```
1: //*****************************************************************************
2: PROGRAM Tower_of_Hanoi_Steps_SolutionSequence;
3: {*****************************************************************************
4:  solution of the tower of hanoi with a codelist, double recursion!,
5:  codelist with 12 patterns of 4 steps, codesequence in 24 solution steps!
6:   loc's= 110, ex. _80ff
7:  try to program it with a canvas object to visualize
8:  ***************************************************************************** }
9:
10: Const SOLUTIONFILE = 'hanoilist.txt';
11:
12: Type
13:     TPatterns=  array[1..12] of shortstring;
14:
15: var answer: string;
16:     steps, step4, i: integer;
17:     patt: shortstring;
18:     varray: TPatterns;
19:     pattlst, seqlist: TStringlist;
20:
21:
22: procedure initPatternArray;    //codelist
23: begin
24:   varray[1]:=   'a to b a to c b to c a to b ';
25:   varray[2]:=   'a to b a to c b to c b to a ';
26:   varray[3]:=   'a to c a to b c to b a to c ';
27:   varray[4]:=   'a to c a to b c to b c to a ';
28:   varray[5]:=   'b to a b to c a to c b to a ';
29:   varray[6]:=   'b to a b to c a to c a to b ';
30:   varray[7]:=   'b to c b to a c to a b to c ';
31:   varray[8]:=   'b to c b to a c to a c to b ';
32:   varray[9]:=   'c to a c to b a to b c to a ';
33:   varray[10]:=  'c to a c to b a to b a to c ';
34:   varray[11]:=  'c to b c to a b to a c to b ';
35:   varray[12]:=  'c to b c to a b to a b to c ';
36: end;
37:
38: procedure Search_Write_Codes;
39: var vs, tmp: shortstring;
40:    i,k, found: integer;
41: begin
42:   found:= 0;
43:   writeln('pattern codes ---------- for solution: '+answer);
44:   for i:= 0 to pattlst.count - 1 do begin
45:     vs:= pattlst.strings[i]
46:     for k:= 1 to high(varray) do
47:       if vs = varray[k] then begin
48:         inc(found)
49:         tmp:= tmp +(inttostr(k)+'-')
50:         //write(inttostr(k)+'-');              //-fast
51:         //if found mod 32 = 0 then writeln(''); //-fast
52:       if found mod 24 = 0 then begin
53:         seqlist.add(tmp);
54:         tmp:= '';
55:       end;
56:     end;
57:   end;
58:   seqlist.add(tmp) //last/first segment
59:   writeln('Nr of codes: ' +inttostr(found))
60: end;
61:
62: procedure move(high: integer; a,c,b: char);
63: begin
64:   if high > 1 then begin
65:     move(high-1,a,b,c);
66:     //writeln(a+' to '+c);   //-fast
67:     inc(step4)
68:     patt:= patt+a+' to '+c+' ';
69:     if step4 mod 4 = 0 then begin
70:       pattlst.add(patt)
71:       patt:= '';
72:     end;
73:     move(high-1,b,c,a);
74:     inc(steps)
75:   end else begin
76:     //writeln(a+' to '+c)   //-fast
77:     inc(step4)
78:     patt:= patt+a +' to '+c+' ';
79:     if step4 mod 4 = 0 then begin
80:       pattlst.add(patt)
81:       patt:= '';
82:     end;
83:     inc(steps)
84:   end;
85: end;
86:
```

```
 87:
 88: begin     //main
 89:   steps:= 0;
 90:   step4:= 0;
 91:   initPatternArray;
 92:   pattlst:= TStringlist.create;
 93:   seqlist:= TStringlist.create;
 94:   answer:= readln('How much on a pile ?');
 95:   Writeln('Pile solution of: '+(answer))
 96:   move(strtoInt(answer),'a','b','c');
 97:   Writeln('had total '+inttoStr(steps)+ ' steps');
 98:   for i:= 0 to pattlst.count - 1 do      //-fast
 99:     writeln(pattlst.strings[i]);
100:   Search_Write_Codes;
101:   writeln('Nr of codelines: '+inttostr(seqlist.count)+ ' in file '+SOLUTIONFILE)
102:   seqlist.savetoFile(exepath+'examples\'+SOLUTIONFILE)
103:   seqlist.Free;
104:   pattlst.Free;
105:
106:   {Writeln('or '+chr(bintoint(inttobin(ord('A') OR ord('B')))))
107:   Writeln('xor '+chr(ord('A') XOR ord('B')))
108:   Writeln('and '+chr(ord('A') AND ord('B')))
109:   Writeln('not and'+chr((NOT ord('A') AND ord('B'))))}
110: End.
111:
112: The 12 Main Step Patterns!
113:
114: --1-------2-------3-------4-------  A 1,2,3,4
115: a to b  a to b   a to c  a to c
116: a to c  a to c   a to b  a to b
117: b to c  b to c   c to b  c to b
118: a to b  b to a   a to c  c to a
119: --5-------6--------7-------8------  B 5,6,7,8
120: b to a  b to a   b to c  b to c
121: b to c  b to c   b to a  b to a
122: a to c  a to c   c to a  c to a
123: b to a  a to b   b to c  c to b
124: --9-------10-------11------12-----  C 9,10,11,12
125: c to a  c to a   c to b  c to b
126: c to b  c to b   c to a  c to a
127: a to b  a to b   b to a  b to a
128: c to a  a to c   c to b  b to c
129: ---------------------------------
130:
131:
132: //************************************************************************
133: This is the solution code sequence for all even piles! repeat n/24:
134:
135: 3-6-11-3-5-12-3-(5/6)-11-4-5-11-3-6-11-(3/4)-5-12-3-5-11-4-5-(11/12)
136:
137: This is the solution code sequence for all odd piles!:
138:
139: 1-10-7-1-9-8-1-(9/10)-7-2-9-7-1-10-7-(1/2)-9-8-1-9-7-2-9-(7/8)
140: //************************************************************************
141: ------------------------------------------------------------------------------
142:
143:
144: Example for 12 piles(n) 2^12-1 Tester repeat n/24 lines
145:
146:                       X                  X                  X
147:   12 3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
148:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
149:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
150:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
151:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
152:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
153:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
154:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
155:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
156:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
157:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
158:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
159:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
160:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
161:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
162:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
163:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
164:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
165:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
166:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
167:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
168:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
169:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
170:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
171:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
172:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
```

```
173:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
174:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
175:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
176:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
177:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
178:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
179:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
180:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
181:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
182:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
183:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
184:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
185:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-12-
186:      3-6-11-3-5-12-3-5-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
187:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-3-5-12-3-5-11-4-5-12-
188:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-4-5-12-3-5-11-4-5-11-
189:      3-6-11-3-5-12-3-6-11-4-5-11-3-6-11-
190:
191:  for 10 is ((15*17=255*4=1020+3=2^10-1))
192:
193:
```