curved bow and negative

Written for and requested by Anton Lukoszevieze, this piece was originally conceived to be played on cello using a curved bow such that all strings of the cello are bowed / sounding throughout. In the score, each line of the staff represents one of the cello strings. A filled diamond notehead indicates that the respective string be played open. An unfilled diamond notehead indicates that the string be slightly touched resulting in either string noise or a harmonic, the selection of which should be the same for all strings and remain the same throughout the piece. If harmonics are chosen, the harmonic number should also be the same for all strings and remain the same throughout the piece (for example, the 3rd harmonic of every string). Except for the beginning of each system (see below), noteheads are only given when there is a change. That is, a given position is assumed to be held for a duration equivalent to the space until the next notehead such that each string is always in one of 2 states (as mentioned above: open or slightly touched).

The score is written in a proportional notation with a quantization scheme of 16 divisions per measure with 4 measures per system and 8 systems per page (note that only the systems are numbered). While the tempo is open, the piece was conceived to be played quite fast. Initial auditions were made such that each system lasted 8 seconds which results in a realization of 34 minutes 8 seconds (or 2048 seconds) if all measures are played.

However, for a shorter realization, the performer may choose to play only 2 or 3 measures of each system such that the number of measures per system remains the same throughout the piece. The gray noteheads at the beginning of each system are cautionary reiterations of the position at the end of the previous system if all 4 measures are interpreted. When 2 or 3 measures of each system are played, the position at the end of the of the previous system may differ than the position indicated by the gray noteheads at the start of the following system. For such cases, the performer must change the position at some point between the interpretation of the last notehead in the previous system and the start of the following system.

Without a curved bow, the piece can be performed by two stringed instruments (each playing two strings) by splitting the staff in half such that one player reads the upper two lines and the other player reads the lower 2. Or, by 4 stringed instruments (each playing one string) such that each player reads one line of the staff.

The general concept of the piece—that the two different noteheads each indicate one of two binary states / sounds—can be further abstracted by using other instruments / sounds resulting in a variety of potential realizations. However, similar to the original concept for cello, the difference between the states / sounds and among the four parts (represented by the lines of the staff) should be subtle.

A new score may also be generated for a unique version of the piece. Instructions are provided after the included score.

Clear and present with a constant dynamic throughout; perhaps amplified; perhaps with a computer program that tracks the state of each string and generates simple, very soft sounds that vary / change subtly based on the given state.

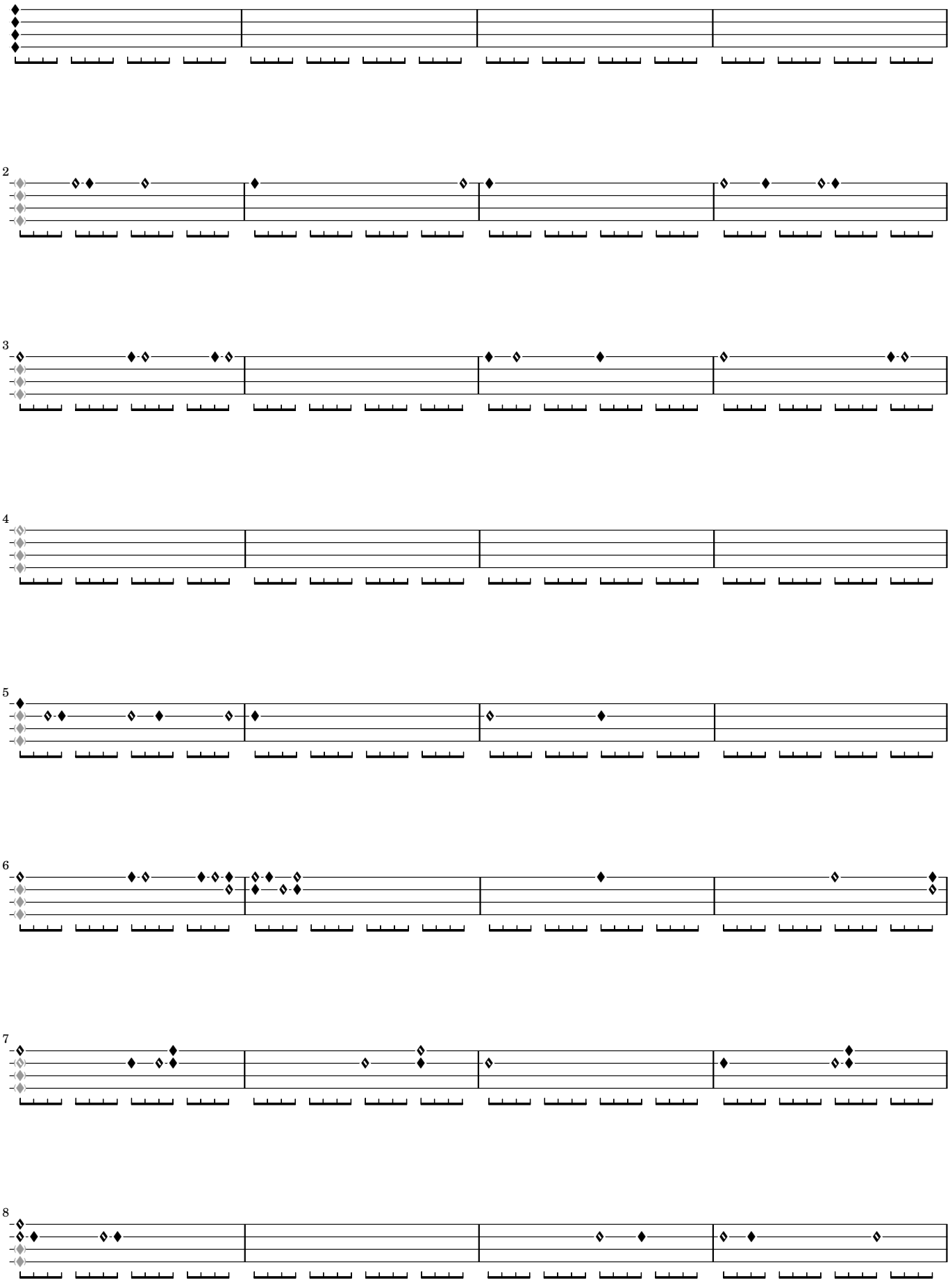
-michael winter

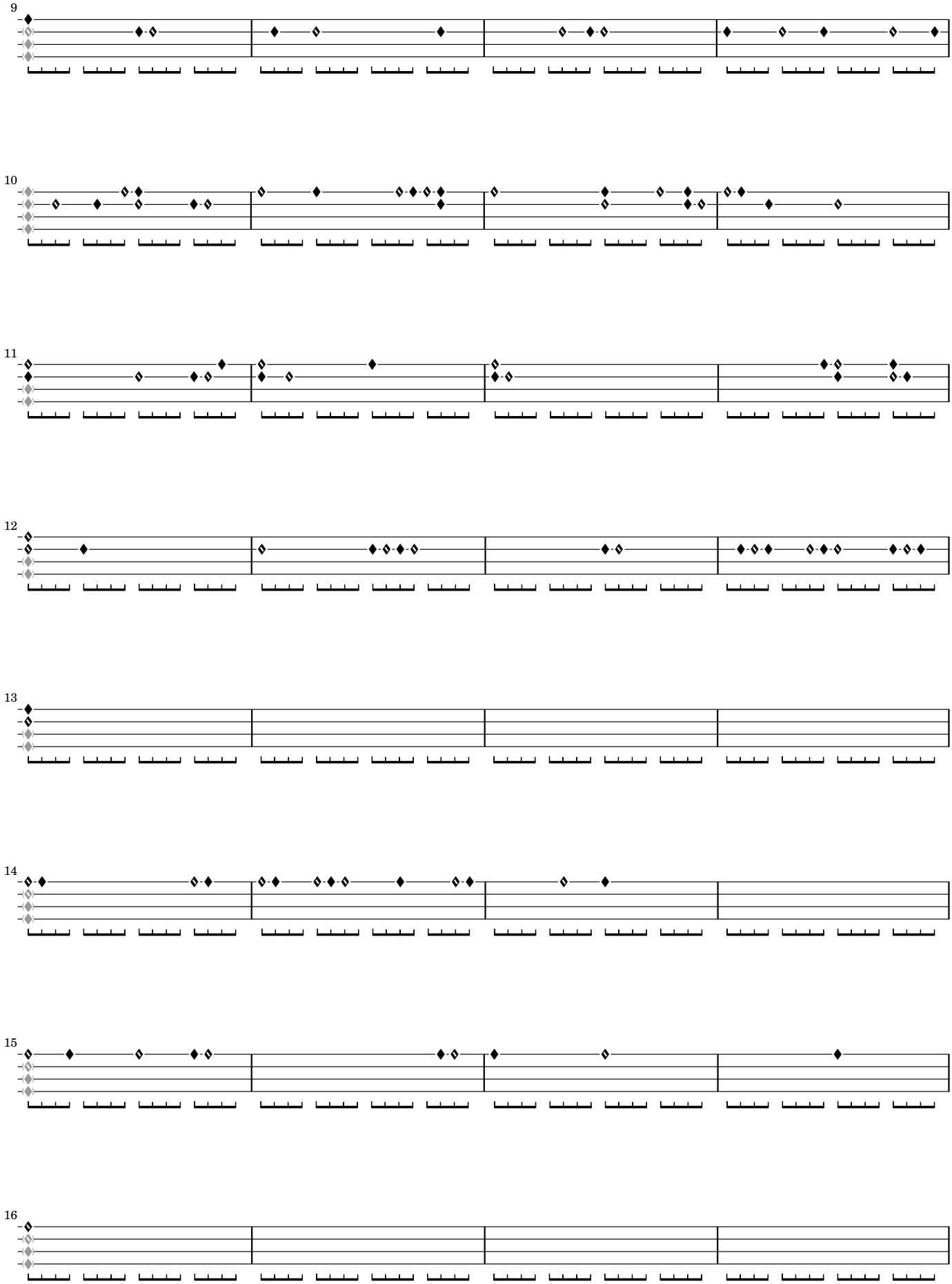
(Lima, Peru →
Trujillo, Peru →
Chiclayo, Peru →
Cocachimba, Peru →
Loja, Ecuador →
Cuenca, Ecuador →
Guayaquil, Ecuador →
Quito, Ecuador →
Cali, Colombia →
Bogota, Colombia →
Medellin, Colombia;
2017)

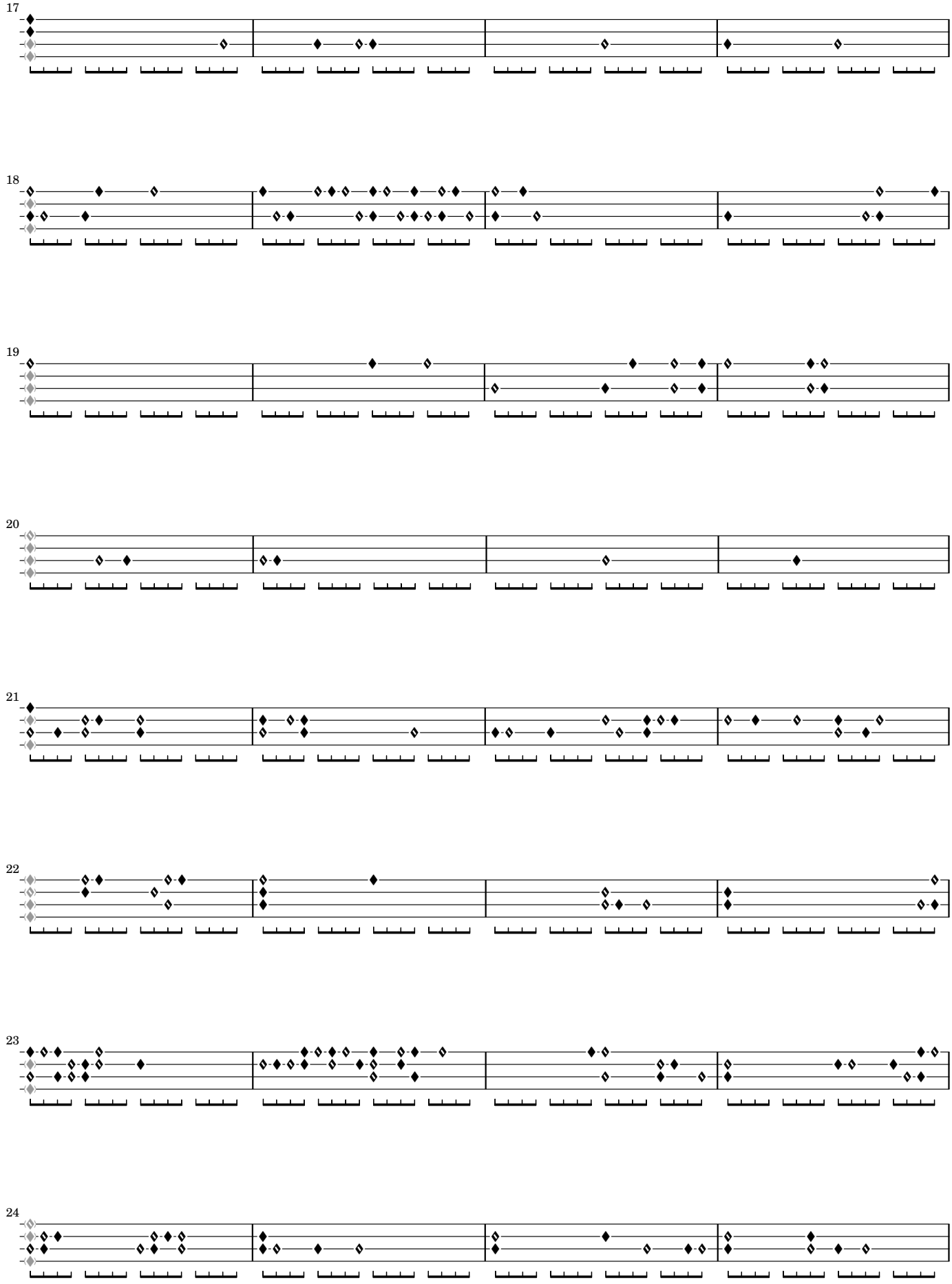
dedicated to Fatima Carolina Llerena Diaz

