Modes of internal symmetry - taking a closer look at Messiaen's modes of limited transposition

Stefaan.Himpe@gmail.com

06/March/2014 - last updated: Friday 14th March, 2014 at 22:18

1 Introduction

Before we start, make sure to get the latest version of this document from https://github.com/shimpe/mints . If you are unsure what to do on that page, click "Download ZIP". It will download an archive containing all the files required to recreate this document. In the *out* folder is a document.pdf file which corresponds to the article itself. In the *output* folder are .pdf and .midi files of the modes listed in sections 8, 9, 10, 11, 12, 13 and 14. Feel free to send your questions and remarks with respect to this document to stefaan.himpe@gmail.com. Feel free to report errors, or corrections, additions, to https://github.com/shimpe/mints/issues

After reading Olivier Messiaen's book "The technique of my Musical Language" I started wondering how exactly his "modes of limited transposition" create the sound they create. Major questions for me became:

- What is so special about the property of *limited transposition* that makes the music derived from it work so well?
- Is music derived from a random subset of a chromatic scale on some sense "crippled" compared to music written based on Messiaen's modes? Can we find some kind of intuitive explanation why Messiaen's modes work well, arguably better than other non-diatonic modes? Can we use this insight to propose other modes that perhaps do not have limited transposition, but still might result in an interesting, fresh, sound?

I set out to do a series of experiments and came up with some insights that I intend to explain here. I apologize in advance if what I'm about to describe is already well-known and obvious to more informed readers. Please accept that I'm writing this down only to further my own understanding of the matter. I do not claim to have found something new.

2 Symmetry? What symmetry?

Messiaen himself explains how his modes can be thought of as consisting of *symmetrical* groups of notes. The exact nature of this symmetry was not entirely clear to me. In fact, at first sight it seemed more like repeating patterns than symmetry, and this is what I wanted to clarify.

As a starting point I set out to systematically enumerate all modes derived from a chromatic scale that have intervals symmetrically distributed around the f# (the middle note of the chromatic scale starting on c). Note that during the experiments I don't directly take into account any property of *limited transposition* but I will find back many of Messiaen's modes by considering only symmetry arguments anyway.

2.1 Systematic enumeration

The first thing to explain is how the modes with intervals symmetrically distributed around f# can be systematically enumerated.

I started from a chromatic scale. On that scale I added symbols under each note. Note how the symbols that occur left of f# return later on to the right of f#. This is important to keep symmetry of the intervals around f#, as will hopefully become clearer in the next step. For now, remember that the symbols under the complete chromatic scale form a *palindrome*, i.e. if you read them left-to-right you get exactly the same sequence of symbols as when you read them right-to-left.



In what follows we will now construct modes by assigning values 0 or 1 to each of the symbols p,q,r,s,t,u,v. A value of 0 indicates that the note to which the symbol is attached is not to be selected from the chromatic scale while constructing a mode. A value of 1 indicates that the notes to which the symbol is attached are to be selected from the chromatic scale.



This method for constructing modes has some properties:

- Any combination of values 0,1 assigned to all of p, q, r, s, t, u, v results in a mode derived from the chromatic scale with intervals distributed symmetrically around f#.
- A different combination of values 0,1 assigned to p, q, r, s, t, u, v results in a different mode. In other words, there are no two different keys

(p,q,r,s,t,u,v) that result in the same set of notes selected from the chromatic scale.

- If we look at all possible ways that we can assign 0, 1 to p, q, r, s, t, u, v, we construct all possible subsets of the chromatic scale with intervals distributed symmetrically around f#. In other words, we don't skip any modes symmetrical around f# by using this method.
- All in all, this means that any (p,q,r,s,t,u,v) key uniquely defines one mode with intervals symmetrically distributed around f#.
- There are $2^7 = 128$ different ways to assign the values 0, 1 to the variables p, q, r, s, t, u, v. This means that there are 128 unique modes with symmetrical distribution of intervals around f#. One of those modes is mode 0, which has no notes at all. We don't consider it further.

2.2 Intermezzo: binary numbers

In computer science a key like (0,1,1,0,0,1,0) can be interpreted as a binary number. Each number 1 or 0 is called a "bit". For every binary number, there's an equivalent decimal number and vice versa. In the systematic enumeration of modes in the appendices of this explanation, I use decimal equivalents of binary numbers, because they take much less typesetting space. If you want to make sense of the explanations that follow, it's useful to understand how binary numbers relate to decimal numbers, as explained now:

2.2.1 Conversion from binary number to decimal number

In order to convert from a binary number to a decimal number, one writes powers of two underneath the binary number and then multiplies and adds the results. As an example, consider conversion of number (0,1,1,0,0,1,0) to decimal:

Note that if I add extra 0's to the left of a binary number, its value doesn't change. The same is true for a decimal number: if you write 6 or you write 06 you really have the same number.

2.2.2 Conversion from a decimal number to a binary number

In order to convert a decimal number back to a binary number one keeps on dividing the number by 2, and notes down the rest after division. As an example, consider converting 50 back to binary representation:

- \bullet We start from 50
- divide 50 by 2 to get 25, with rest after division=0
- divide 25 by 2 to get 12, with rest after division=1

- divide 12 by 2 to get 06, with rest after division=0
- divide 06 by 2 to get 03, with rest after division=0
- divide 03 by 2 to get 01, with rest after division=1
- divide 01 by 2 to get 00, with rest after division=1

If you now look at the rests after division from bottom to top, you get (1,1,0,0,1,0). Remember from section 2.2.1 that one can add zeros to the left of any number without changing its value. Since we prefer to work with binary numbers (keys) of length 7 (i.e. the number of symbols p,q,...,v) we turn the binary number into key (0, 1, 1, 0, 0, 1, 0).

2.2.3 Tip about binary/decimal conversion

Practically all operating systems nowadays have some built-in calculator application that knows how to convert between binary and decimal should you ever need to do so.

3 What's the point? How do these binary numbers help us analyze music?

Using this enumeration method, I listed all the 127 modes that are symmetrical around f# as can be seen in section 8. I've sorted the modes by length. Each mode is annotated with a few numbers, e.g. 74:(2)7-71. The number 74 means it's the 74th mode in the list of modes sorted by length. The number (2) means that we're looking at modes wity binary symmetry (the mode is divided in two symmetrical parts). The number 7 means that the mode consists of 7 notes, distributed symmetrically around f#. The number 71 is the decimal number that corresponds to the binary key (p,q,r,s,t,u,v)=(1,0,0,0,1,1,1) that uniquely defines the mode:



3.1 The symmetry of Messiaen's modes around f#

A first thing that struck me as interesting is that all of Messiaen's modes can be found back in the list of 127 modes enumerated in section 8 (albeit not always in first transposition, which is caused by our restriction to only look at modes symmetrical around f#, and not to look at modes symmetrical around - say - g.):

• Messiaen's mode 1 corresponds to our mode (2)85 (p,q,r,s,t,u,v) = (1,0,1,0,1,0,1)



• Messiaen's mode 2 corresponds to our mode (2)54 (p,q,r,s,t,u,v) = (0,1,1,0,1,1,0)



• Messiaen's mode 3 corresponds to our mode (2)93 (p,q,r,s,t,u,v) = (1,0,1,1,1,0,1)



• Messiaen's mode 4 corresponds to our mode (2)103 (p,q,r,s,t,u,v) = (1,1,0,0,1,1,1)



• Messiaen's mode 5 corresponds to our mode (2)99 (p,q,r,s,t,u,v) = (1,1,0,0,0,1,1)



• Messiaen's mode 6 corresponds to our mode (2)107 (p,q,r,s,t,u,v) = (1,1,0,1,0,1,1)



• Messiaen's mode 7 corresponds to our mode (2)119 (p,q,r,s,t,u,v) = (1,1,1,0,1,1,1)



3.2 More symmetry in Messiaen's modes

Now take a close look at the binary keys for Messiaen's modes. In all but one of his modes, the binary keys are themselves palindromes, indicating that not only are the modes symmetrical around f#, but they also have extra internal symmetry around d# (left-hand side of f#) and around a (right-hand side of f#). Having a palindromic key thus implies that there's a second level of symmetrical interval distribution inside the upper and lower halves of the modes.

It would not be correct to say that only modes with palindromic binary keys sound good. Look e.g. at mode of binary internal symmetry number (2)90 (1,0,1,1,0,1,0), which is better known as "c dorian". The binary key is not

palindromic but it does show other extra symmetries in its structure (observe how its binary key becomes a palindrome if you leave out the last 0). Similarly, Messiaen's fourth mode of limited transposition also becomes a palindrome if you leave out the last 1 in its binary key, indicating that it has significant internal symmetries beyond the always present symmetry around f#.

I'd like to speculate here that additional internal symmetries play an important role in making modes (and their "modes") sound good. The human brain is optimized for pattern matching. It is sensitive to symmetries and the listener probably subconsciously picks up the patterns and hears the symmetries present in the intervals that make up the mode. (If that were true, we might even nominate the "Dorian" mode as the most natural of modes based on a diatonic scale.) Modes with extra internal symmetries look like good candidates for harmony and melody experiments.

3.3 Are there any modes with similar internal symmetries that are not Messiaen modes?

3.3.1 Perfect internal symmetry

In section 9 all modes are listed that have palindromic binary keys. Note that almost all of Messiaen's modes appear here in one form or another, and that some modes appear that are not part of Messiaen's musical language. This is because we used *internal symmetry* instead of *limited transposition* as criterion. From these results, it's quite clear that there's a close connection between the two criteria. It's also interesting that a number of (shorter) modes appear which, to the best of my knowledge, were not used directly by Messiaen, but which may be interesting for further harmonic and melodic experiments.

