'+ NumbStr(SerieLnr^[cc][l2],4));
1994              With SRs^[ix] Do begin
1995                kKlass:= cc;
1996                lag1:= SerieLnr^[cc][l1];
1997                lag2:= SerieLnr^[cc][l2];
1998              end;
1999            Until (kk>= nG);
2000          end else begin
2001            lRec:= SRrecs^[1];
2002            FixtSelectSR(RcRtn,'',lRec,iss,fe);
2003            ErrorMsgX(201,'antalet i klassen "'+
2004                      Klass^.Bet+ '" överskrider maxantalet '+
2005                      RightNumbStr(mxInclass,3));
2006          end;
2007        end;
2008      end;
2009    Until (cc>= L^.nKlasser);
2010    mxNr:= ix;
2011    For ii:= 1 To mxNr Do SRs^[ii].ee:= ii;
2012    If (mxNr> 0) Then begin
2013      If (mxNr< 2) Then begin
2014        AddSRndx(1,mxNr);
2015            end else begin
2016                          SetKlasser(1,mxNr,antalklasser);
2017                    Lottning(0,1,mxNr,antalklasser);
2018                          AddSRndx(1,mxNr);
2019      end;
2020    end;
2021    RestoreCountW;
2022    ReadClockDate;
2023    Close(xF); io:= IOresult;
2024    MyMsg:= 'Lottningsarbete tid (mmm:ss) '+ popTimer(mmmss);
2025    If ErrorMsg(1) Then;
2026    If (io <> 0) Then begin
2027      MyMsg:= 'Fel vid stänging av filen "'+ L^.Flottname+ '"';
2028      If ErrorMsg(1) Then;
2029    end;
2030    Dispose(gsave); Dispose(gcheck);
2031    SRrecen[1]:= 0;
2032  end;
2033  (*-------------------------------------------------------------------------*)
2034  procedure MakeTeamsGame(Aowner: TComponent);
2035  Var
2036    nn, ii, jj, kk, tii: integer;
2037    kRec, mxNr, pRec, li: longint;
2038    iss, fe, Exst, FileEnd, up, done: boolean;
2039  begin
2040    If Info^.Lottad Then begin
```

```
2041      MyMsg:= 'Lottning redan utförd! Lotta om ';
2042      If Not(ErrorMsg(101)) Then exit;
2043    end;
2044    tii:= pushTimer;
2045    RandoMize;
2046    Info^.Lottad:= true;
2047   {$ifdef windows}
2048    MakeCountW(Aowner,5,'Registrerar matcher för varje lag!');
2049   {$else}
2050    MakeCountW(5,'Registrerar matcher för varje lag!');
2051   {$endif}
2052    FixtSelectKlass(ClearRtn,'',kRec,Exst,FileEnd);
2053    Repeat
2054      FixtSelectKlass(NextRtn,'',kRec,Exst,FileEnd);
2055      If Exst Then begin
2056        WriteCountW(2,'Klass^ "'+ Klass^.Bet+ '"'+ RightNumbStr(kRec,4));
2057        Ml.ClsRec:= kRec; Ml.SortO:= SSnr;
2058        SRMakeList(Aowner,Ml,mxNr);
2059        If (SRRecs^[0]> 0) Then begin
2060          kk:= 0;
2061          Repeat
2062            Inc(kk); pRec:= SRRecs^[kk];
2063            FixtSelectSR(RcRtn,'',pRec,iss,fe);
2064            WriteCountW(3,RightNumbStr(pRec,3)+ ' ['+
2065                          RightNumbStr(SRRecs^[0],3)+ ']'+
2066                          ' Lag "'+ SR^.Klubb+ '"');
2067            nn:= 0; jj:= 0;
2068            Repeat
2069              Inc(jj); up:= Odd(jj);
2070              If up Then
2071                ii:= 0
2072              else ii:= Succ(SRRecs^[0]);
2073              Repeat
2074                If up Then
2075                  Inc(ii)
2076                else Dec(ii);
2077                li:= SRrecs^[ii];
2078                If (li <> prec) Then begin
2079                  Inc(nn);
2080                  With SR^.Encounter[nn] Do begin
2081                    If up Then
2082                      lagr:= li
2083                    else lagr:= -li;
2084                  end;
2085                end;
2086                WriteCountW(4,'Matcher:'+ RightNumbStr(nn,3));
2087                If up Then
2088                  done:= (ii>= SRrecs^[0])
2089                else done:= (ii<= 1);
2090              Until done;
2091            Until (jj>= Klass^.roundCount);
2092            SR^.Encounter[0].lagr:= nn;
2093            PutRecSelectSR(pRec);
2094          Until (kk>= SRrecs^[0]);
2095        end;
2096      end;
2097    Until FileEnd;
2098    RestoreCountW;
2099    ReadClockDate;
2100    MyMsg:= 'Lottningsarbete tid (mmm:ss) '+ popTimer(mmmss);
2101  end;
2102  (*-------------------------------------------------------------------------*)
2103   {$ifdef debug}
2104  begin
2105    Write('SRLOTTN   ');
2106   {$endif}
2107  end.
2108
```