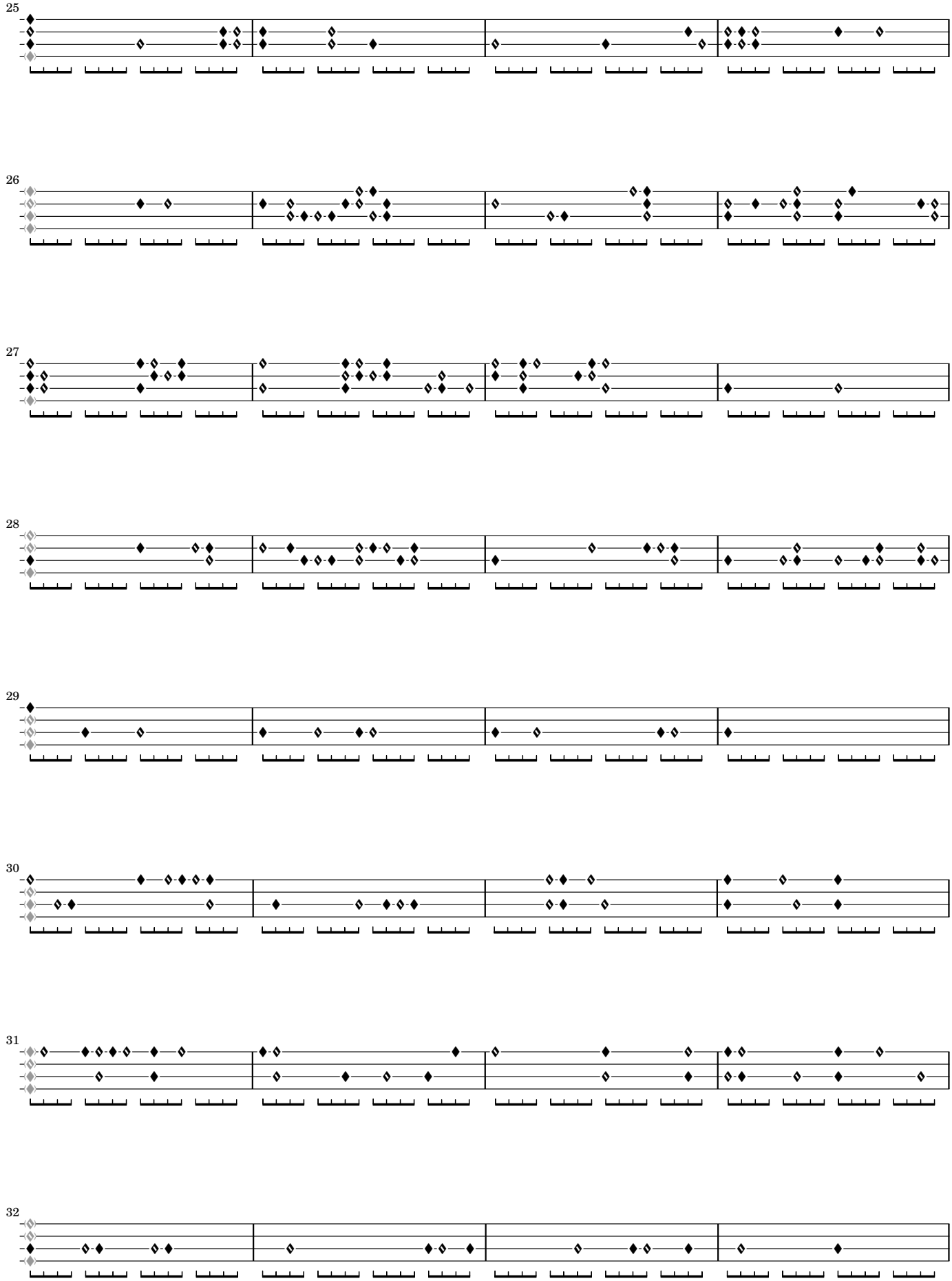
curved bow and negative

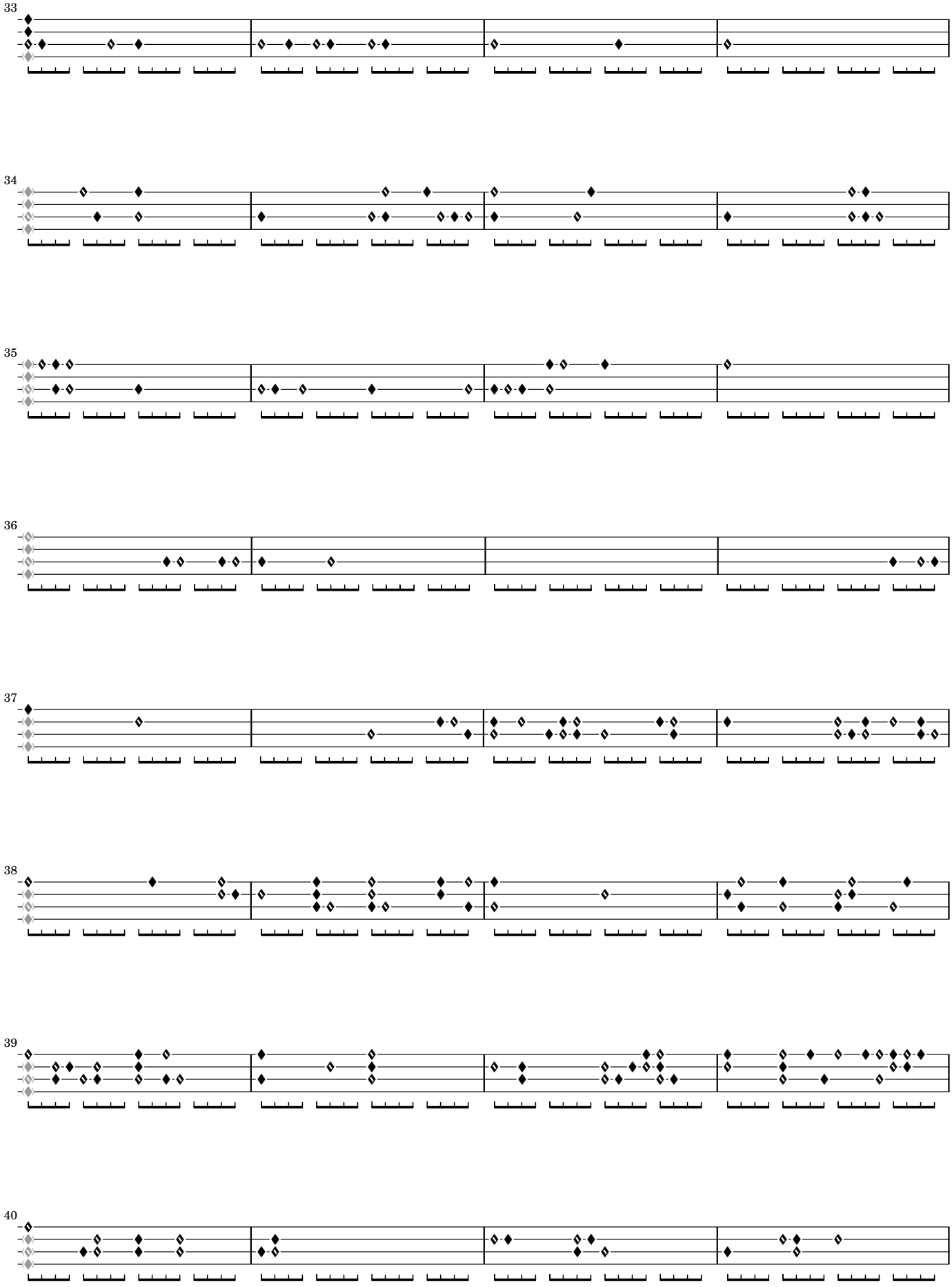
Medellin, Colombia; May 12, 2017

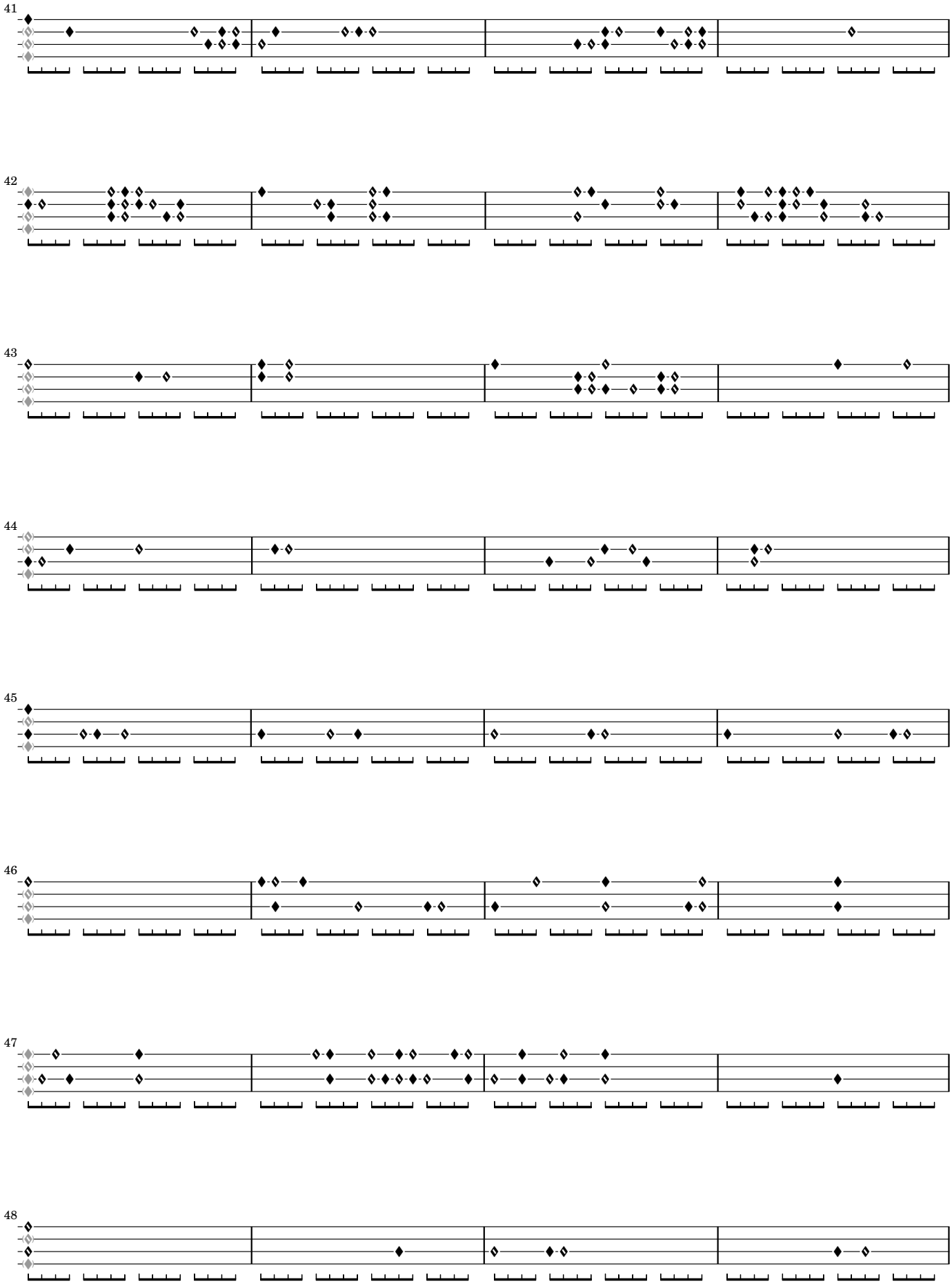


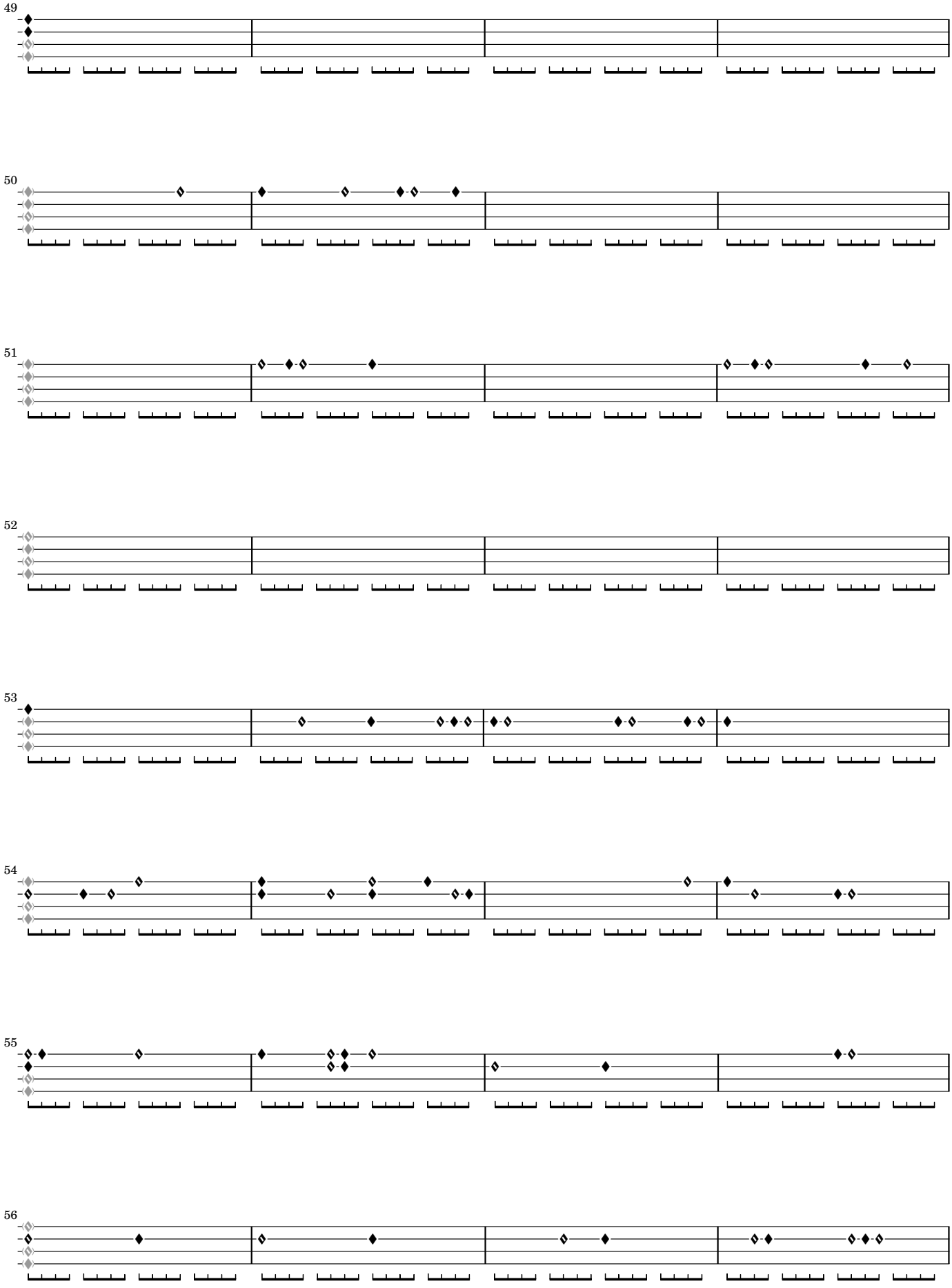

9
10
11
12
13
14
15
16

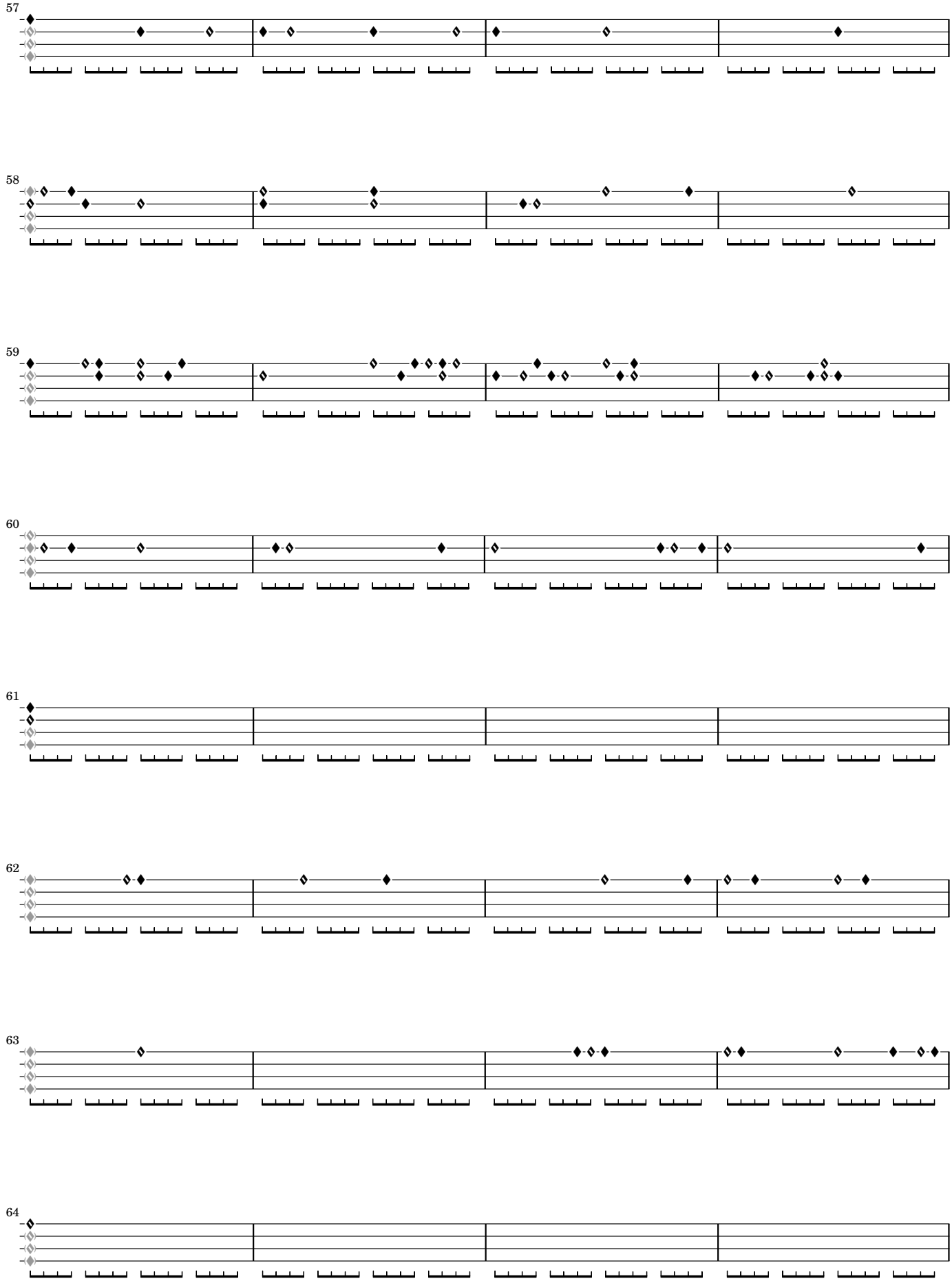

17
18
19
20
21
22
23
24

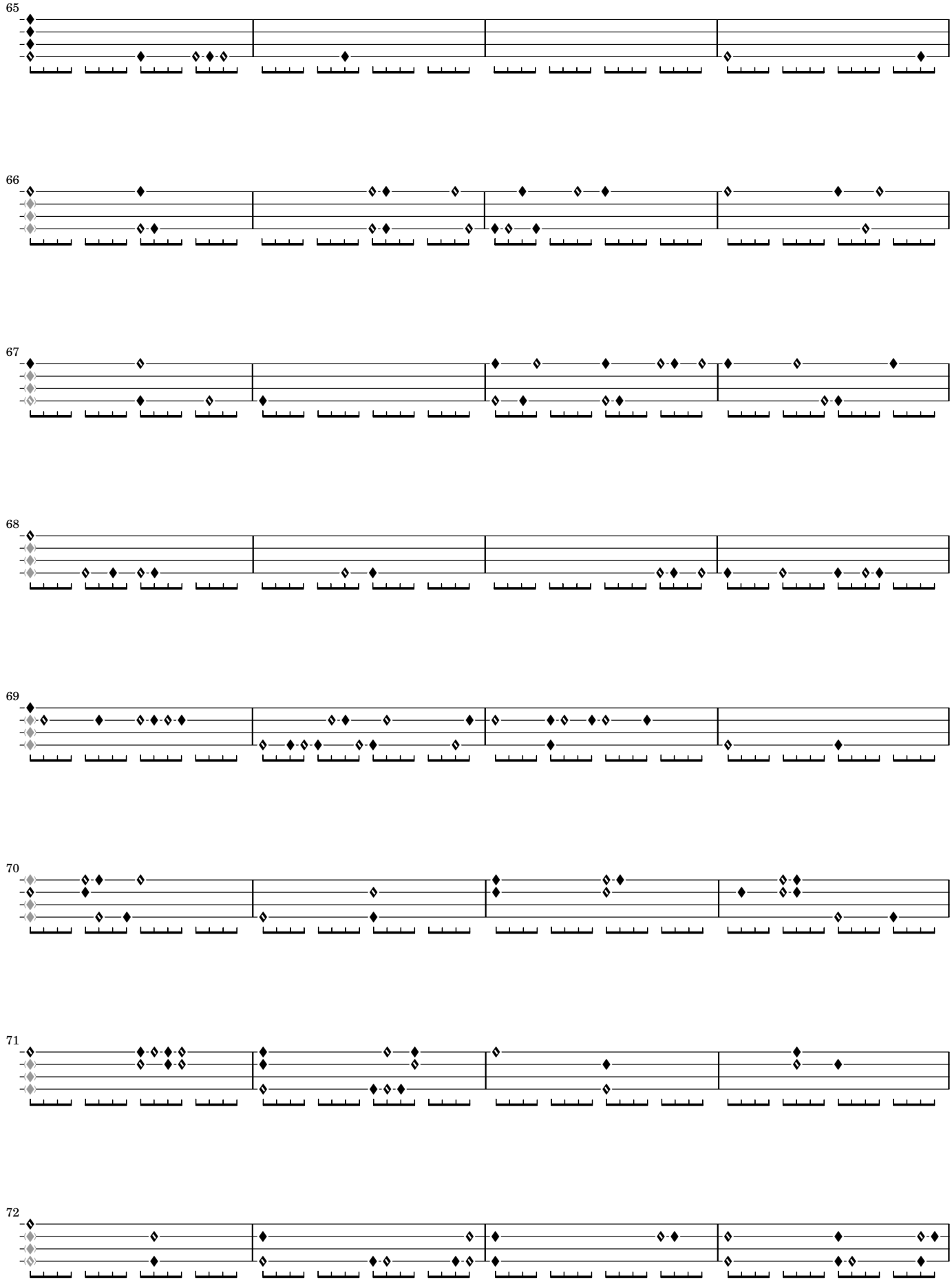

25
26
27
28
29
30
31
32

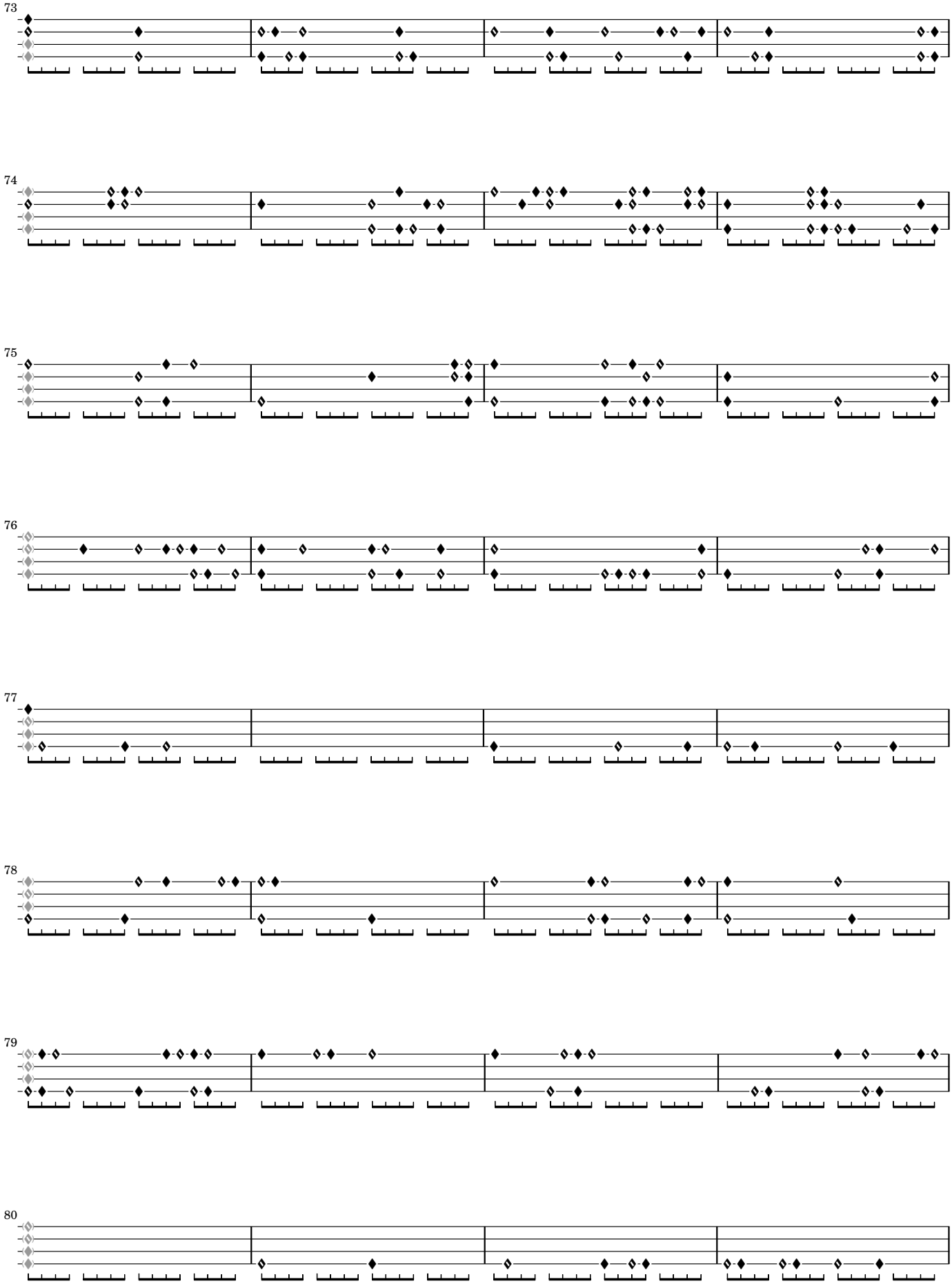

33
34
35
36
37
38
39
40

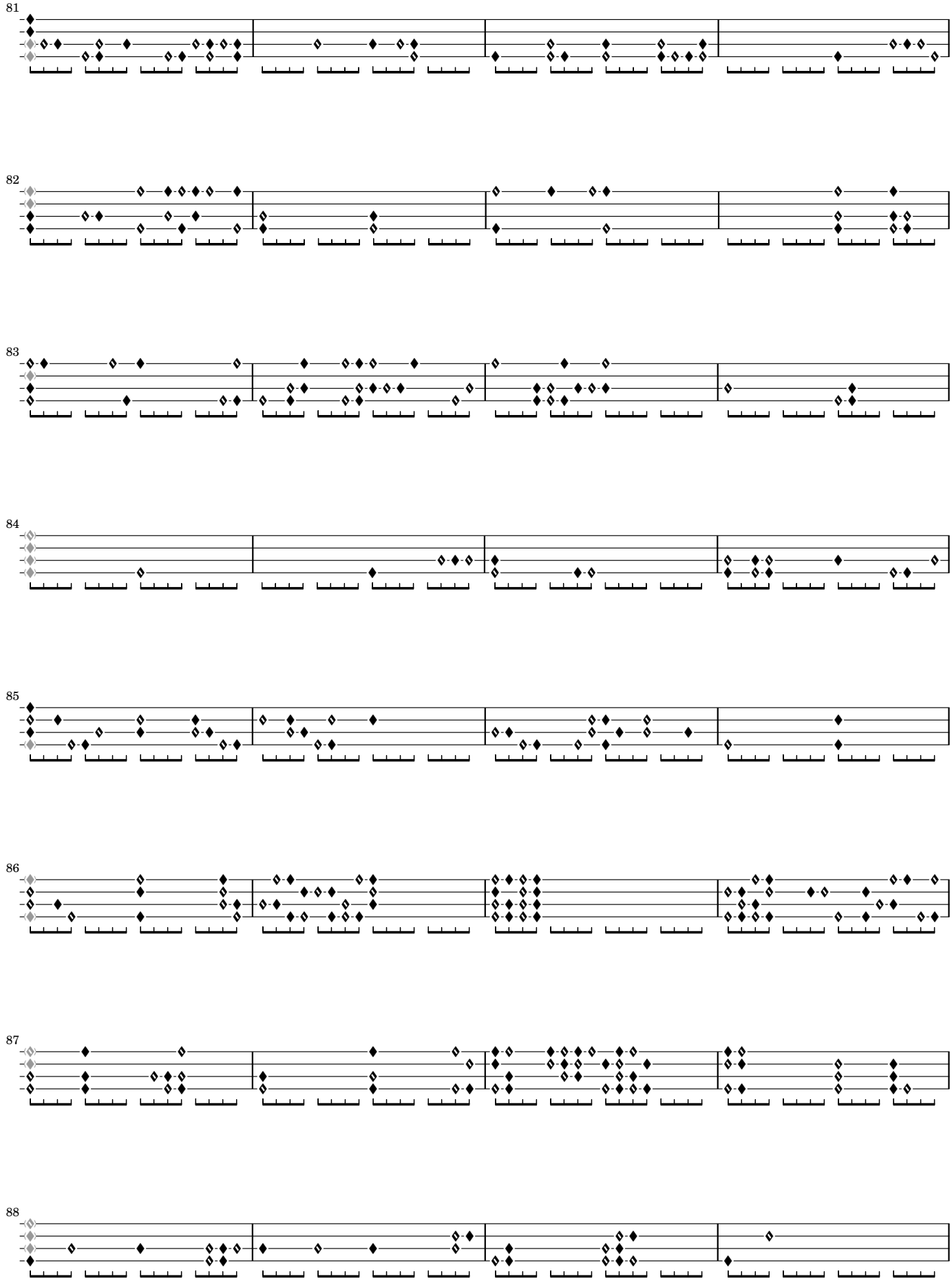

41
42
43
44
45
46
47
48

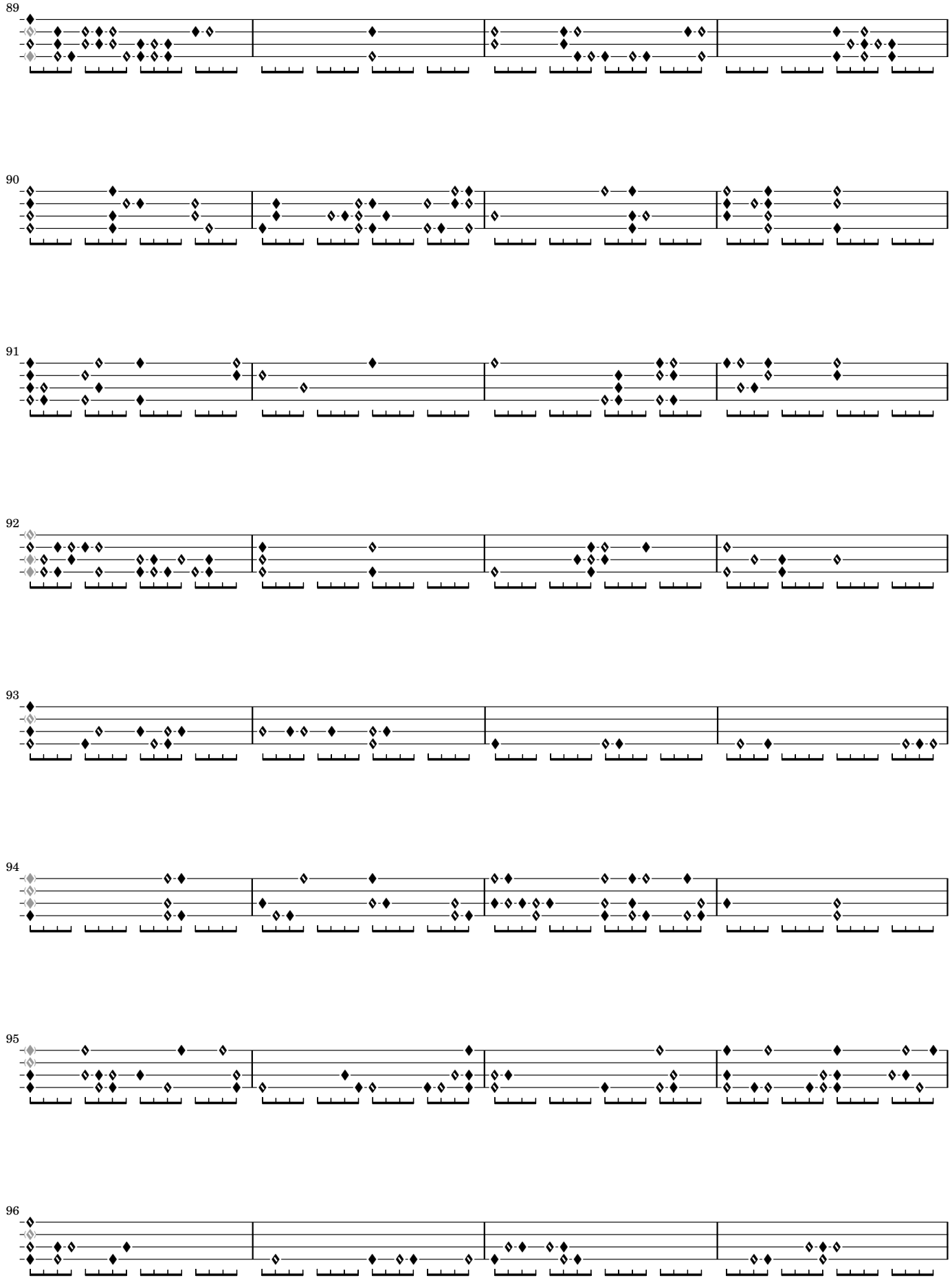

49
50
51
52
53
54
55
56

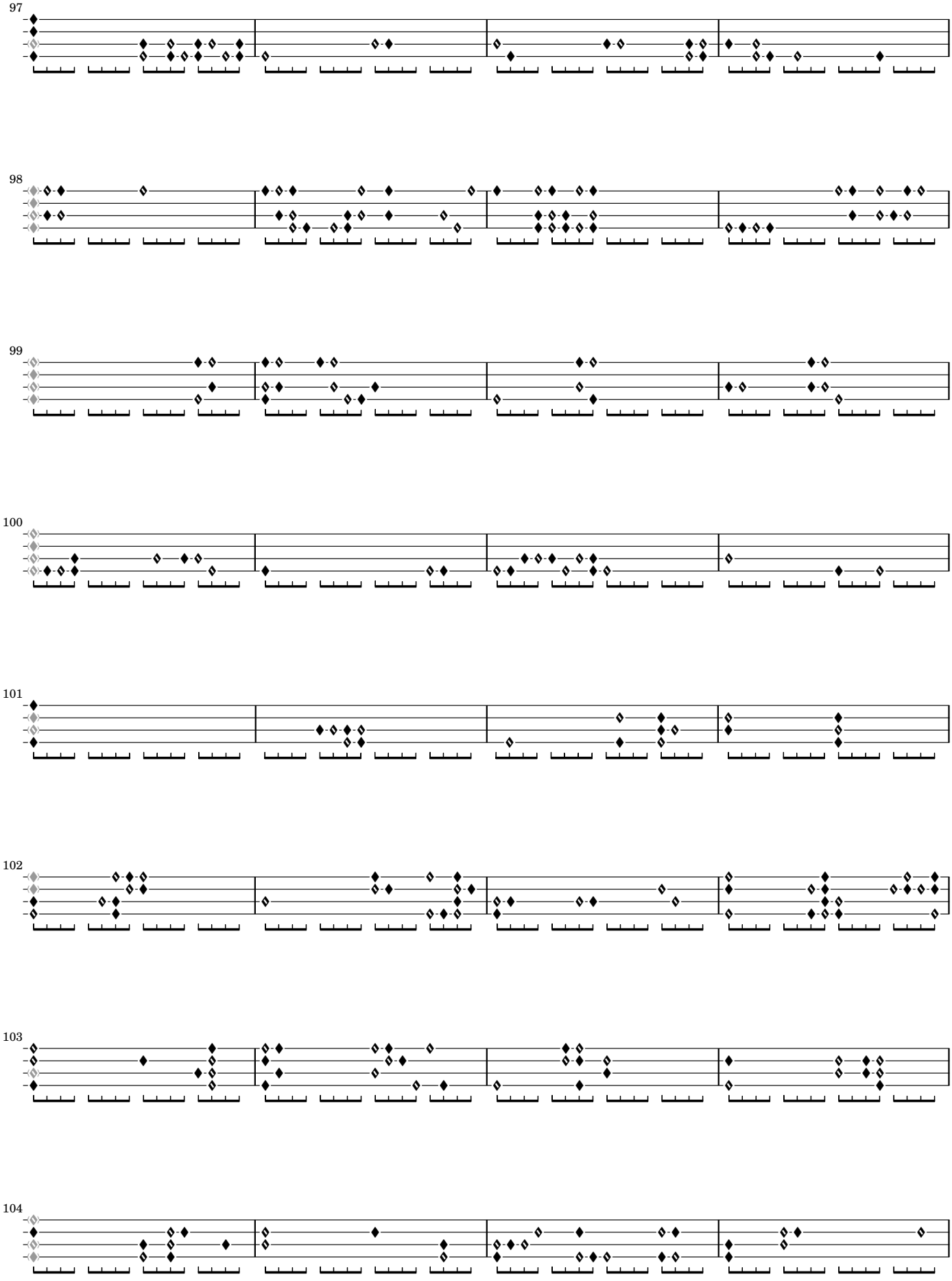

57
58
59
60
61
62
63
64

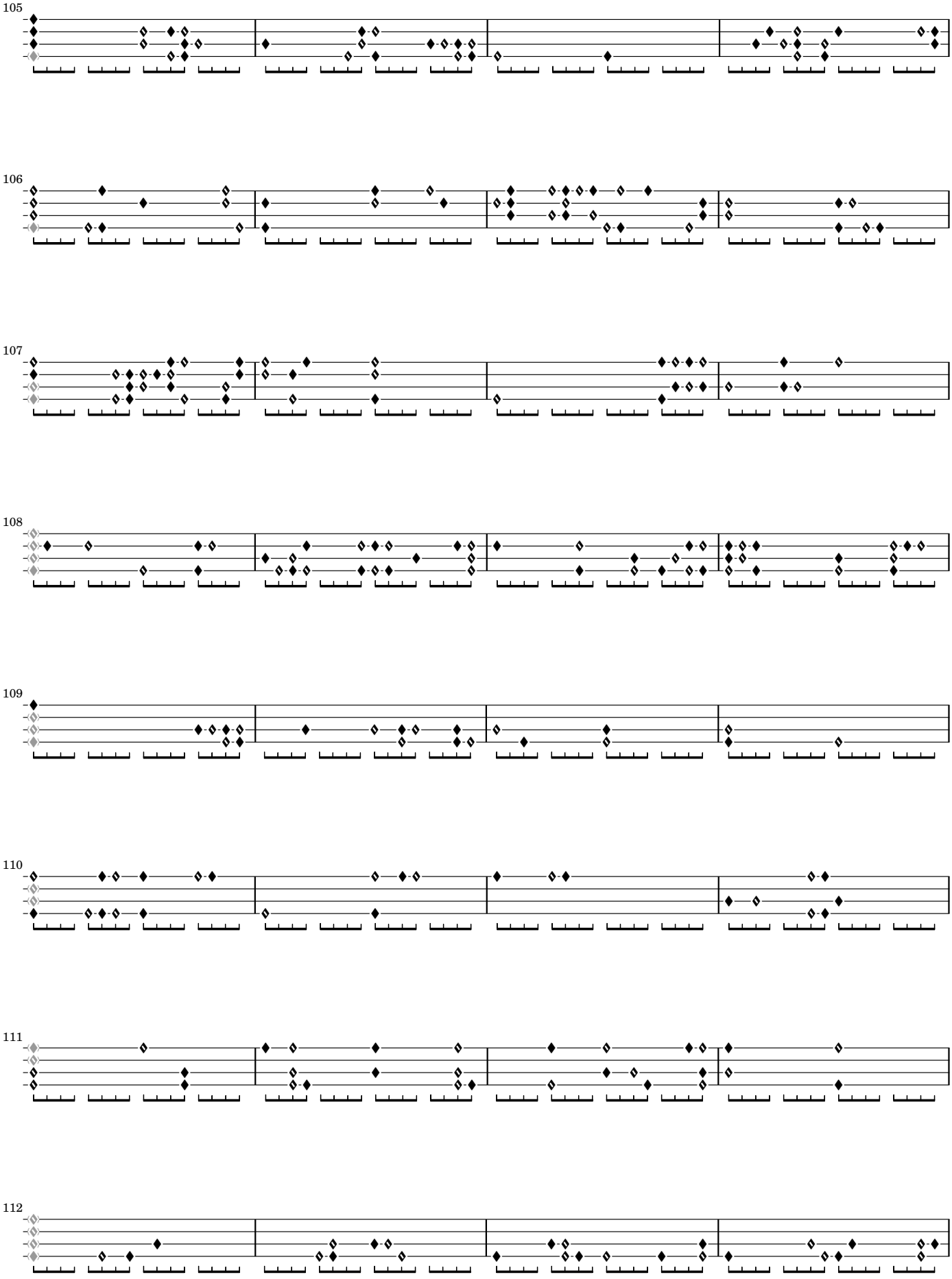

65
66
67
68
69
70
71
72

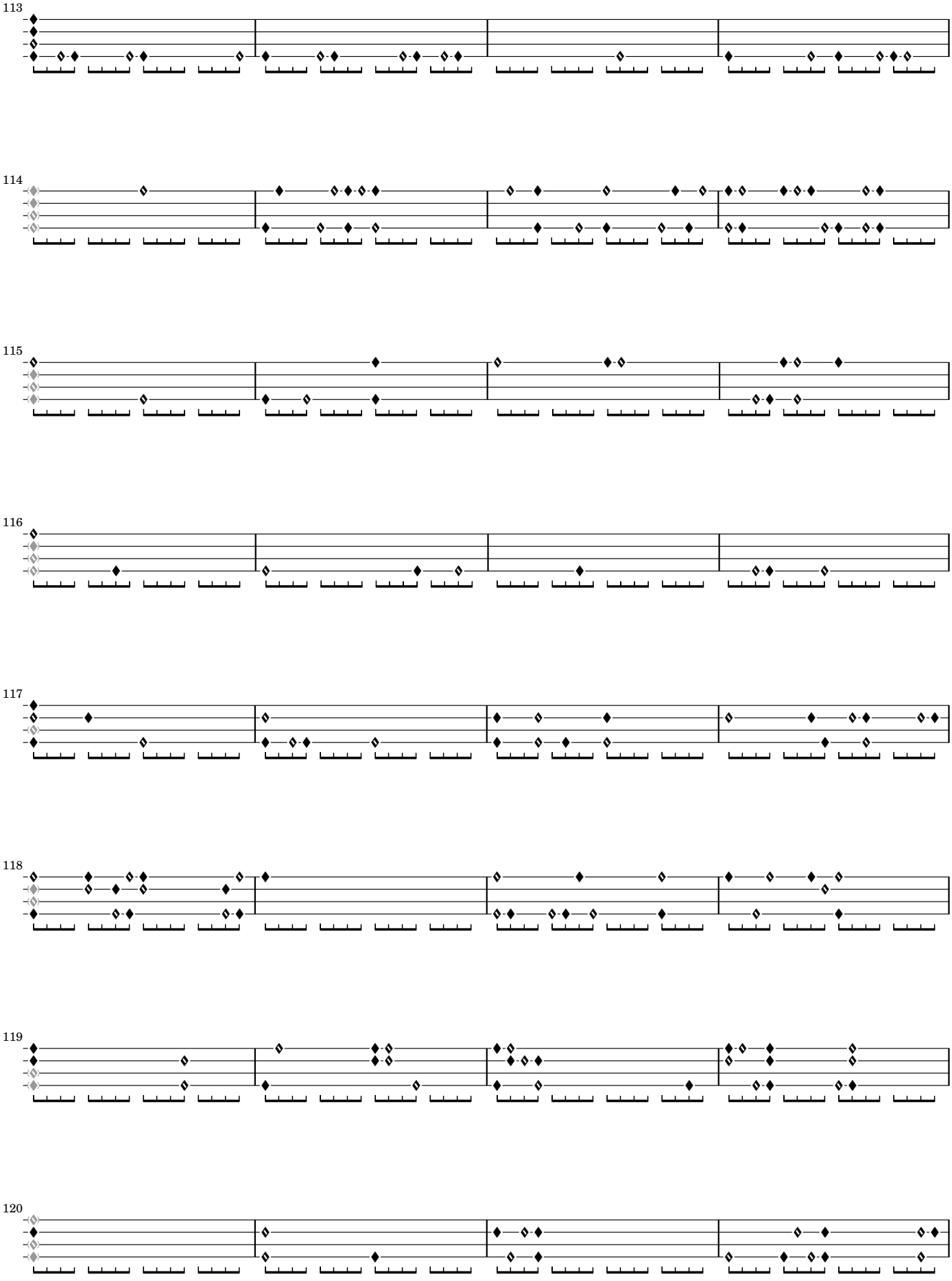

73
74
75
76
77
78
79
80

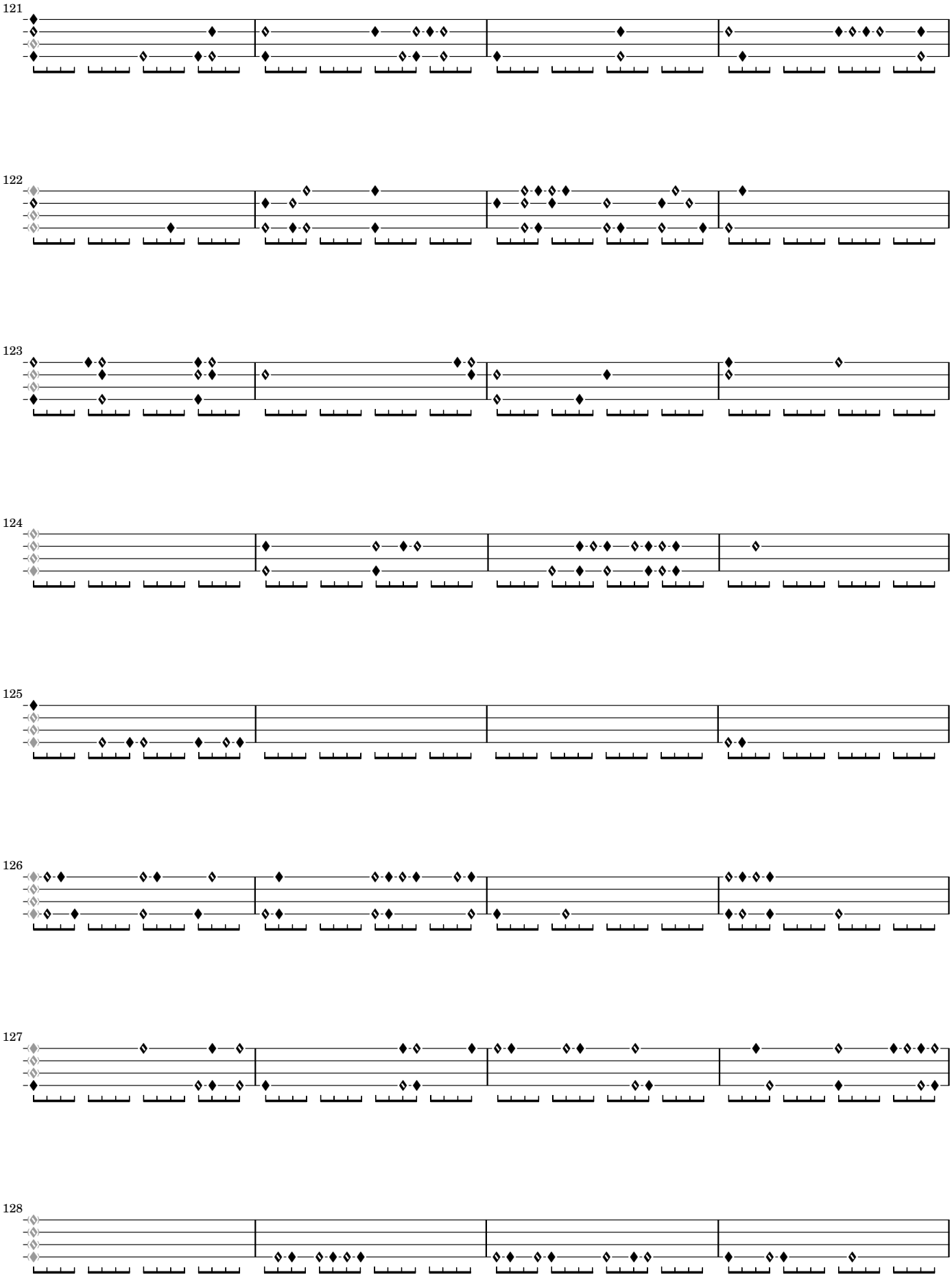

81
82
83
84
85
86
87
88

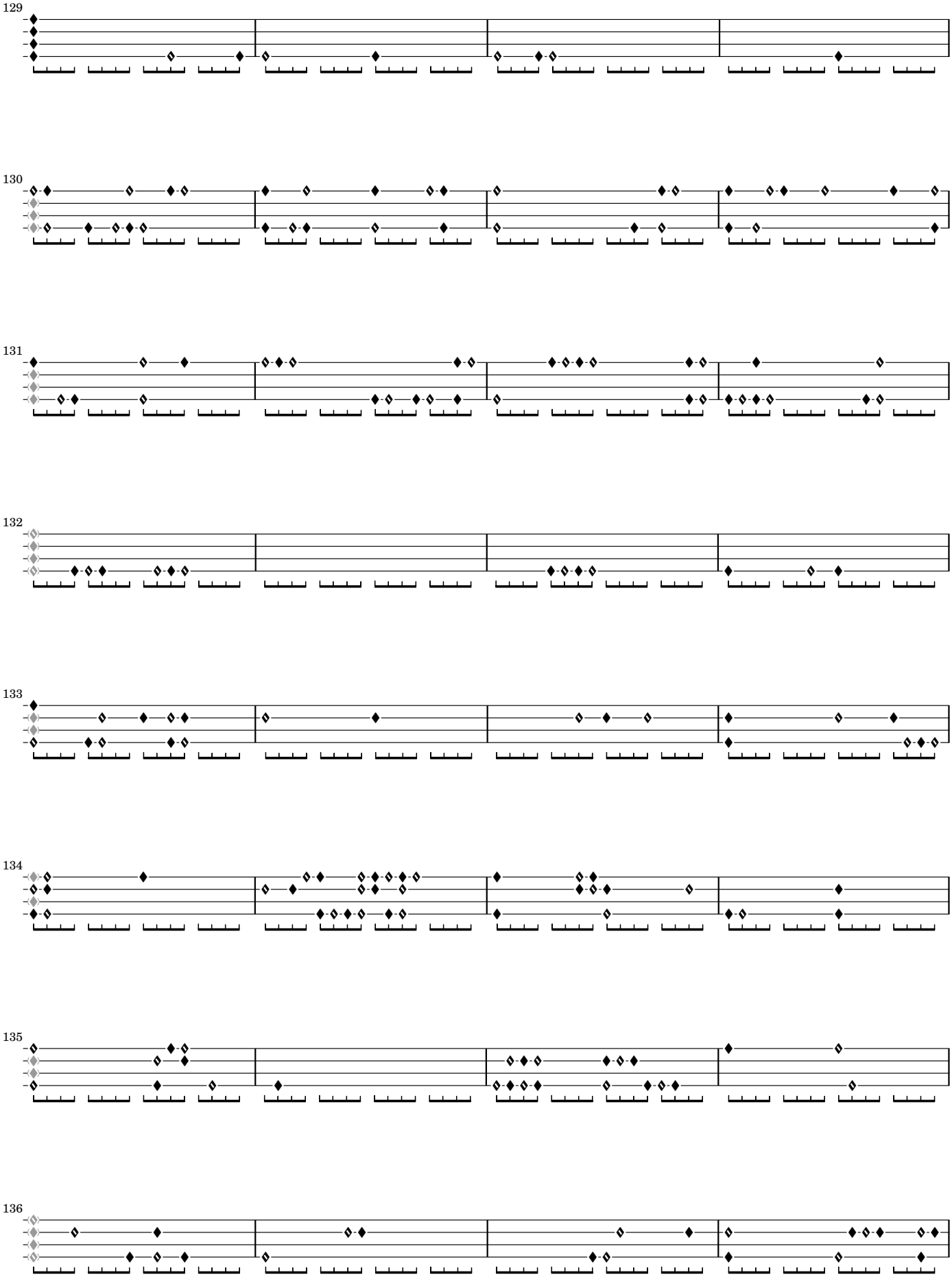

89
90
91
92
93
94
95
96

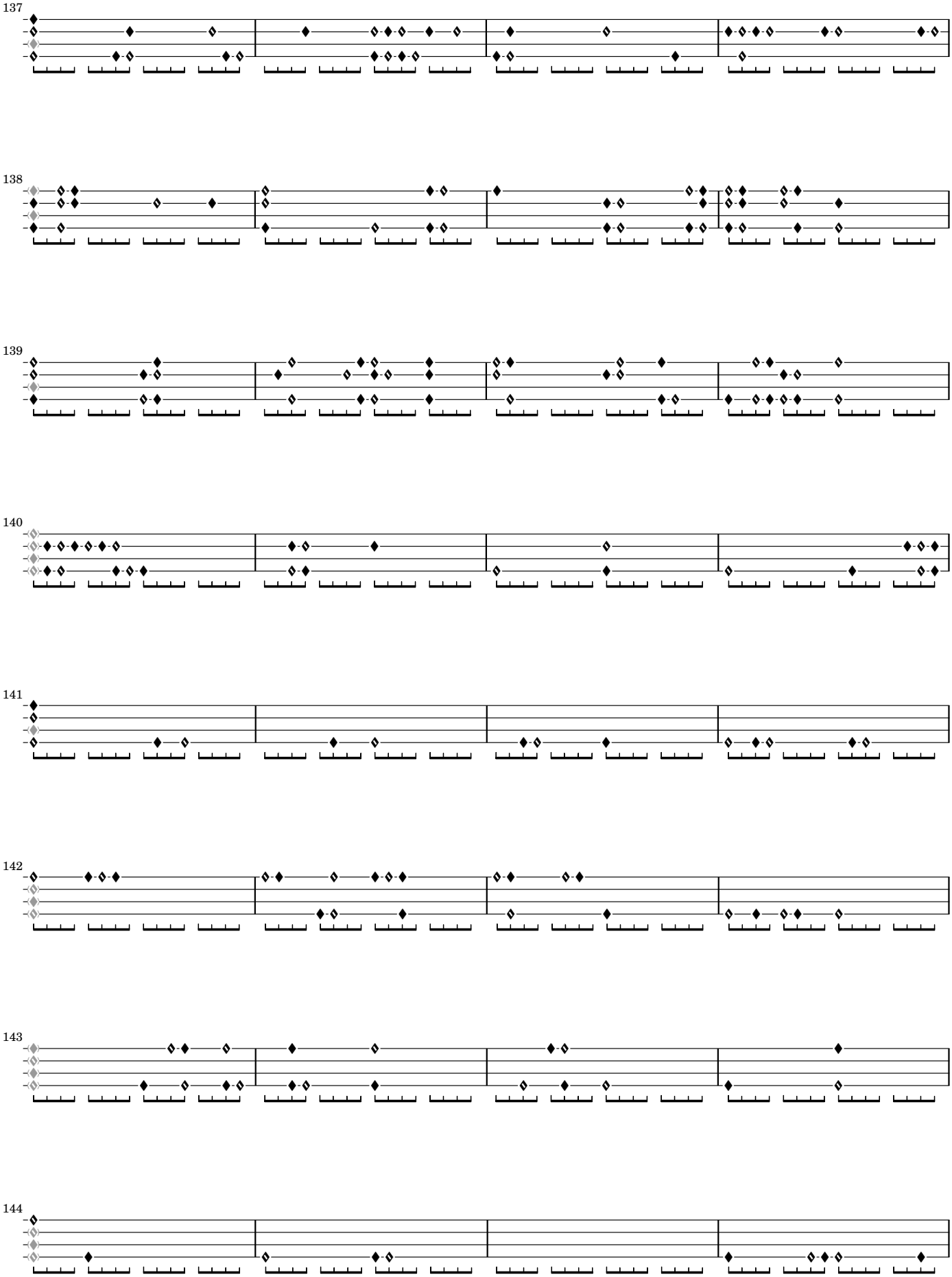

97
98
99
100
101
102
103
104

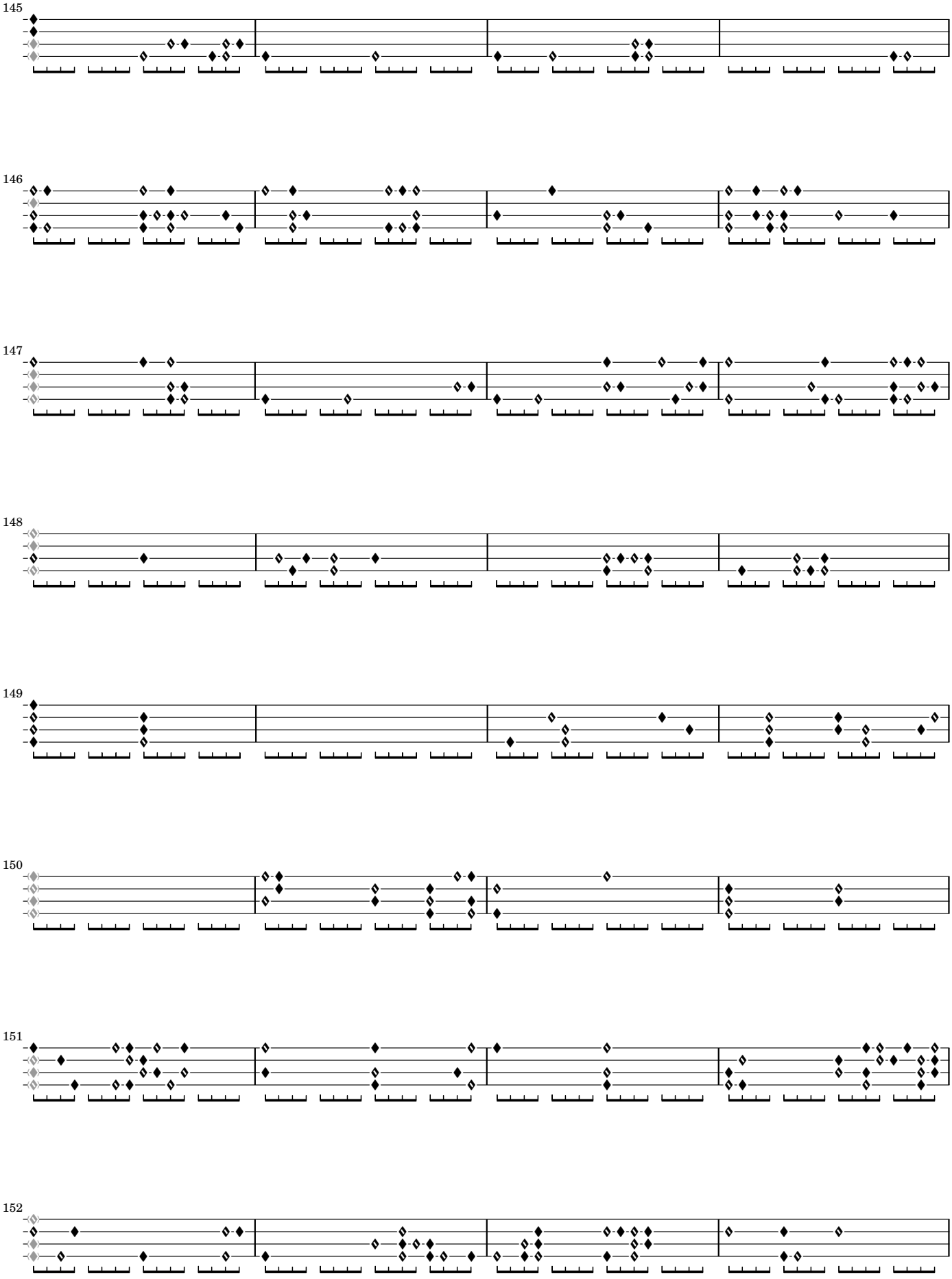

105
106
107
108
109
110
111
112


113
114
115
116
117
118
119
120

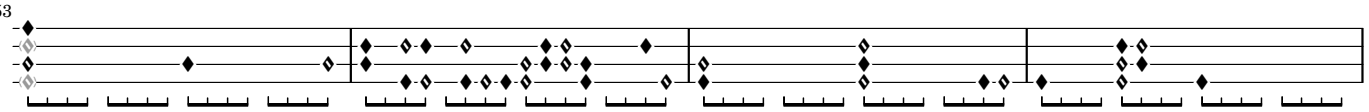

121
122
123
124
125
126
127
128


129
130
131
132
133
134
135
136

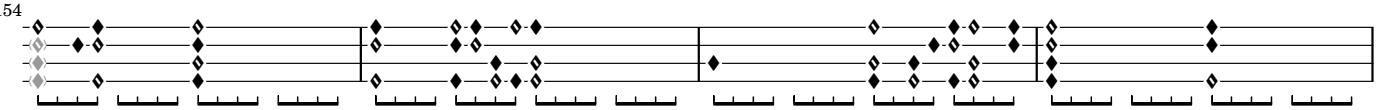

137
138
139
140
141
142
143
144


145
146
147
148
149
150
151
152

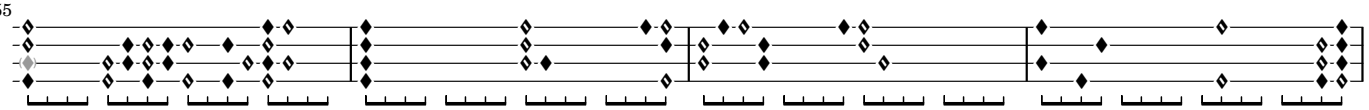
153



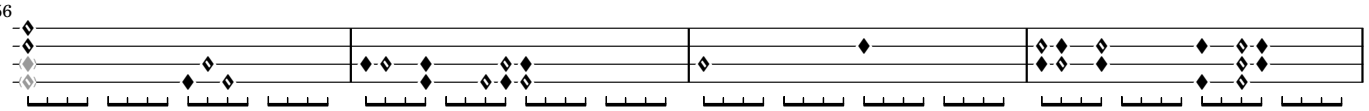