In what follows, remember that the notation "mode (2)x-y" means "a mode with binary symmetry, x notes and key y, where y should be converted to binary to see which notes are present in the mode. Compare the descriptions given here to the modes as listed in section 9.

- Mode (2)02-008 (0,0,0,1,0,0,0) is a ditonic mode. Two notes may be a bit limited for a composition. Interesting though that it is a tritone, which is one of the basic building blocks for Messiaen's musical language.
- Mode (2)04-020 (0,0,1,0,1,0,0) is a tetratonic mode.
- Mode (2)06-028 (0,0,1,1,1,0,0) is a hexatonic mode.
- Mode (2)04-034 (0,1,0,0,0,1,0) is a tetratonic mode. Interesting about this mode is that there's even more symmetry present in the lower and upper half of the binary key. This is a third level of symmetry in the mode.
- Mode (2)06-042 (0,1,0,1,0,1,0) is a hexatonic mode. Interesting about this mode is that there's even more symmetry present in the lower and upper half of the binary key. This is a third level of symmetry in the mode.
- Mode (2)08-054 (0,1,1,0,1,1,0) is an octotonic mode. It's also known as Messiaen's second mode of limited transposition.

- Mode (2)10-062 (0,1,1,1,1,1,0) is a decatonic mode. This mode is not listed by Messiaen. However, if we extend this mode with a c# at the right, we get a "mode" of Messiaen's seventh mode of limited transposition built on note d.
- Mode (2)03-065 (1,0,0,0,0,0,1) is a tritonic mode. It consists of 2 tritones.
- Mode (2)05-073 (1,0,0,1,0,0,1) is a pentatonic mode.
- Mode (2)07-085 (1,0,1,0,1,0,1) is a heptatonic mode. It's also known as Messiaen's first mode of limited transposition, or as the whole-tone scale. Interesting about this mode is that there's even more symmetry present in the lower and upper half of the binary key. This is a third level of symmetry in the mode.
- Mode (2)09-093 (1,0,1,1,1,0,1) is a nonatonic mode. This is Messiaen's third mode of limited transposition. Interesting about this mode is that there's even more symmetry present in the lower and upper half of the binary key. This is a third level of symmetry in the mode.
- Mode (2)07-099 (1,1,0,0,0,1,1) is a heptatonic mode. This is Messiaen's fifth mode of limited transposition.
- Mode (2)09-107 (1,1,0,1,0,1,1) is a nonatonic mode. This is a "mode" of Messiaen's sixth mode of limited transposition built on note c#.
- Mode (2)11-119 (1,1,1,0,1,1,1) is an undecatonic mode. This is a "mode" of Messiaen's seventh mode of limited transposition built on note b. Interesting about this mode is that there's even more symmetry present in the lower and upper half of the binary key. This is a third level of symmetry in the mode.
- Mode (2)13-127 (1,1,1,1,1,1) is the chromatic scale itself. This mode also has a third and even fourth level of symmetry.

3.3.2 Partial internal symmetry

Now follows a list of modes that are partially palindromic as follows: the binary keys of the modes listed here become palindromic if you leave out either the first or last bit. They are listed in musical form in section 10. These form another subset of modes (slightly less "perfect" than the modes in the previous section). Probably some of these modes are better known under other names, but my knowledge of existing scales is not large enough to recognize all of them.

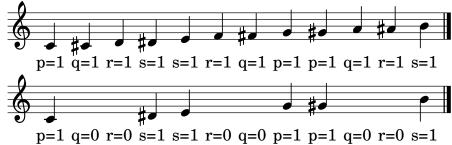
By reducing the constraints on symmetry also other interesting modes can be selected (e.g. modes that are a palindromic if you leave out 2 outer bits), but listing those is left as an exercise to the interested reader.

- (2)001 (0,0,0,0,0,0,0,1) is a mode consisting of a single note f#. This is a bit limited to compose with :)
- (2)012 (0,0,0,1,1,0,0) tetratonic
- \bullet (2)018 (0,0,1,0,0,1,0) tetratonic

- (2)024 (0,0,1,1,0,0,0) tetratonic
- (2)025 (0,0,1,1,0,0,1) tetratonic. Sounds quite exotic (arabic?).
- (2)030 (0,0,1,1,1,1,0) octatonic
- (2)033 (0,1,0,0,0,0,1) tritonic
- (2)036 (0,1,0,0,1,0,0) tetratonic
- (2)037 (0,1,0,0,1,0,1) pentatonic
- (2)045 (0,1,0,1,1,0,1) heptatonic. This is c# aeolian mode (natural minor diatonic scale).
- (2)051 (0,1,1,0,0,1,1) heptatonic. Sounds quite exotic (arabic?)
- (2)060 (0,1,1,1,1,0,0) octatonic
- \bullet (2)061 (0,1,1,1,1,0,1) nonatonic
- (2)063 (0,1,1,1,1,1,1) undecatonic. Like a chromatic scale but without note c.
- (2)064 (1,0,0,0,0,0,0,0) ditonic. Consists of only notes c.
- (2)066 (1,0,0,0,0,1,0) tetratonic. Consists of the notes of a sus4 chord built on c.
- (2)067 (1,0,0,0,0,1,1) pentatonic.
- (2)076 (1,0,0,1,1,0,0) hexatonic.
- (2)082 (1,0,1,0,0,1,0) hexatonic.
- (2)090 (1,0,1,1,0,1,0) octatonic. This is really just the dorian mode of c.
- \bullet (2)091 (1,0,1,1,0,1,1) nonatonic. Like the dorian mode of c but with f# added.
- \bullet (2)094 (1,0,1,1,1,1,0) decatonic.
- \bullet (2)097 (1,1,0,0,0,0,1) pentatonic.
- (2)102 (1,1,0,0,1,1,0) octatonic. Sounds quite exotic (arabic?).
- (2)103 (1,1,0,0,1,1,1) is the fourth mode of limited transposition of Messiaen.
- \bullet (2)109 (1,1,0,1,1,0,1) nonatonic. Left halve sounds darker than right halve.
- (2)115 (1,1,1,0,0,1,1) nonatonic.
- \bullet (2)126 (1,1,1,1,1,1,0) dodecatonic. Like a chromatic scale, but with f# left out.
- (2)127 (1,1,1,1,1,1,1) is the chromatic scale. This is the only key that also appears in the fully palindromic modes.

4 Ternary symmetries

So far we've only considered symmetries that divide the chromatic scale in a symmetrical left half and a right half. But a chromatic scale consists of 12 half tones, and therefore it can also be divided in groups of 3. The way of working remains more or less the same. First we propose an enumeration scheme. Note that to find ternary symmetries, we leave out the repeated tonic at the end of the mode. We then need a 4-bit binary key. Note how the mode now is divided in three parts $p, q, r, s \rightarrow s, r, q, p \rightarrow p, q, r, s$. This unfolded key (p,q,r,s,s,r,q,p,p,q,r,s) is not a palindrome anymore. This is a fundamentally different form of symmetry, namely between d#,e and between g,g#.



We can now reuse the knowledge we gathered before: given that we use 4-bit binary keys, there must be $2^4=16$ such modes (of course mode 0 has no notes, so we don't consider it further). As before we can translate the binary keys to decimal numbers, but the decimal denote a different mode than the same decimal numbers we used while examining binary symmetries. The difference lies in the structure of the unfolded key. In the binary symmetry case we had a key (p,q,r,s,t,u,v) that after unfolding becomes (p,q,r,s,t,u,v,u,t,s,r,q,p). Now in the ternary case we have a key (p,q,r,s) that after unfolding becomes (p,q,r,s,s,r,q,p,p,q,r,s). To distinguish the decimal numbers for modes with binary symmetry from the decimal numbers for modes with ternary symmetry, we preceded the former ones with (2) and we precede the latter ones with (3).

Sections 15, 16, 17, 18, 19 list all 16 modes with ternary symmetry.

5 Deriving harmonies from modes of internal symmetry

Another question I was struggling with was why Messiaen chose to build chords based on the interval of a fourth. During my investigation I think I saw a possible explanation as follows:

- Sections 11, 12, 13, 14 systematically list the chords built on the modes of binary internal symmetry by taking notes from consecutive mode (scale) degrees, every third mode degree, every fourth mode degree, every fifth mode degree respectively. Sections 16, 17, 18 and 19 do the same for modes of ternary internal symmetry.
- As is clearly visible, the fewer notes are in a mode that is symmetrical around f#, the more widely spaced those notes are (on average).

- In wider spaced modes (many 0's in the binary key), it makes sense to build chords from mode degrees that are close enough to each other.
- In more dense modes (many 1's in the binary key), it makes sense to build chords from mode degrees that are spaced further apart. If we use notes too close together, every chord sounds very dissonant and there's not much room for creating harmonic contrasts.
- Since Messiaen's modes are relatively dense compared to say a diatonic scale (Messiaen's modes all have 8 or more notes, whereas a diatonic scale has 7 notes), my guess is that the harmonies as built using every third note sound a bit too harsh, and Messiaen therefore decided to to use fourths.

It would seem that when deriving harmonies from a set of notes we want to avoid uneven spreading of intervals over the mode, to avoid ending up with a series of very closely spaced chords, followed by a series of much more widely spaced chords. In other words: a mode that consists of a left part with many half tones, and a right part with many whole tones will tend to result in much more dissonant chords derived from that left part, and much more open sounding chords derived from the right part. Can the binary keys can help us see in which modes the intervals are better spread out than in other modes?

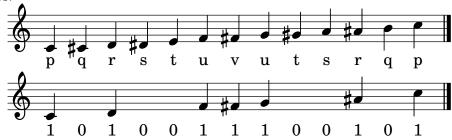
5.1 From binary key to intervals in symmetric modes

First we can convert the binary keys back to intervals between successive notes in the mode. Doing so is simple once we think back of what the binary keys really mean: each 1 or 0 is an inclusion respectively exclusion of a particular note from the chromatic scale built on c. We can find back the intervals in the mode as follows:

- Write down the binary key (p,q,r,s,t,u,v) and unfold it to from the complete mode: (p,q,r,s,t,u,v,u,t,s,r,q,p). Or in the case of a ternary symmetrical mode, take the key (p,q,r,s) and unfold it to (p,q,r,s,r,q,p,q,r,s).
- Now each time count the number of 0's between successive 1's. Each count is one less than the number of half steps in the interval.

An abstract description like the above begs for an example. Let's take mode

83:



Number (2)83 in binary is (1,0,1,0,0,1,1). After unfolding we get (1,0,1,0,0,1,1,1,0,0,1,0,1) (see music example). Now we count zeros between successive 1's. We find 1 zero, 2 zeros, 0 zeros, 0 zeros, 2 zeros, 1 zero:

If we count zero 0's, then the interval between successive notes is 1/2 tone. If we count one 0, the interval is 1 tone. In general, if z is the number of zeros

count 0's between successive 1's

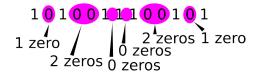


Figure 1: Turning binary keys back into intervals

we counted, the number of half tones in the interval is h = (z + 1). And then the number of tones in the interval is t = h/2 = (z + 1)/2. In the case of mode (2)83, we apply this formula as follows:

- we counted 0's between successive 1's: (1,2,0,0,2,1).
- we apply the formula: ((1+1)/2, (2+1)/2, (0+1)/2, (0+1)/2, (2+1)/2, (1+1)/2) = (1, 1.5, 0.5, 0.5, 1.5, 1).
- so we have a mode with intervals 1 tone, 1.5 tones, 0.5 tones, 0.5 tones, 1.5 tones, 1 tone
- we double check with the musical example and see indeed that $c \to d = 1$ tone, $d \to f = 1.5$ tones, $f \to f\# = 0.5$ tones, $f\# \to g = 0.5$ tones, $g \to a\# = 1.5$ tones, $a\# \to c = 1$ tone

5.2 Ideal interval size. Avoiding clusters and gaps

Intuitively, when one wants to derive harmonies by stacking notes from every n-th scale degree, one could decide to avoid too small intervals (clusters) and too large intervals (gaps), e.g. to avoid a concentration of too dissonant chords on one side of the scale. For this reason it's useful to think about what intervals one reasonably can expect to occur in a mode.

Suppose you create a mode of only 2 different notes. These 2 notes have to span an octave, i.e. 12 half tones. To have maximal spreading of the notes over the mode one should have intervals between the notes of 6 half tones $(c \to f\#)$. The interval $(f\# \to c)$ closes the construction by repeating the first note c, so this second "c" it is not a "third note" in our mode. More general, for n notes spanning an octave (12 half tones), the ideal interval has 12/n half tones.

Suppose this assumption makes sense, then in an octotonic scale (8 distinct notes) the ideal interval between notes is 12/8 = 1.5 half tones. There's one problem with this: 1.5 half tones leads to microtonal music. If we don't want to go there, we need to do the next best thing: round to multiples of 1. We can't round all 1.5 half tone intervals to 2 half tone intervals because then we end up with way too many half tones to fit in an octave. Similarly we can't round all 1.5 intervals down to 1 because then we end up with too few half tones to span the octave. We can however approximate this 1.5 by alternating between rounding 1.5 up to 2 and rounding 1.5 down to 1 to get the following configuration of half tone intervals: (1,2,1,2,1,2,1,2) which nicely adds up to

1+2+1+2+1+2+1+2=12 half tones over 8 notes. But now look closely at what we really constructed...

We find Messiaen's second mode of limited transpostion! Even though I didn't present a rigorous mathematical proof here, I hope it is clear that Messiaen's second mode of limited transposition is a best possible approximation of evenly spread intervals in an octotonic scale (best possible if we don't allow microtonal intervals).¹

When we select modes for our own compositions, we may want to pay some attention to avoiding clusters and gaps in the intervals that lie between the notes in the modes we select. An interval G can be named a "gap" if it has more half tones than the ideal interval rounded up, and an interval C can be named a "cluster" if it has less half tones than the ideal interval rounded down. In all this, the ideal interval is itself a function of the number of notes in the mode. Summarizing:

- The ideal interval I(n) in a mode is I(n) = 12/n where n is the number of distinct notes in the mode.
- An interval G is a gap if $B > \lceil I(n) \rceil$. (The symbols \lceil and \rceil should be read as "round up")
- An interval C is a cluster if $S < \lfloor I(n) \rfloor$. (The symbols \lfloor and \rfloor should be read as "round down")

6 Conclusions

To summarize:

- First we systematically listed all modes with internal symmetry around f# derived from a chromatic scale. We found that there are 127 such modes.
- Then, we saw how all of Messiaen's modes are part of this list of 127 modes, meaning that they at least have some symmetry in interval distribution.
- After that, we noticed how all but one of the Messiaen modes contain extra symmetries in the lower half and upper half of the mode (palindromic keys). We listed all modes that have both symmetry around f# and the extra symmetries in the lower half and upper half of the mode, and discovered some potentially interesting modes not directly used by Messiaen (with fewer notes). Even though we didn't care about the property of limited transposition, by just considering symmetry arguments we arrived at a very similar set of modes.
- While doing so, we also formulated a possible explanation for why Messiaen may have chosen to build his chords using fourths, and we speculated about how internal symmetry and even spreading of intervals may contribute to make Messiaen's modes work better than other, randomly chosen, modes.

¹This makes me wonder how octotonic scales sound if we do allow microtonal intervals.

7 Some resources

While investigating I made extensive use of Jackson Hardaker's Messiaen mode visualizer: http://messiaen.jacksonhardaker.com/

The complete code required to reproduce the experiments and the text in this document (together with midi files for the listed modes and chords in the following sections) can be found online at https://github.com/shimpe/mints. To recreate this document you need at the very least the following free software:

- \bullet python 2.x
- lilypond
- LaTeX
- a .pdf viewer (I used okular, but any viewer should do).

The provided build script is written in bash. For windows or other systems, you may need to translate to an appropriate format.

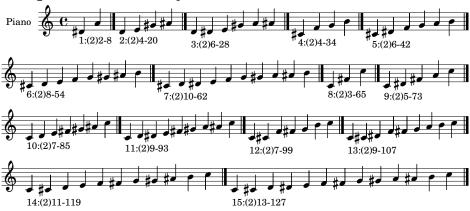
Feel free to send your questions and remarks with respect to this document to stefaan.himpe@gmail.com. Feel free to report errors, or corrections, additions, to https://github.com/shimpe/mints/issues

8 All modes with binary internal symmetry around f#, ordered by length





9 All modes of binary internal symmetry with palindromic keys



10 All modes of binary internal symmetry with partially palindromic keys



11 All chords built by stacking every second note from the modes of internal binary sym-









12 All chords built by stacking every third note from the modes of internal binary symmetry

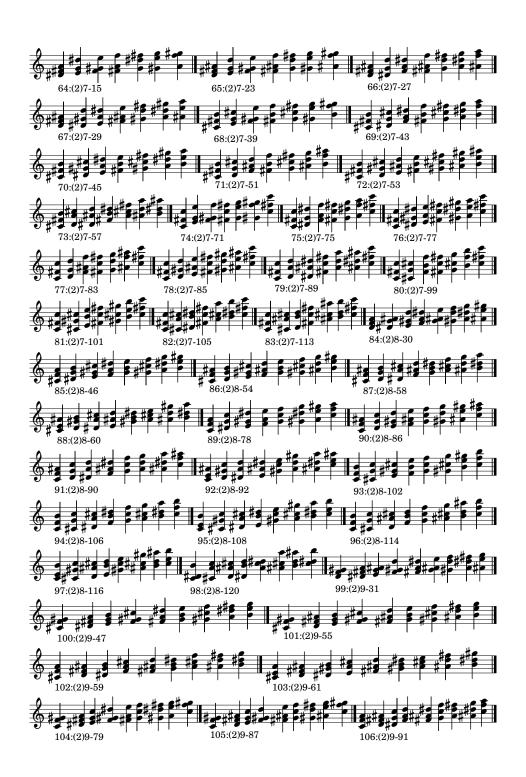






13 All chords built by stacking every fourth note from the modes of internal binary symmetry







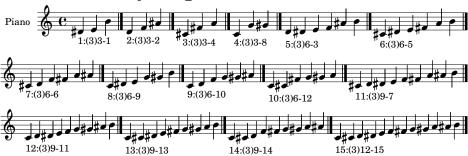
14 All chords built by stacking every fifth note from the modes of internal binary symmetry







15 All modes with ternary internal symmetry ordered by length



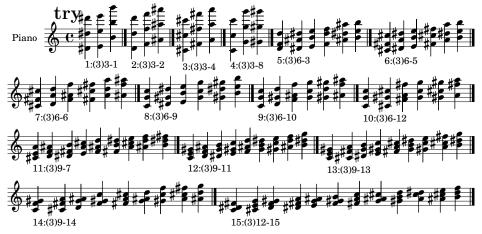
16 All chords built by stacking every second note from the modes of ternary internal sym-



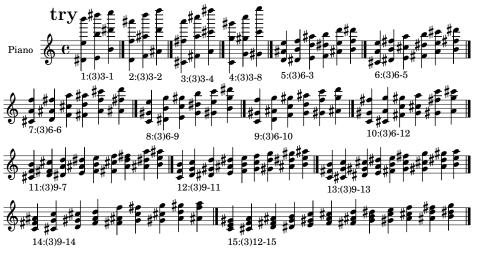
17 All chords built by stacking every third note from the modes of ternary internal symme-