154



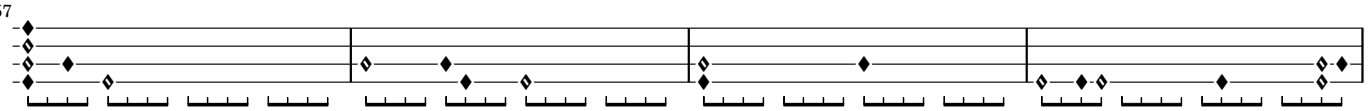
155



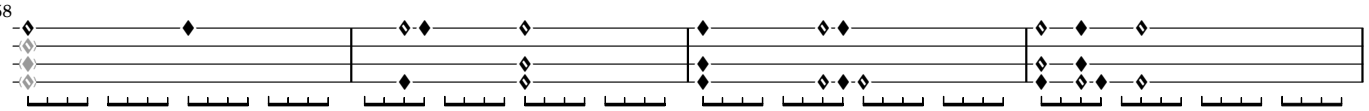
156



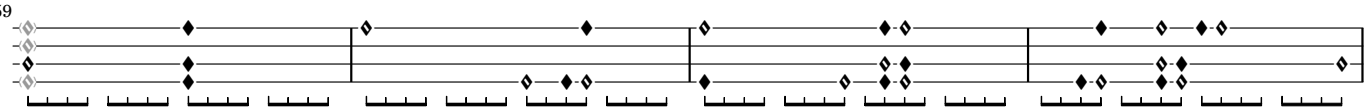
157



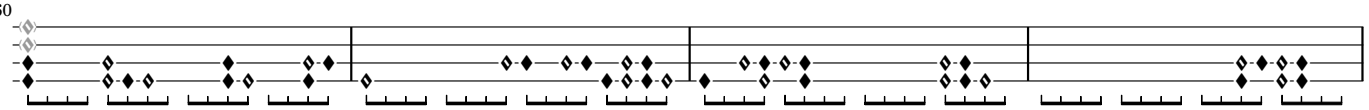
158

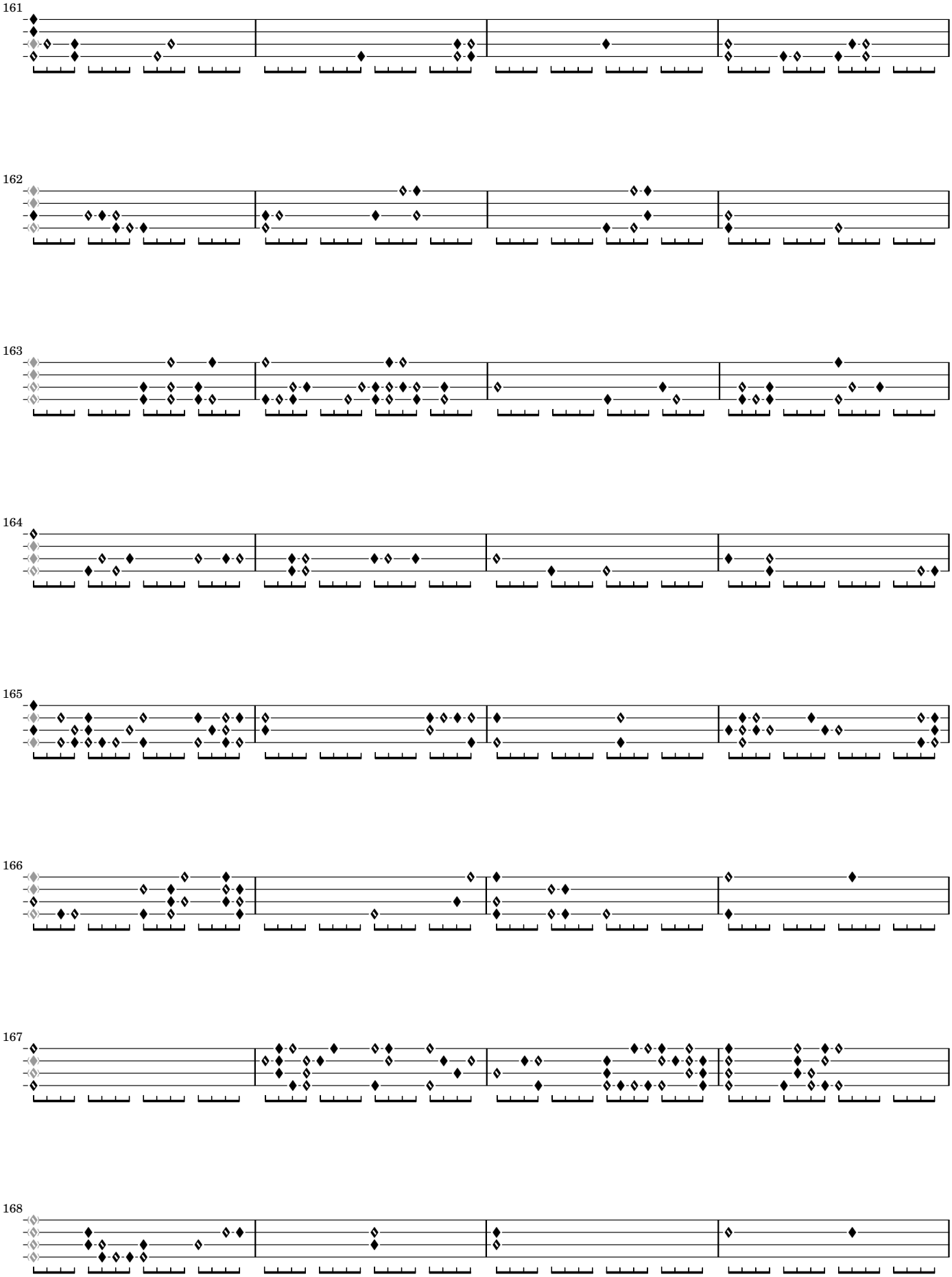


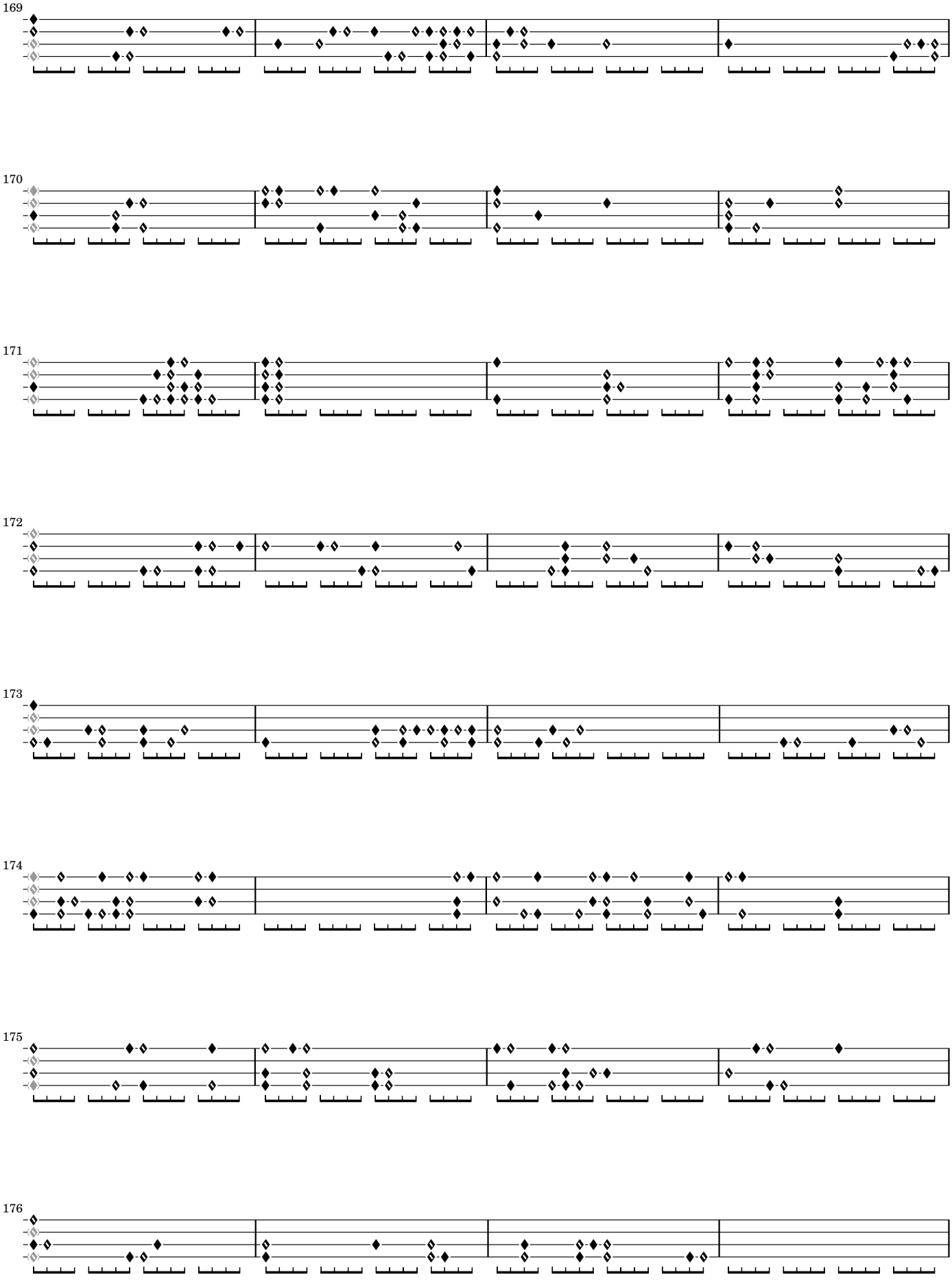
159

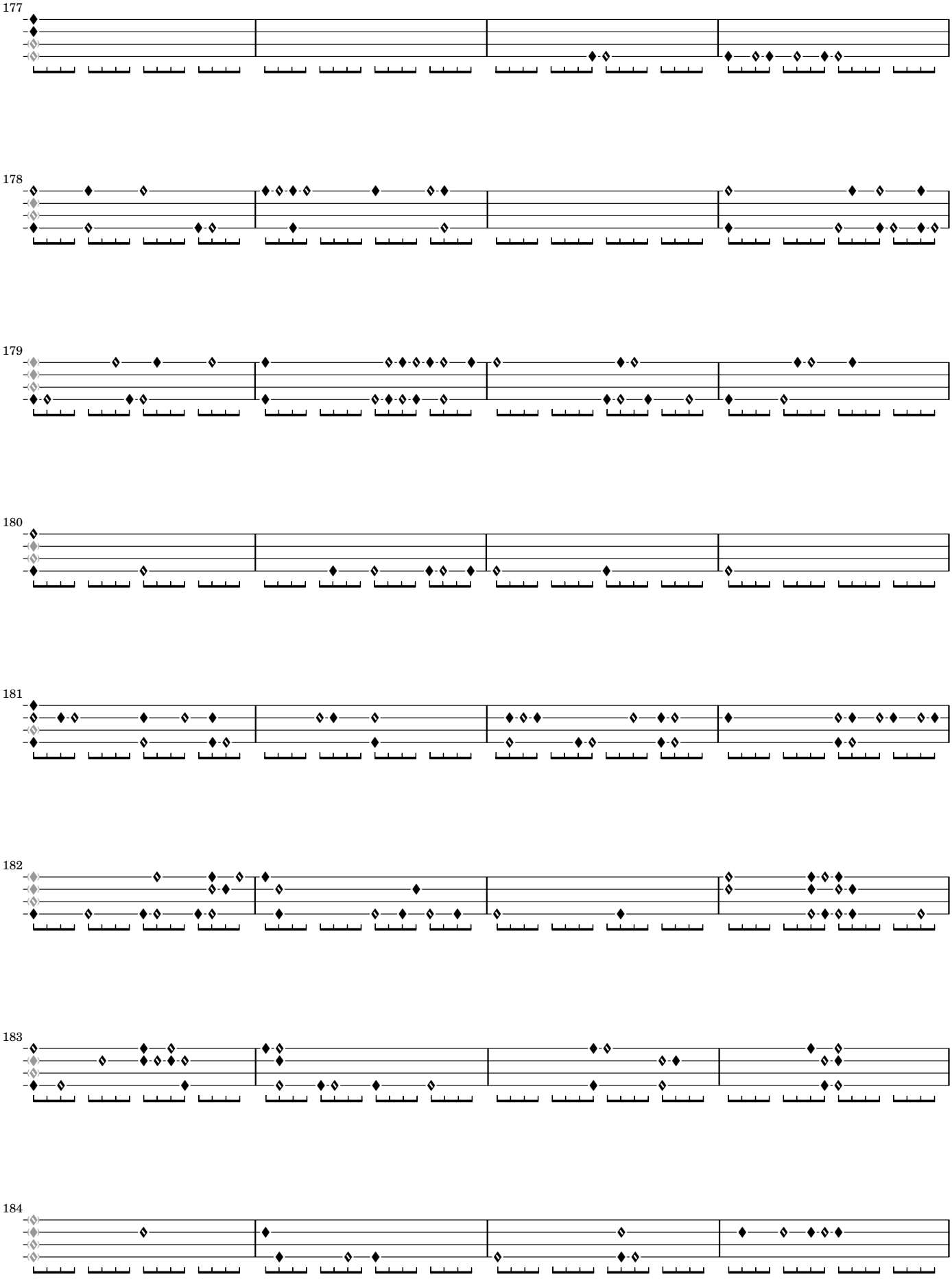


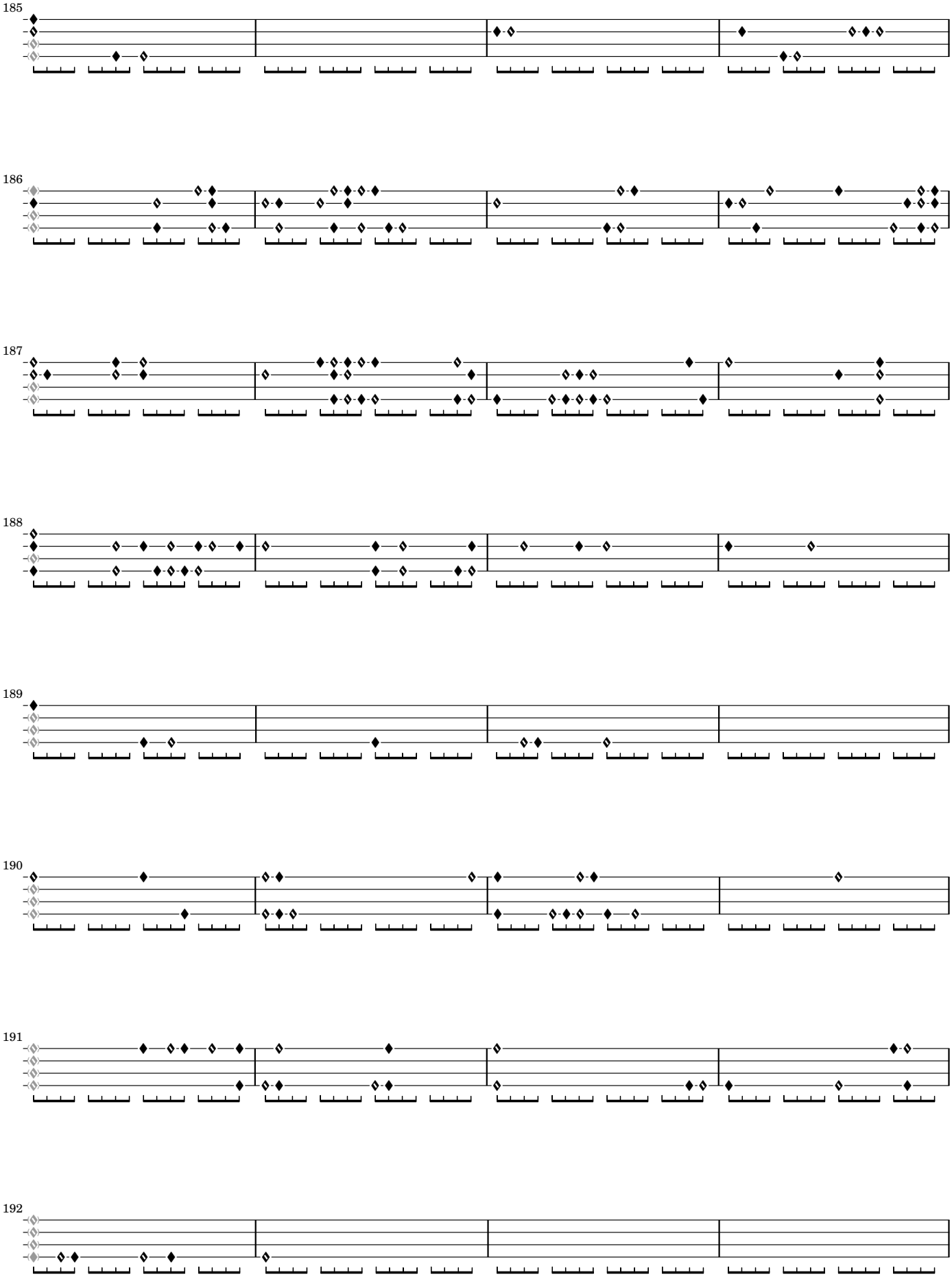
160

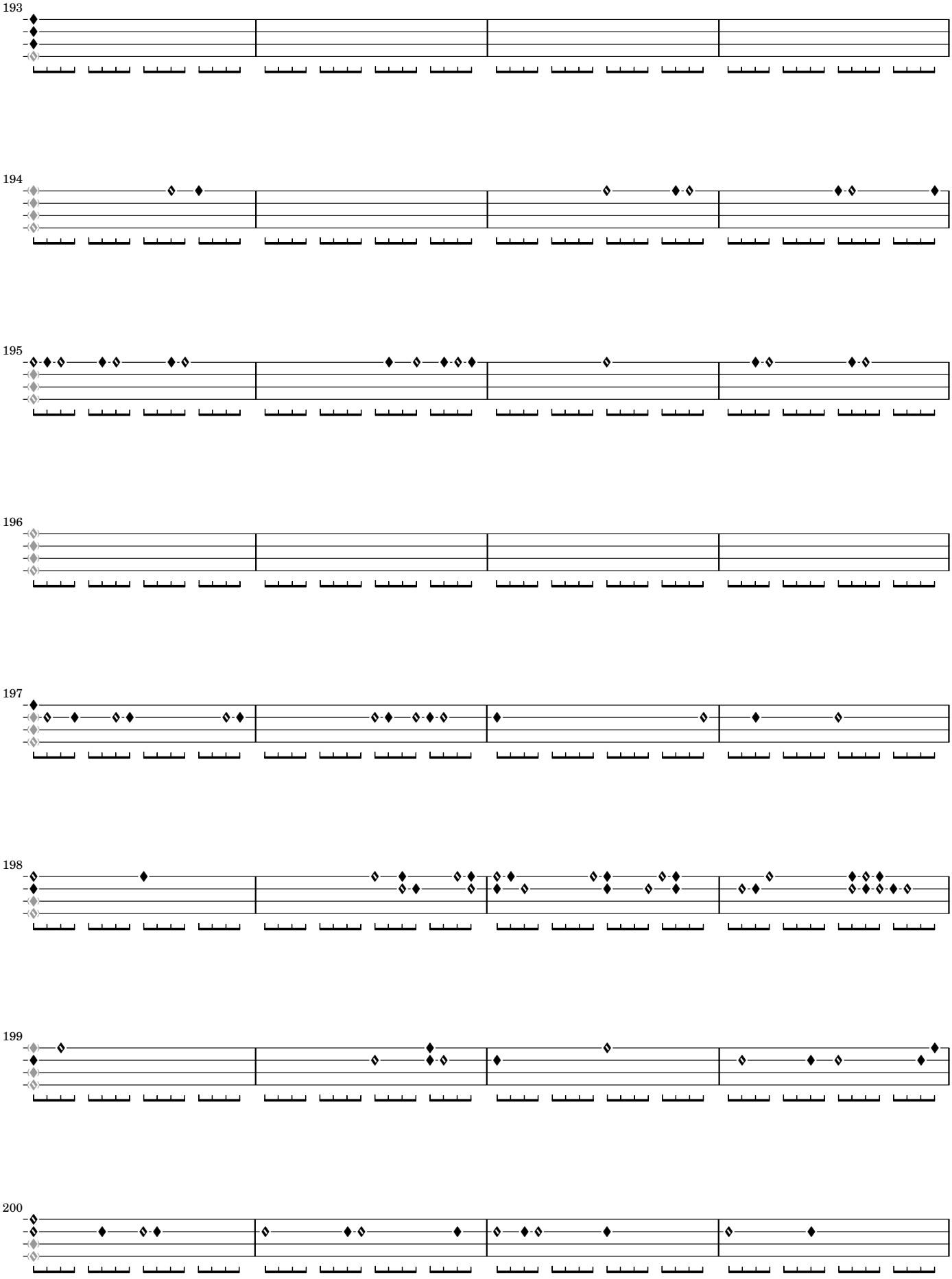


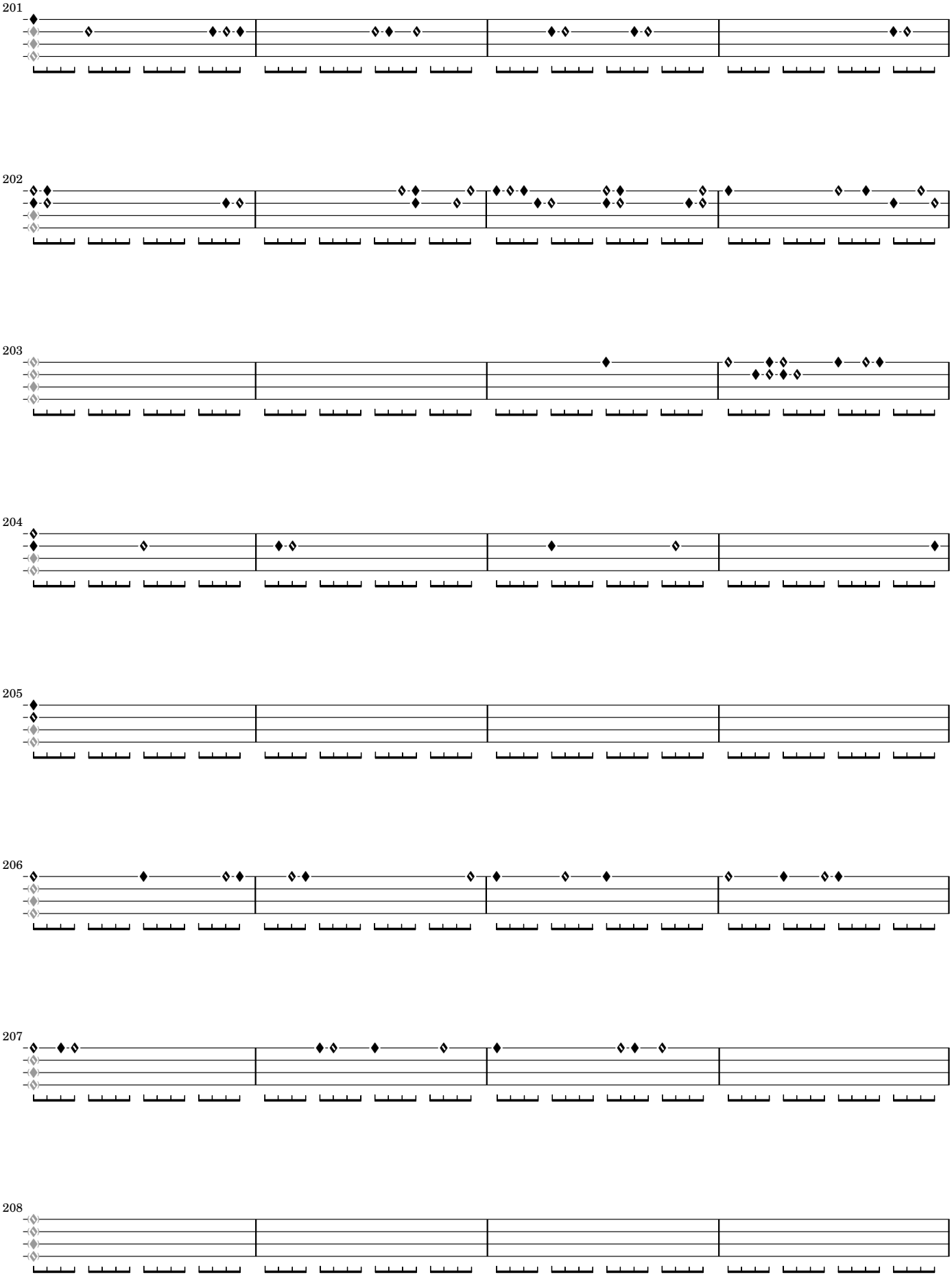

161
162
163
164
165
166
167
168

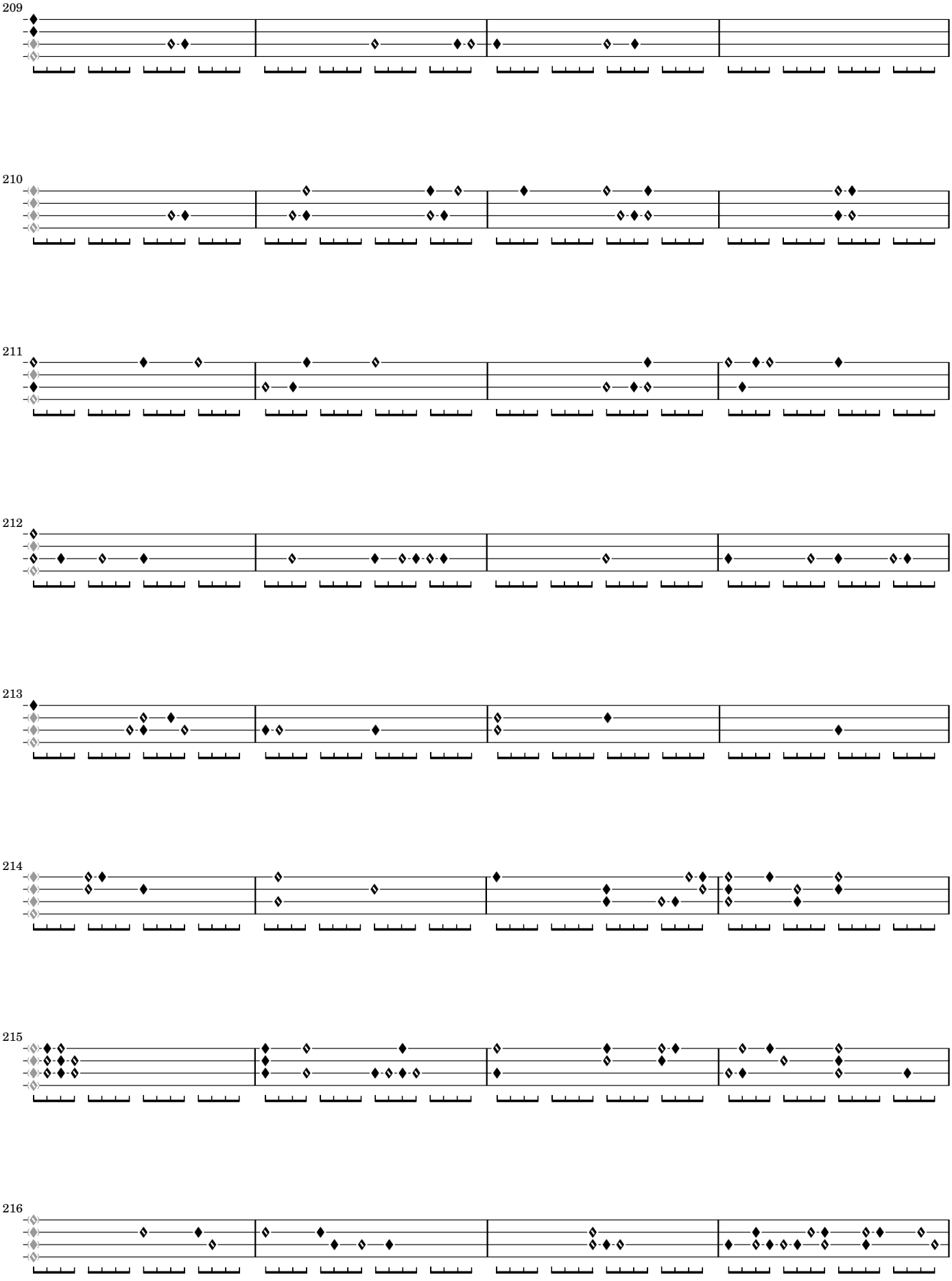

169
170
171
172
173
174
175
176

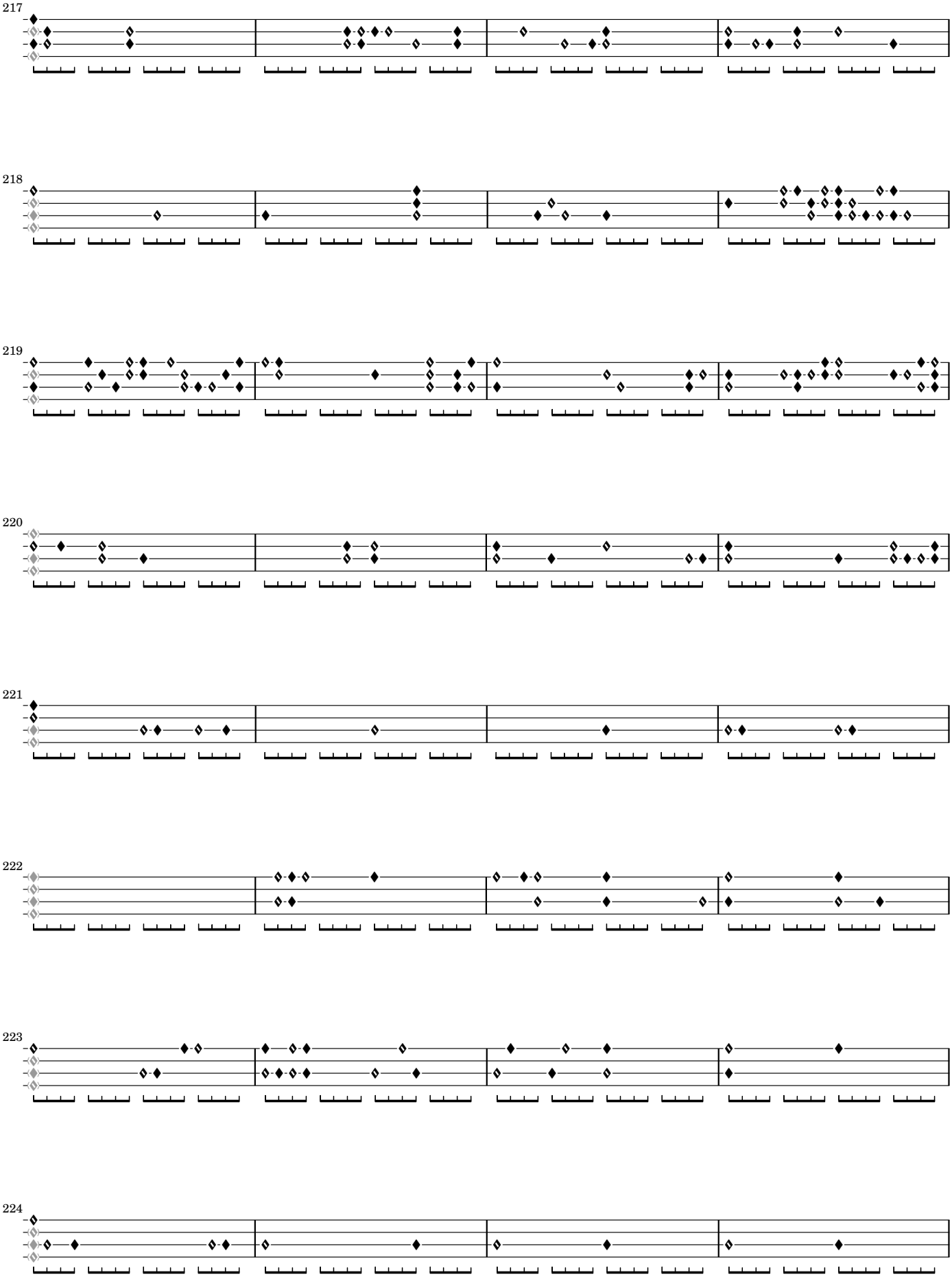

177
178
179
180
181
182
183
184

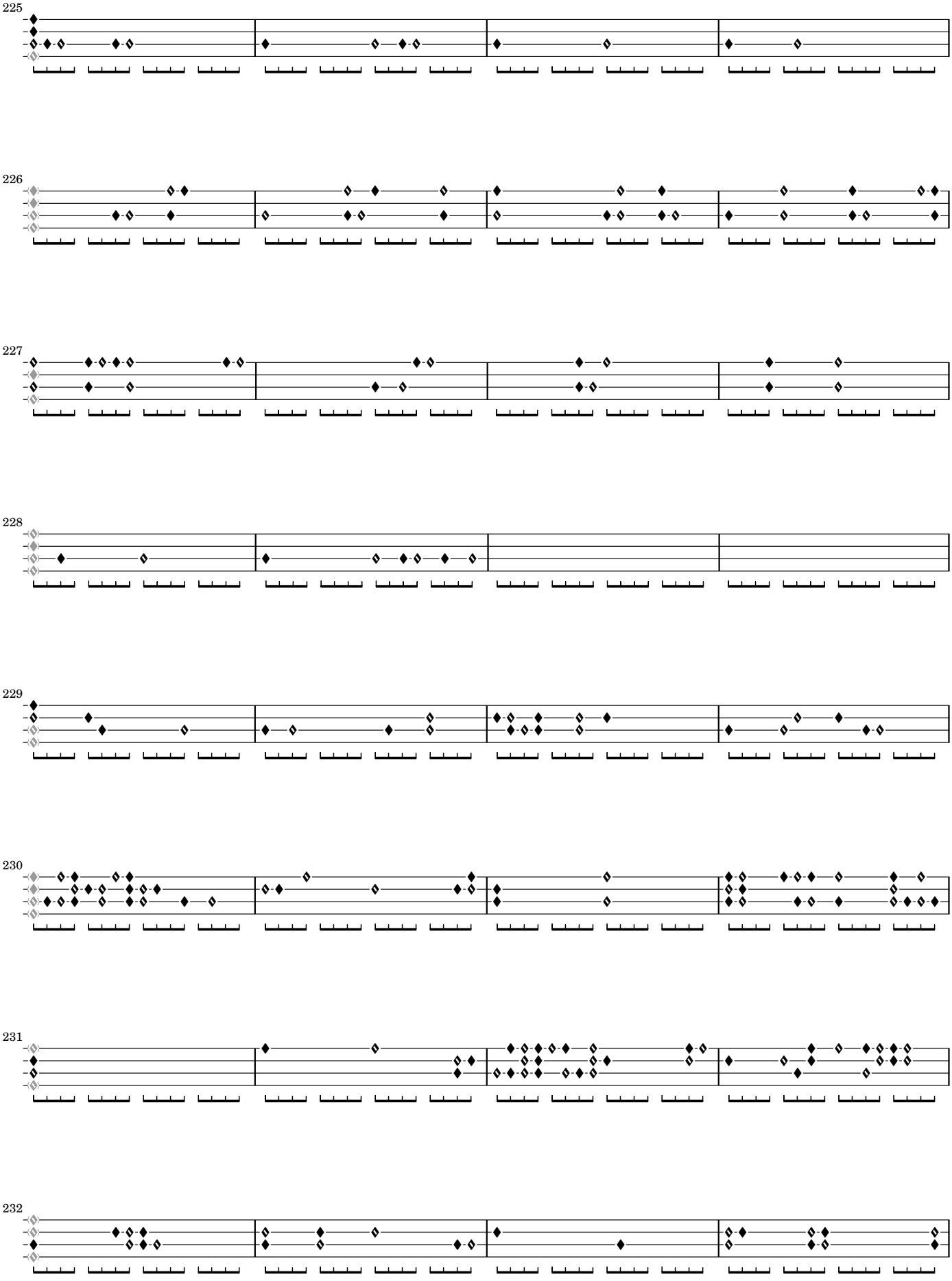

185
186
187
188
189
190
191
192

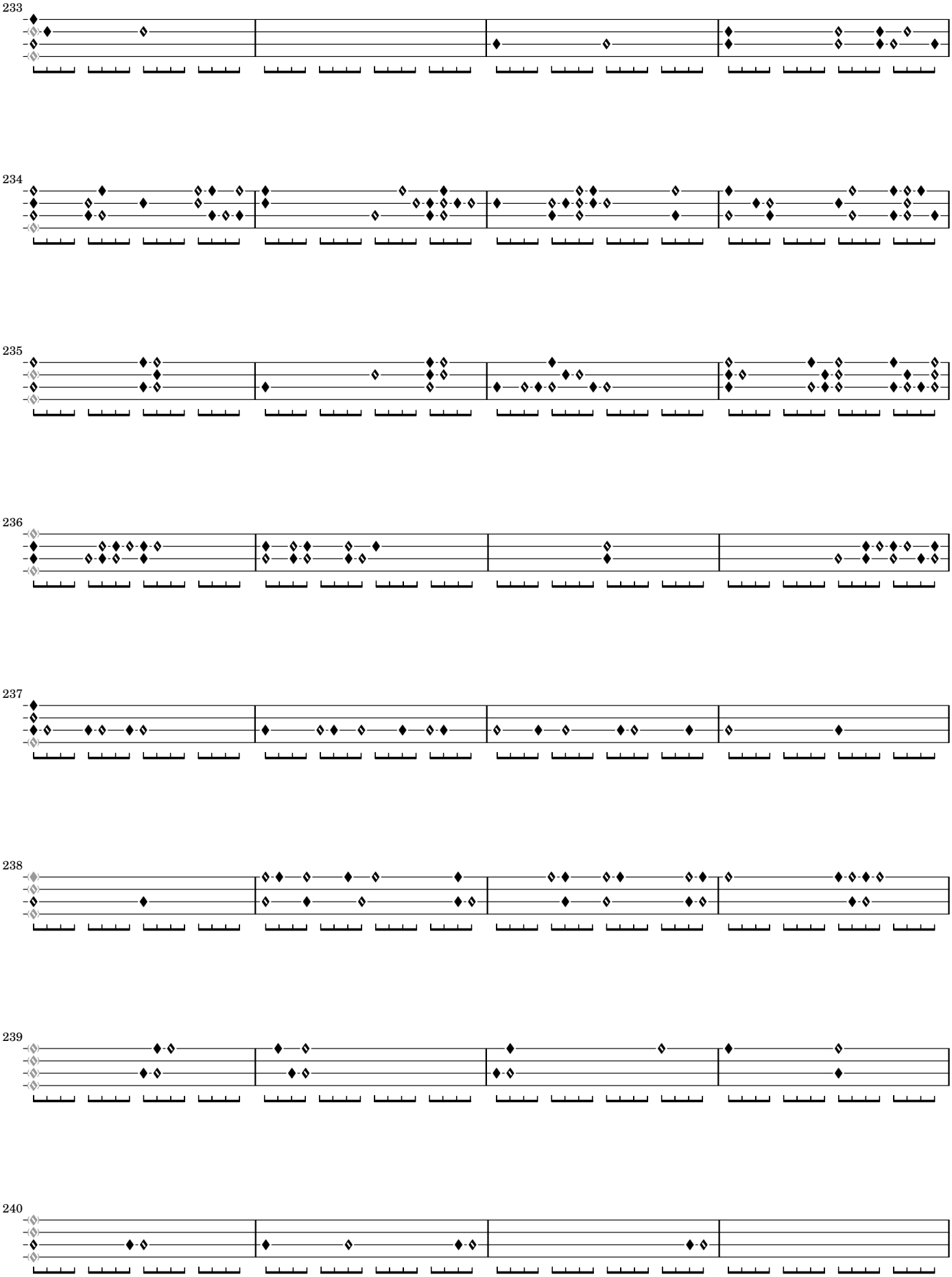

193
194
195
196
197
198
199
200

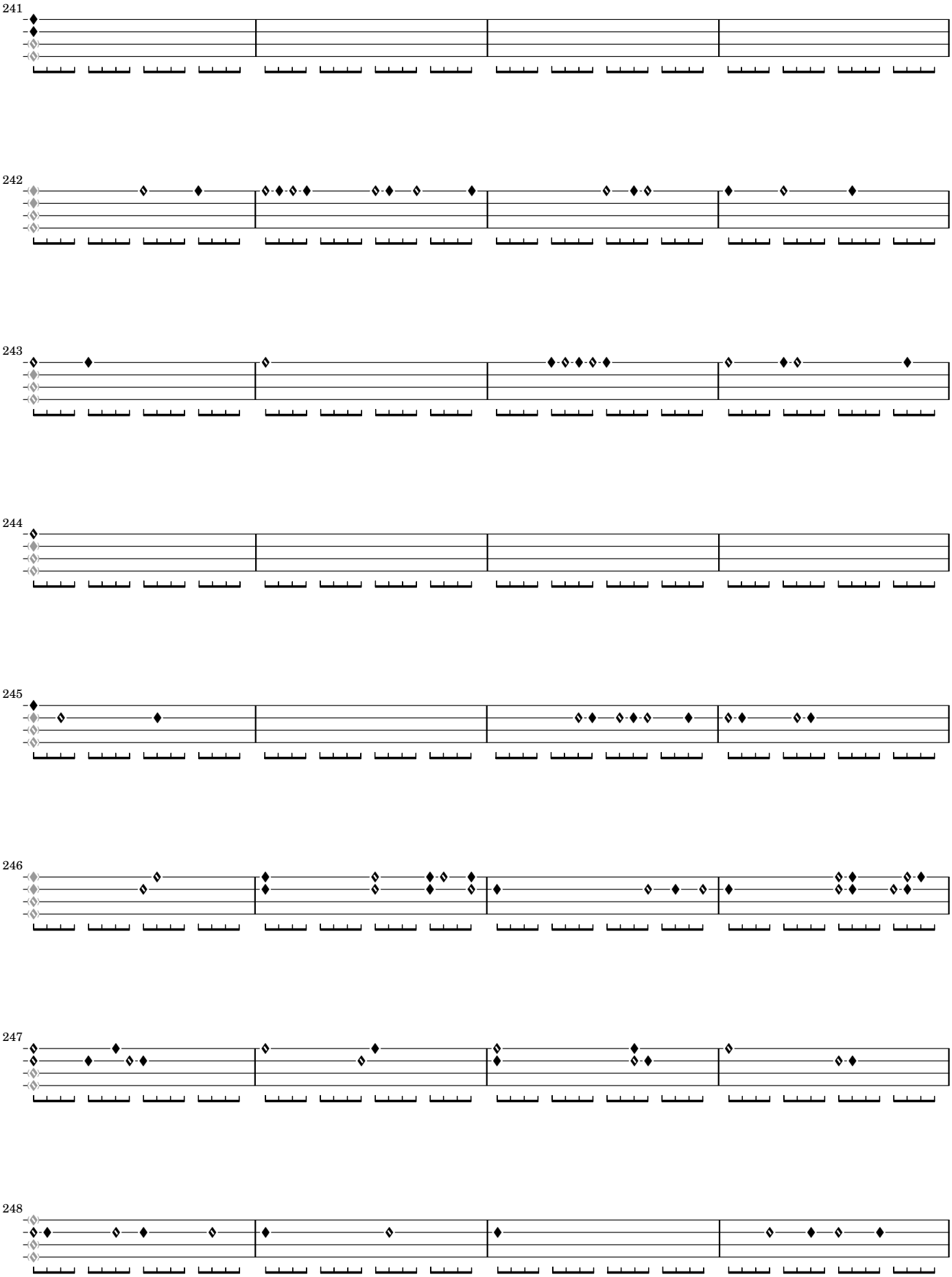

201
202
203
204
205
206
207
208

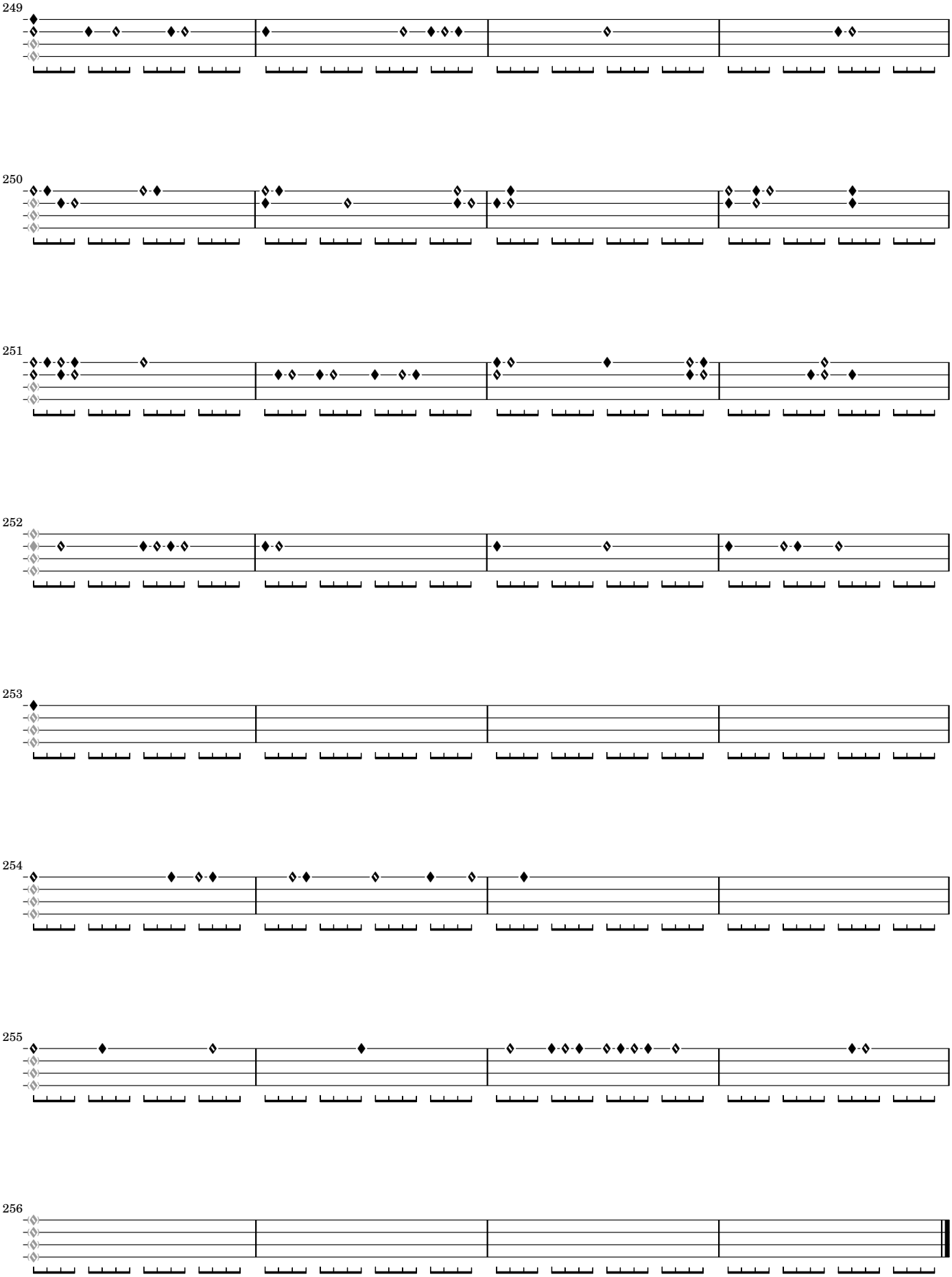

209
210
211
212
213
214
215
216


217
218
219
220
221
222
223
224


225
226
227
228
229
230
231
232


233
234
235
236
237
238
239
240


241
242
243
244
245
246
247
248


249
250
251
252
253
254
255
256

curved bow and negative - generation code and instructions

The score may be completely regenerated by running the following SuperCollider program (originally written using version 3.8.0). The first codeblock generates 4 Lilypond files (one for each line of the final score). The score can then by