18 All chords built by stacking every fourth note from the modes of ternary internal symme-

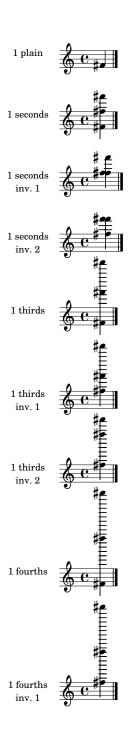


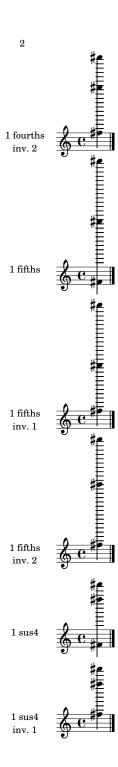
19 All chords built by stacking every fifth note from the modes of ternary internal symme-

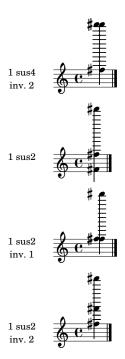


Characterizing mode (2)1

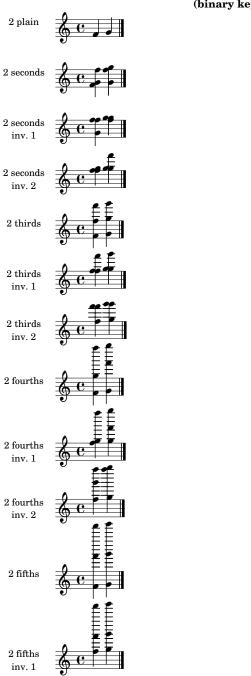
Mode (2)1 (binary key: 0000001)

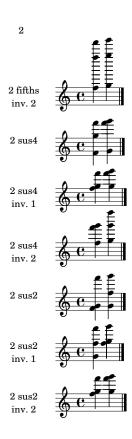




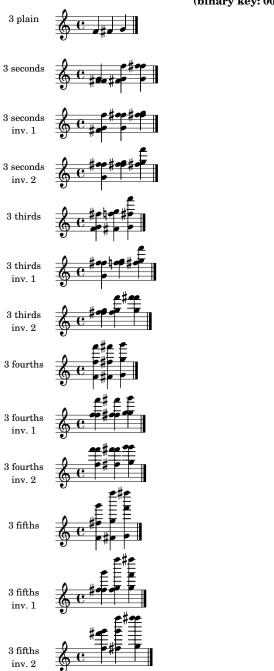


Mode (2)2 (binary key: 0000010)



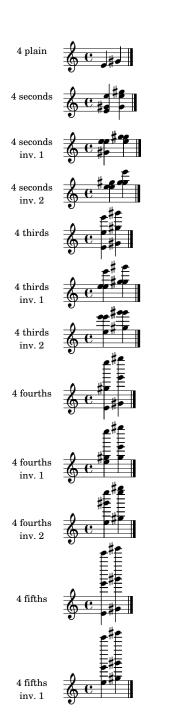


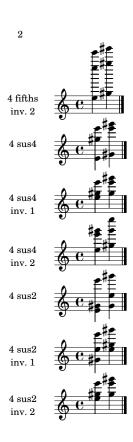
Mode (2)3 (binary key: 0000011)



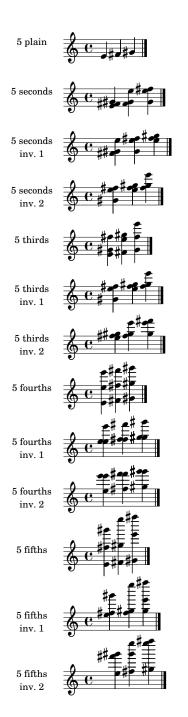


Mode (2)4 (binary key: 0000100)



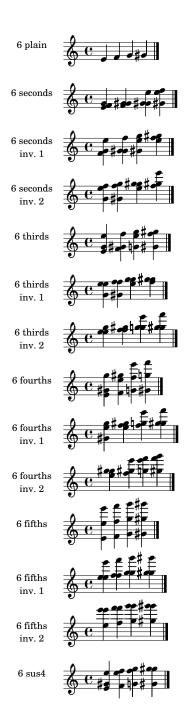


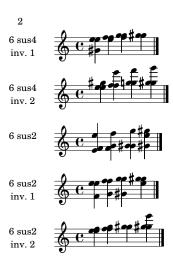
Mode (2)5 (binary key: 0000101)





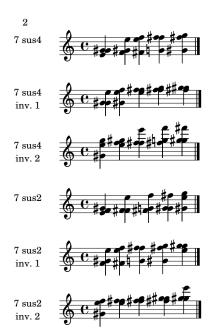
Mode (2)6 (binary key: 0000110)



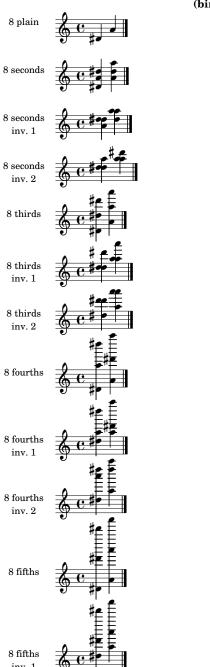


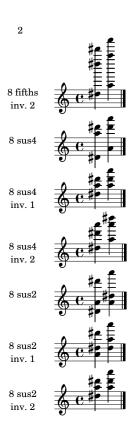
Mode (2)7 (binary key: 0000111)



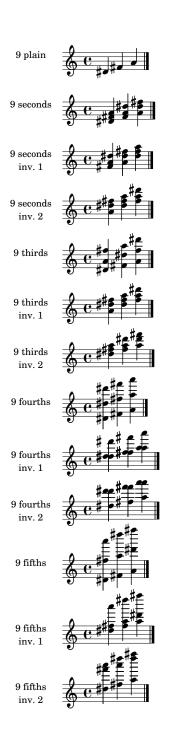


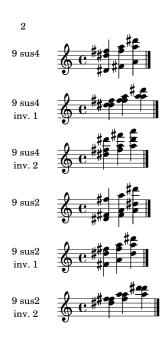
Mode (2)8 (binary key: 0001000)



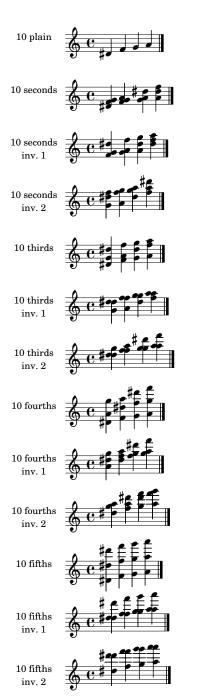


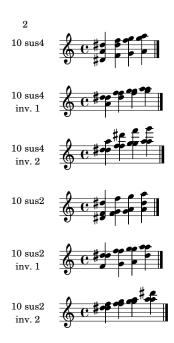
Mode (2)9 (binary key: 0001001)



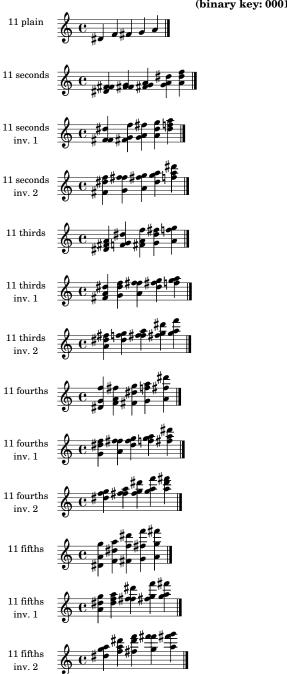


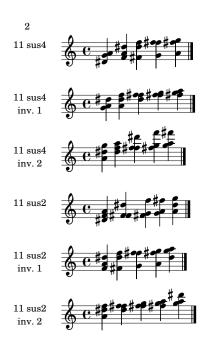
Mode (2)10 (binary key: 0001010)





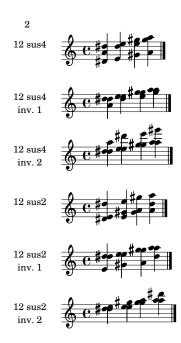
Mode (2)11 (binary key: 0001011)

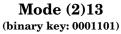


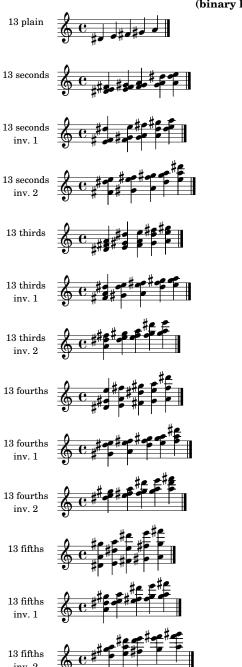


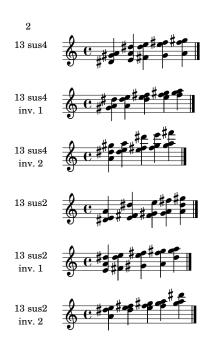
Mode (2)12 (binary key: 0001100)

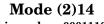


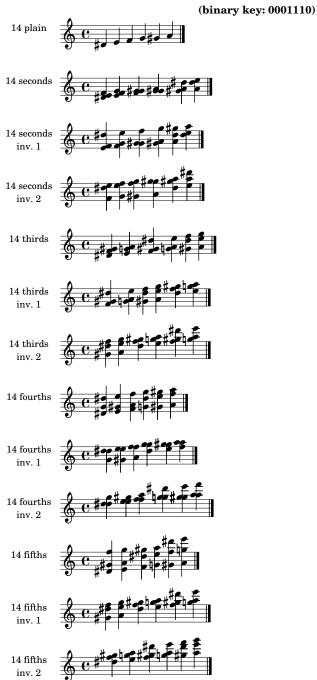


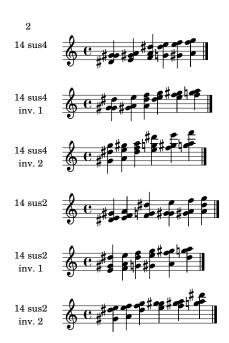




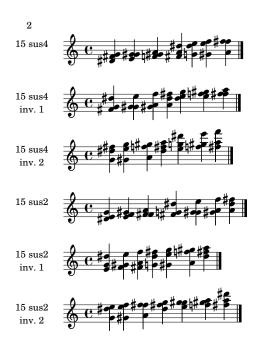




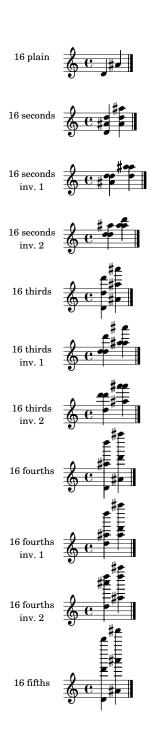


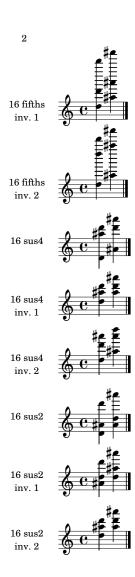






Mode (2)16 (binary key: 0010000)



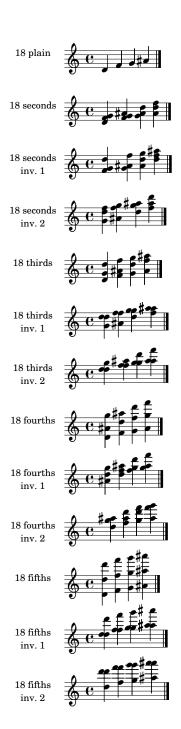


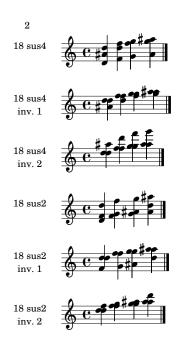
Mode (2)17 (binary key: 0010001)

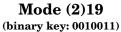


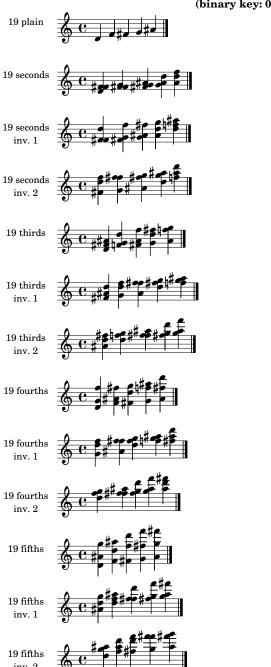


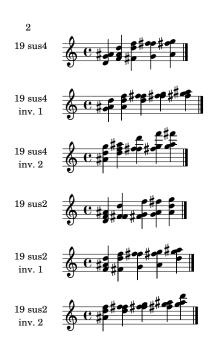
Mode (2)18 (binary key: 0010010)



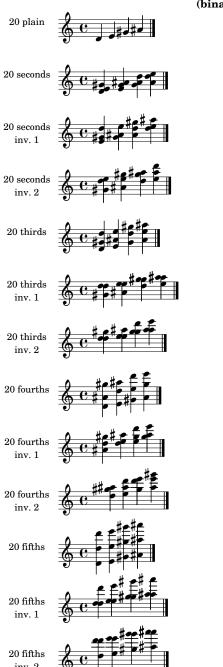


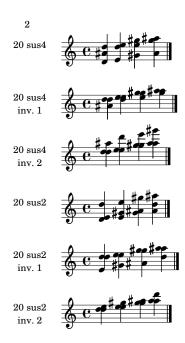


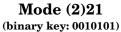


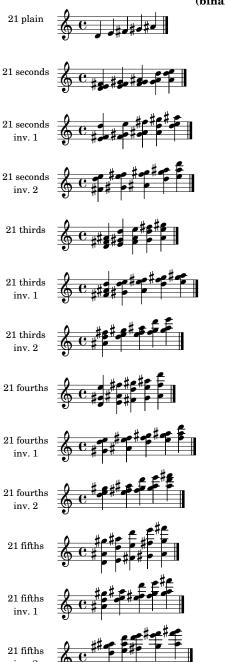


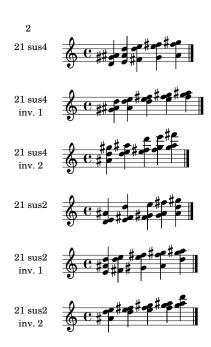
Mode (2)20 (binary key: 0010100)