rendered by Lilypond (originally rendered with version 2.18.2) using the provided template from within the same directory that includes the 4 files generated by the SuperCollider program. To distinguish different generations more easily, change the "piece" variable in the Lilypond template to reflect the location and date of the generation.

The second short codeblock auditions the realization using sine tones (note that the synthesis process is actually defined within the first codeblock). As with cello, the sines tones are in a 5ths tuning where an unfilled notehead is interpreted as an open string and a filled notehead is intepreted as the 3rd harmonic of the given string. The first codeblock generates files that the second codeblock interprets. Therefore the generated files need to be within the same folder to run the audition.

SuperCollider generator

```
(
//Generate - Codeblock 1
var struct, dir, gen_code, transcribe;

//SynthDef to generate the overall structure of the piece
struct = SynthDef(\struct, {
 arg code;
 var hierarchical_dust, strings; //RandSeed.ir(0, 11734); //Seed for first realization

 hierarchical_dust = (
  Impulse.ar(8) *
  TRand.ar(0, 1, Impulse.ar(4)) >= (1/2.0) *
  TRand.ar(0, 1, Impulse.ar(2)) >= (1/4.0) *
  TRand.ar(0, 1, Impulse.ar(1)) >= (1/8.0)
 );
 hierarchical_dust = Select.ar(Impulse.ar(1), [hierarchical_dust, DC.ar(1)]);

 strings = {
  arg i;
  var sel = TIRand.ar(0, 2, hierarchical_dust) <
        Index.ar(code, (PulseCount.ar(Impulse.ar(1/8)) - 1) * 4 + DC.ar(i));
  Out.ar(i, sel);
 } ! 4;

}).asBytes(s);

//SynthDef to audition a realization
SynthDef(\play, {
 arg bufnum;
 var fund, freqs, states, strings;

 fund = 65;
 freqs = [fund, fund * 3/2, fund * 9/4, fund * 27/8];
 states = BufRd.kr(4, bufnum, Line.kr(0, BufFrames.kr(bufnum), BufFrames.kr(bufnum)/16,
     doneAction:2), 0, 1);

 strings = {
  arg i;
  var s1, s2, sel;
  s1 = SinOsc.ar(freqs[i] * 5, 0, 0.025);
  s2 = SinOsc.ar(freqs[i], 0, 0.25);
  sel = Select.ar(states[i] > 0, [s1, s2]);
 } ! 4;

 Out.ar([0,1], Mix.new(strings));
}).send(s);

```

```
//Transcribe to lilypond
dir = thisProcess.nowExecutingPath.dirname;
transcribe = {
 arg p, part;
 var file, lastItem, notes;
 ("----transcribing string " ++ p).postln;
 file = File(dir +/+ "part" ++ p ++ ".ly","w");
 part = part.collect({arg item; (item > 0).binaryValue});
 notes = ["f\'", "a\'", "c\'\'", "e\'\'"];
 lastItem = -1;
 part = part.collect({ arg item, i;
  case
  {(lastItem != item) && (item == 0)} {lastItem = item; "\\filled " ++ notes[p] ++ "16 "}
  {(lastItem != item) && (item == 1)} {lastItem = item; "\\unfilled " ++ notes[p] ++ "16 "}
  {(i % 128 == 0) && (i != 0) && (item == 0)} {lastItem = item;
        "\\once \\override NoteHead.color = #(x11-color \'grey60) \\filled \\parenthesize " ++