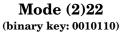


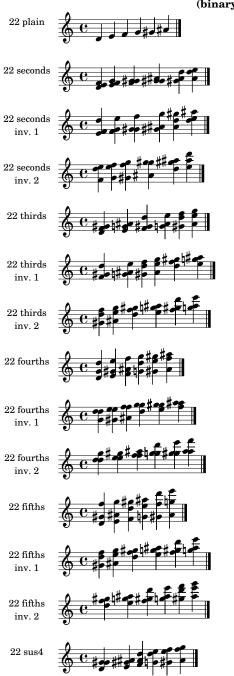


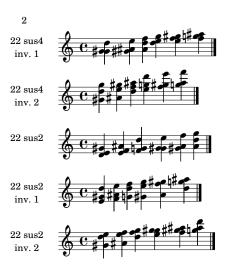




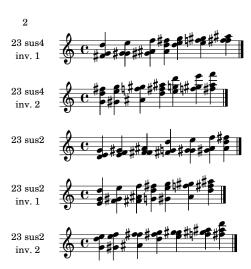


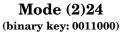


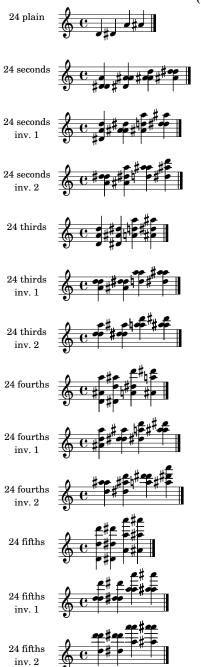


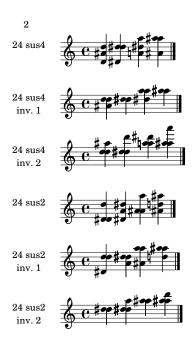


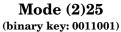


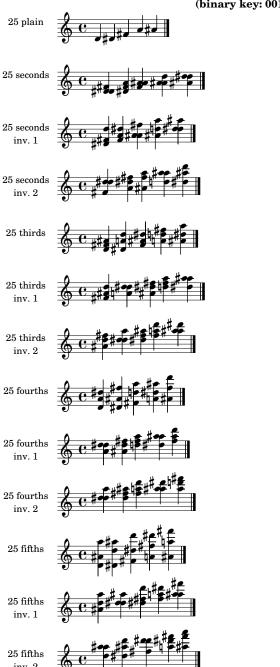


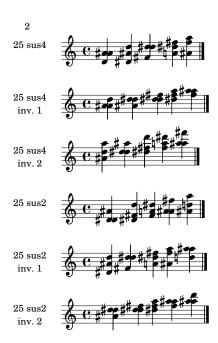


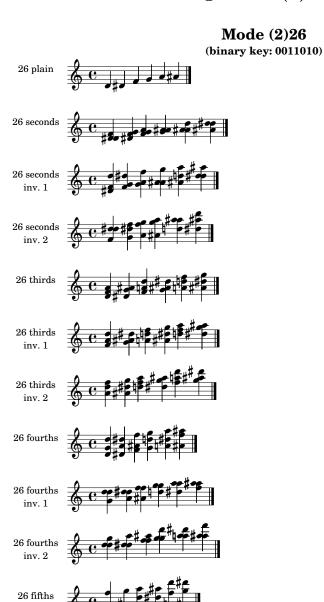




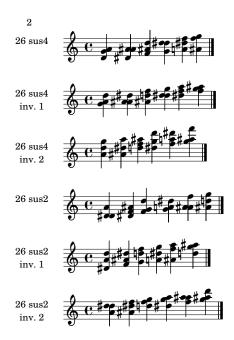


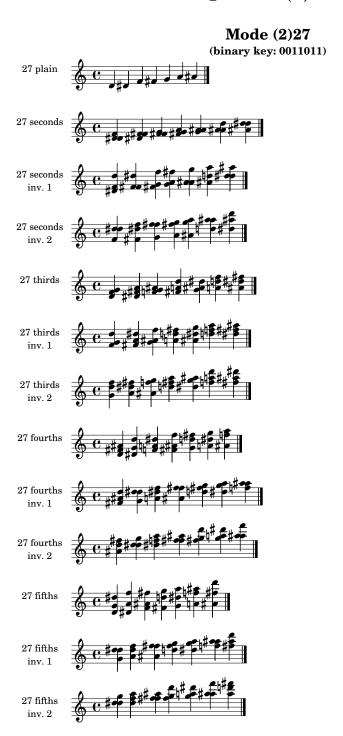




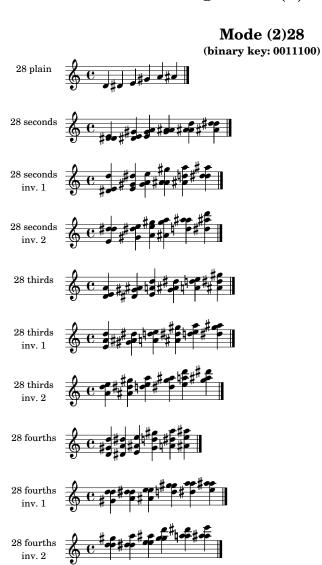


26 fifths



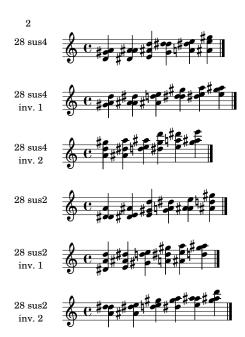


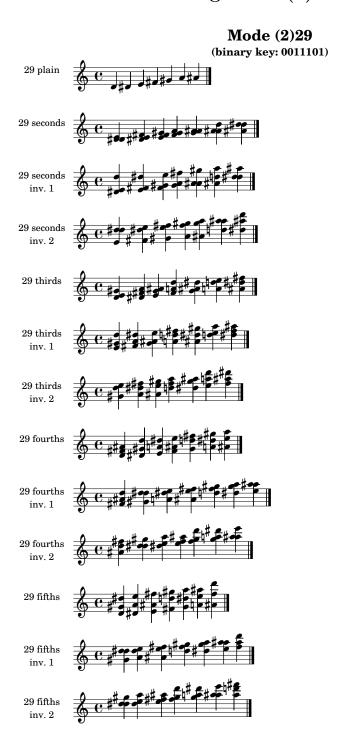


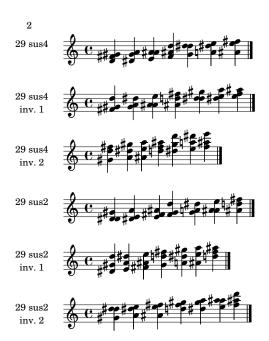


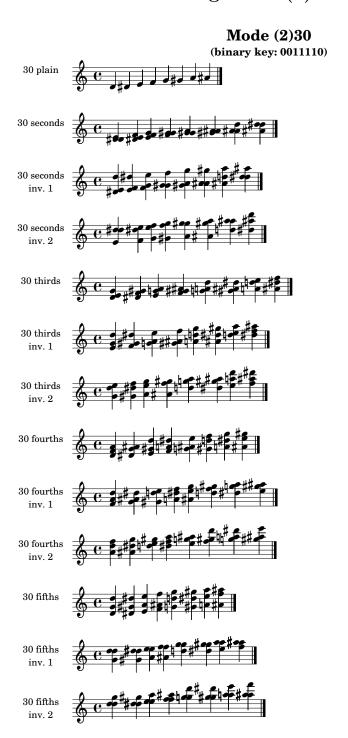
28 fifths

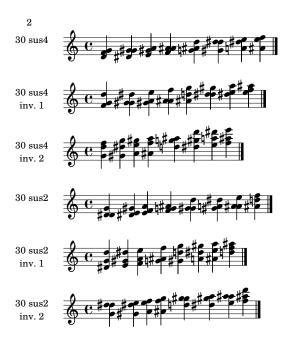
28 fifths

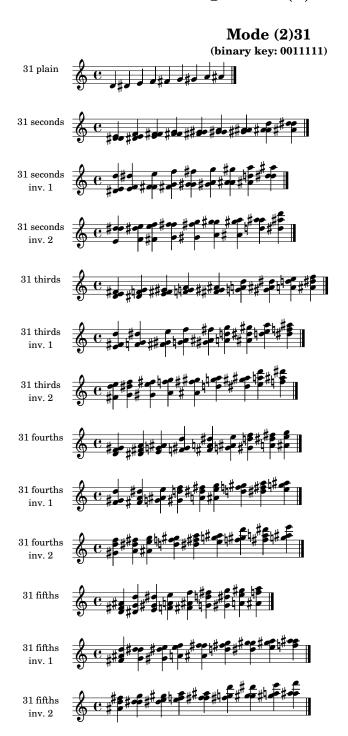


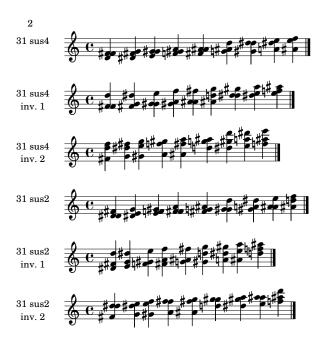




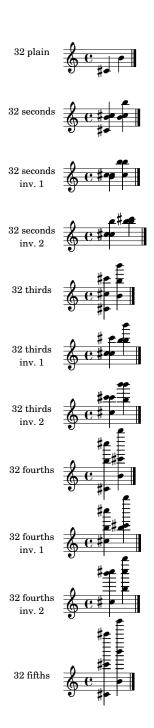


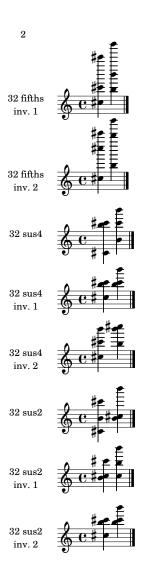




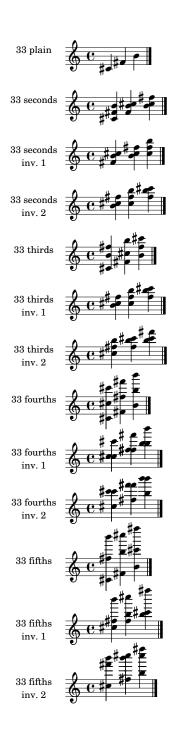


Mode (2)32 (binary key: 0100000)



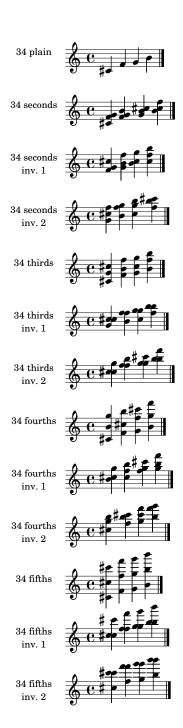


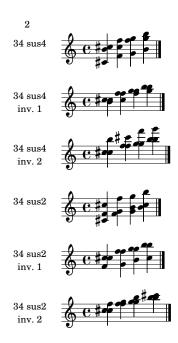
Mode (2)33 (binary key: 0100001)

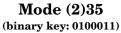


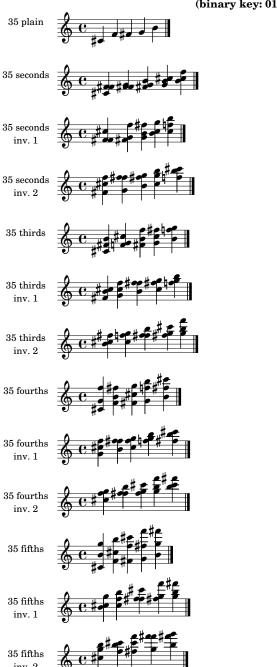


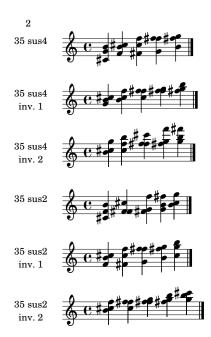
Mode (2)34 (binary key: 0100010)



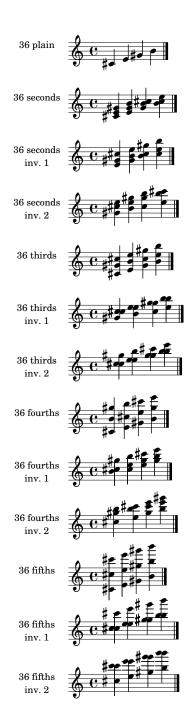


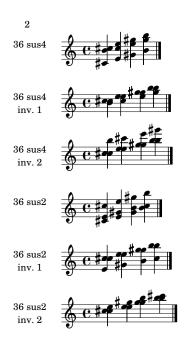


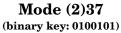


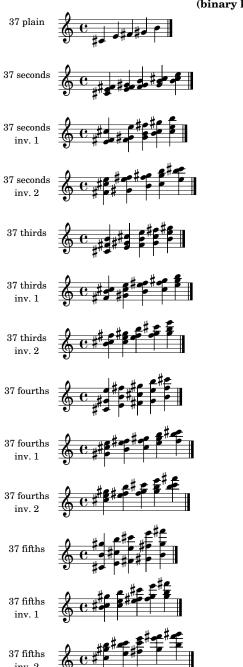


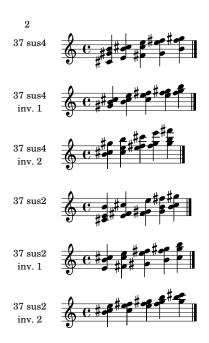
Mode (2)36 (binary key: 0100100)