            notes[p] ++ "16 "}
  {(i % 128 == 0) && (i != 0) && (item == 1)} {lastItem = item;
        "\\once \\override NoteHead.color = #(x11-color \'grey60) \\unfilled \\parenthesize " ++
            notes[p] ++ "16 "}
  {lastItem == item} {lastItem = item; "s16 "}
 }).join;
 file.write("{ " ++ part ++ " \\bar \"|.\" }");
 file.close;
 if(p == 3, {"----done".postln});
};

//Generate code for parametric states
gen_code =  { ("----generate structure").postln; ({ arg i; i.asDigits(4, 4) } ! 256).flat };

//Excecute everything in NRT
Score.recordNRT(
 [[0,[\d_recv, struct],[\b_alloc, 1, 256 * 4],
  [\b_setn,1, 0, 256 * 4, gen_code.value()].flat, [\s_new, \struct, 1000, 0, 0, \code, 1]],
  [256 * 8, [\c_set, 0, 0]]],
 dir +/+ "cello_structure_osc",
 dir +/+ "cello_structure.wav",
 sampleRate: 16, headerFormat: "WAV", options: ServerOptions.new.numOutputBusChannels = 4,
  action: {
  4.do({
   arg i;
   var part;
   b = Buffer.readChannel(s, dir +/+ "cello_structure.wav", channels:[i]);
   b.loadToFloatArray(action: { arg a; transcribe.value(i, a.as(Array))});
  });
});
)

(
//Audition - Codeblock 2
var b = Buffer.read(s, thisProcess.nowExecutingPath.dirname +/+ "cello_structure.wav",
  action: {Synth(\play, [\bufnum, b]);});
)
```

Lilypond template

```
\version "2.18.2"

\paper {

  top-system-spacing =
  #'((basic-distance . 10 )
  (minimum-distance . 10 )
  (padding . 0 )
  (stretchability . 0))