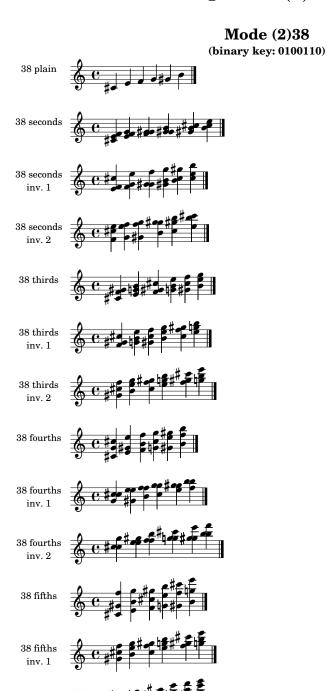


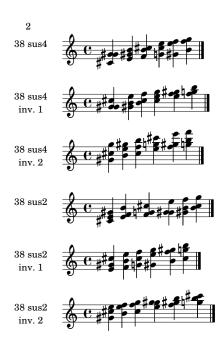


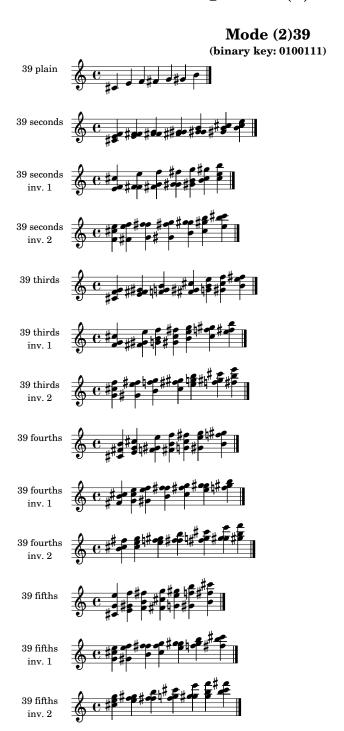


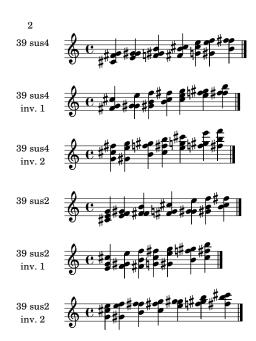




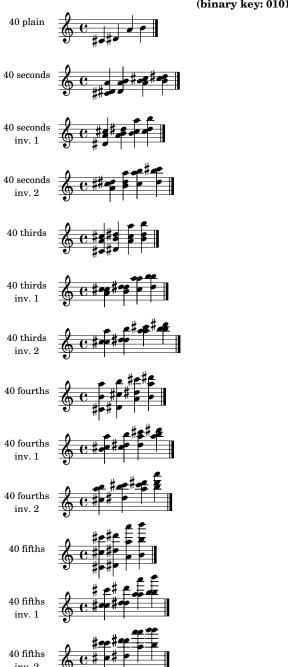


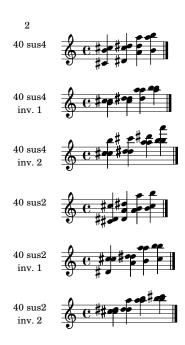


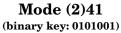


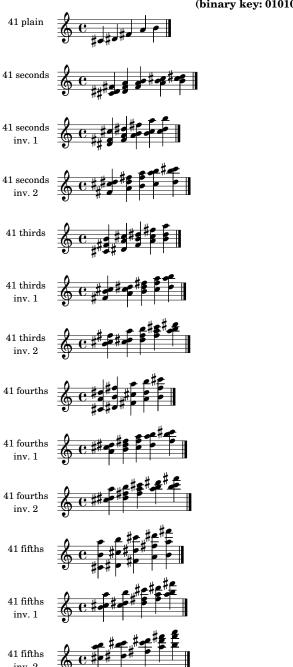


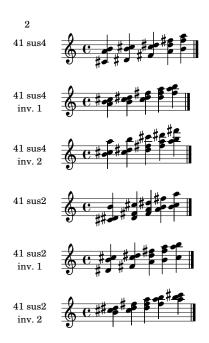
Mode (2)40 (binary key: 0101000)

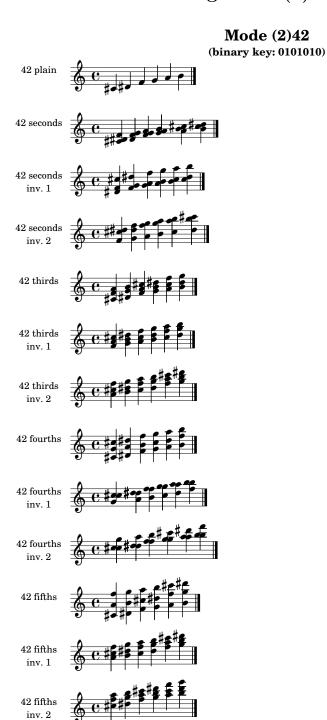


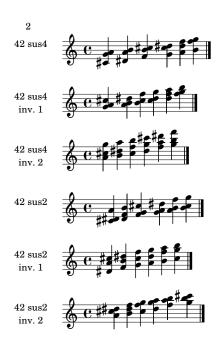


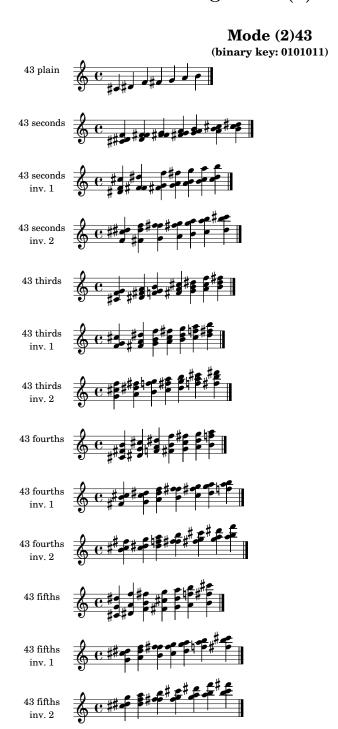


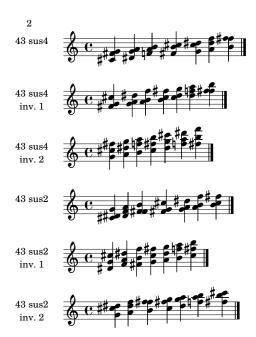


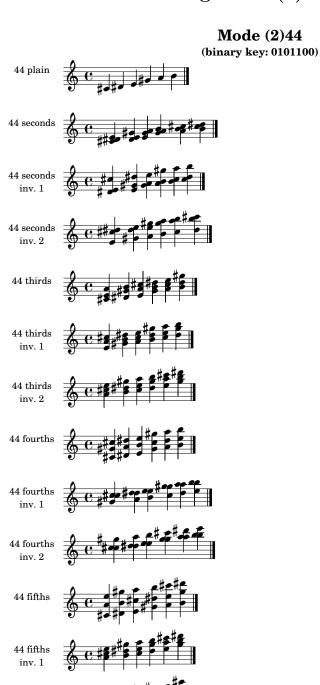




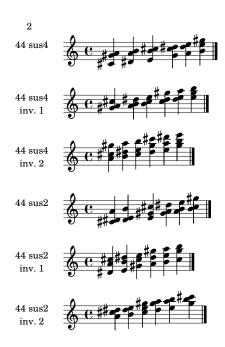


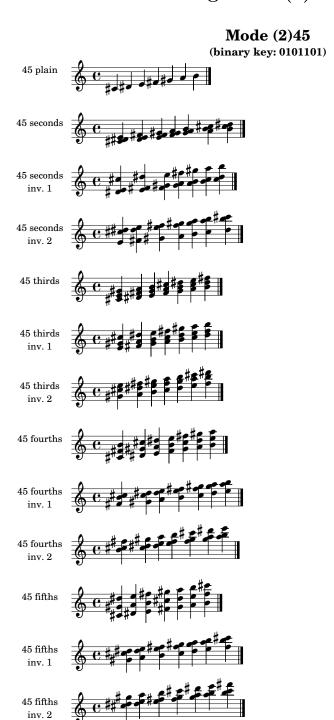


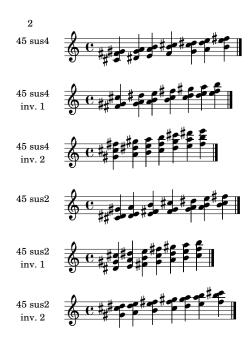


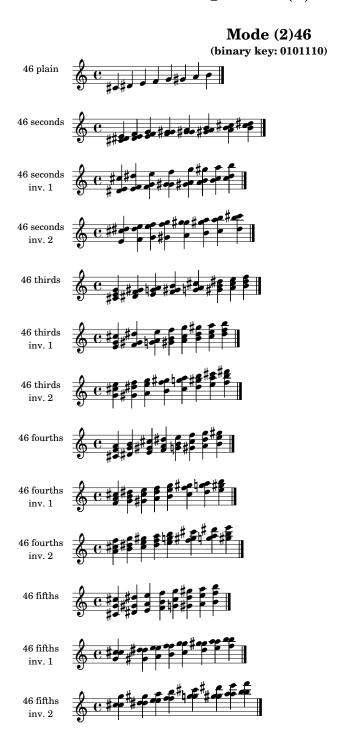


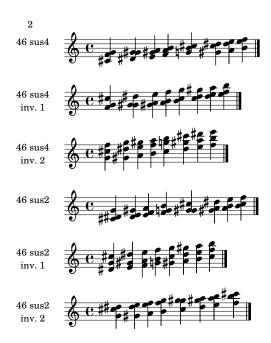
44 fifths

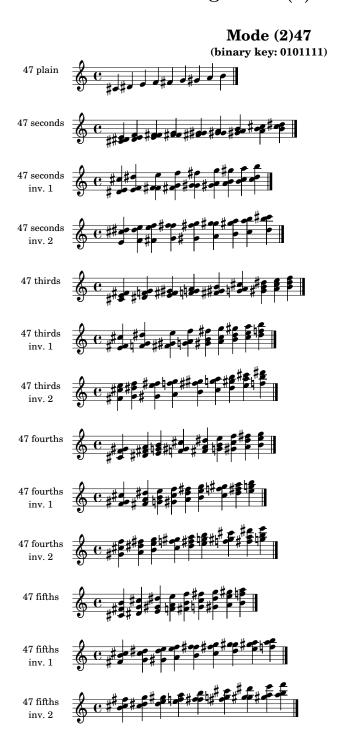






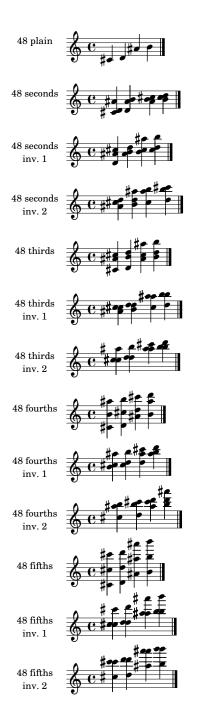


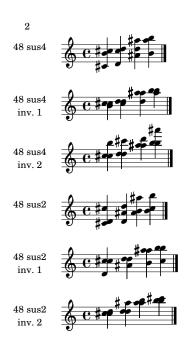


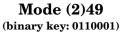


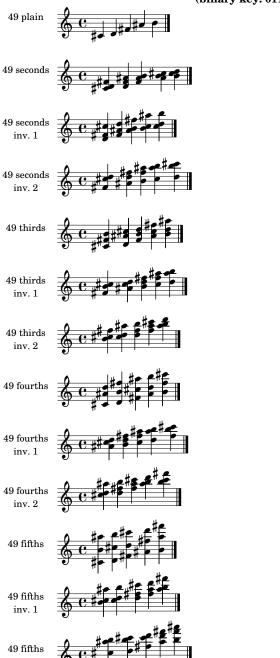


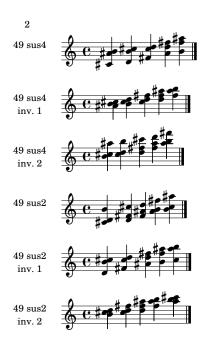
Mode (2)48 (binary key: 0110000)