  #(set-paper-size "a4" 'portrait)
  system-system-spacing =
  #'((basic-distance . 25)
  (minimum-distance . 25)
  (padding . 0)
  (stretchability . 0))

  print-page-number = ##t
  oddHeaderMarkup = \markup \fill-line { " " }
  evenHeaderMarkup = \markup \fill-line { " " }
  oddFooterMarkup = \markup {
  \fill-line {
  \on-the-fly #not-first-page
  \concat {
  "-"
      \fontsize #1.5
      \on-the-fly #print-page-number-check-first
      \fromproperty #'page:page-number-string
   "-"
      }
    }
  }
  evenFooterMarkup = \markup {
  \on-the-fly #not-first-page
    \fill-line {
  \concat {
  "-"
      \fontsize #1.5
      \on-the-fly #print-page-number-check-first
      \fromproperty #'page:page-number-string
   "-"
      }
    }
  }
}

\header {
  title = \markup { \normal-text \italic {curved bow and negative}}
  %change the following line if regenerating the score
  piece = "Medellin, Colombia; May 12, 2017"
  tagline = ""
}

#(set-global-staff-size 16)

\layout {
  indent = 0.0\cm
  \context {
      \Staff
      \override StaffSymbol.line-count = #4
      \override StaffSymbol.staff-space = #1.8
      \override NoteHead.style = #'petrucci
      \override Beam.beam-thickness = #0.2
```

```
      \override Beam.positions = #'(-2.75 . -2.75)
      \override Stem.direction = #DOWN
      \override Stem.stemlet-length = #0.75
      \override ParenthesesItem.color = #(x11-color 'grey60)
      \override ParenthesesItem.padding = #-0.3
      \remove "Clef_engraver"
      \remove "Time_signature_engraver"
    }
}

filled = {
  \once \override NoteHead #'stencil = #ly:text-interface::print
  \once \override NoteHead #'text = \markup {
  \combine
  \with-color #white
  \scale #'(1.5 . 1)
  \translate-scaled #'(-0.2 . 0)
  \musicglyph #"noteheads.s2petrucci"
  \musicglyph #"noteheads.s2petrucci"
  }
}

unfilled = {
  \once \override NoteHead #'stencil = #ly:text-interface::print
  \once \override NoteHead #'text = \markup {
  \combine
  \with-color #white
  \scale #'(1.5 . 1)
  \translate-scaled #'(-0.2 . 0)
  \musicglyph #"noteheads.s2petrucci"
  \musicglyph #"noteheads.s0petrucci"
  }
}

{
  \new Score
  \with {
    proportionalNotationDuration = #(ly:make-moment 1 16)
  }
  <<
  \new Staff
  \with {
    \remove "Stem_engraver"
  }
    <<
      \repeat unfold 256 { \repeat unfold 4 { s1 s1 \bar "|" } }
      \repeat unfold 256 { \repeat unfold 7 { s1 \noBreak } s1 \break }
      \repeat unfold 256 { \repeat unfold 16 { \hide  e'8[ \hide r8 \hide r8 \hide r8 ] } }

      \time 16/2
      \new Voice \with { \remove "Stem_engraver"} \include "part0.ly"
      \new Voice \with { \remove "Stem_engraver"} \include "part1.ly"
      \new Voice \with { \remove "Stem_engraver"} \include "part2.ly"
      \new Voice \with { \remove "Stem_engraver"} \include "part3.ly"
    >>
  >>
}
```