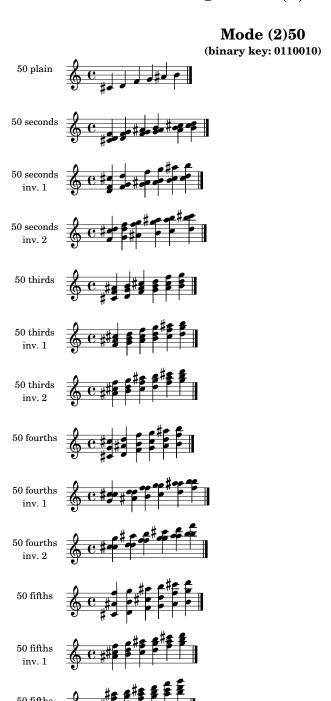


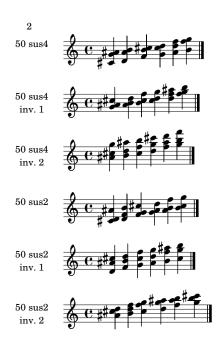


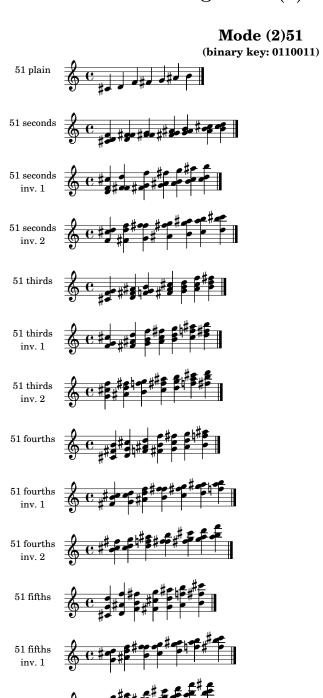


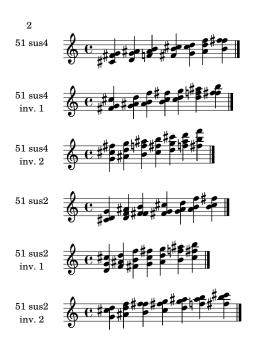




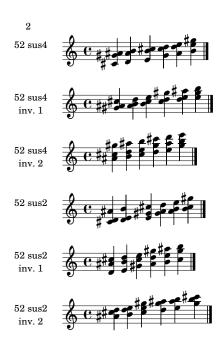


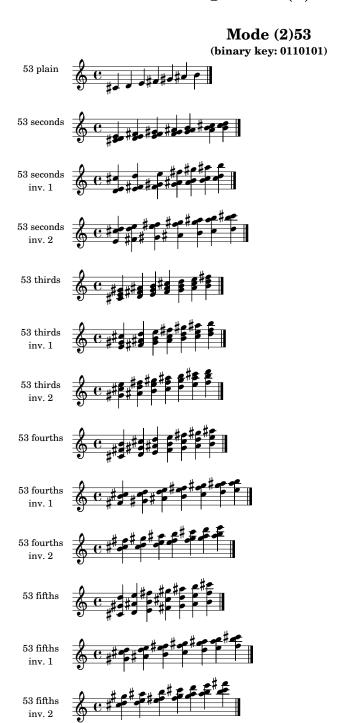


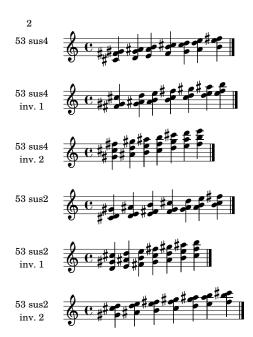


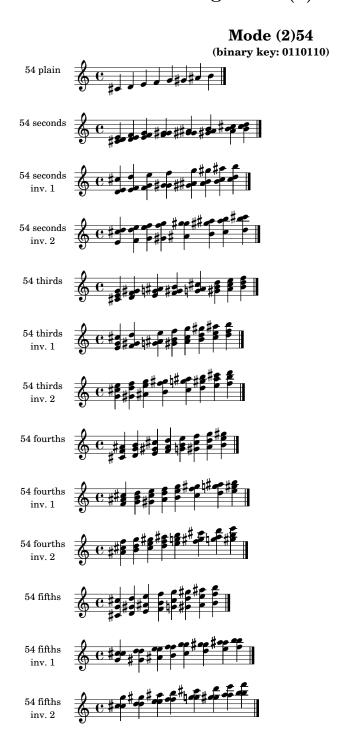


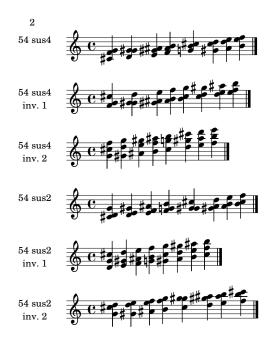


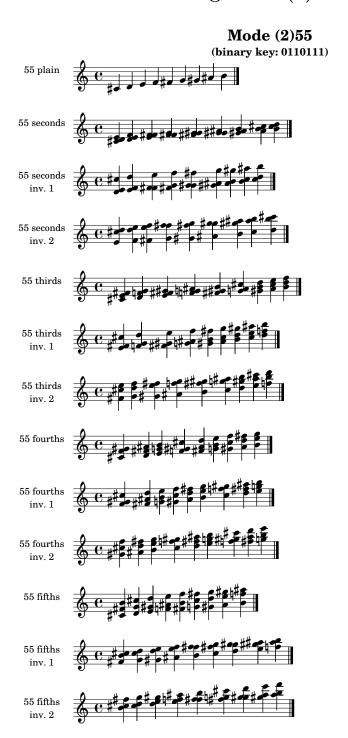


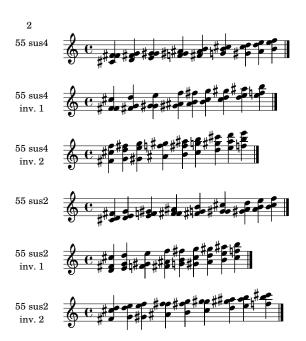




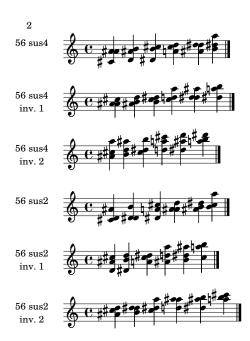


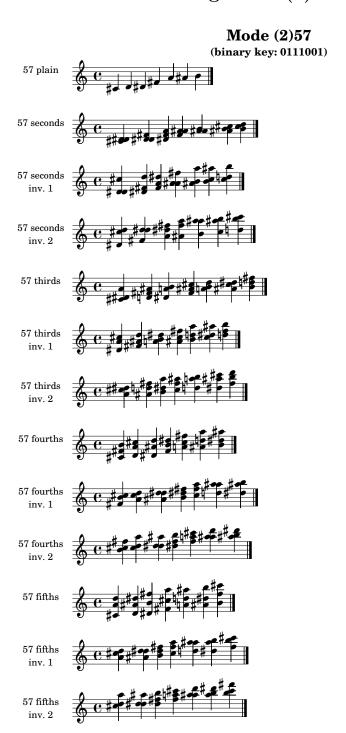


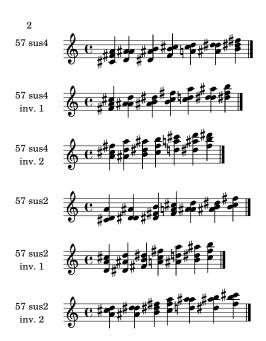


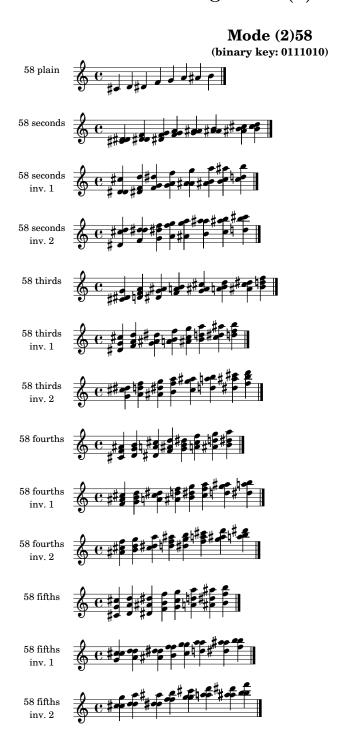


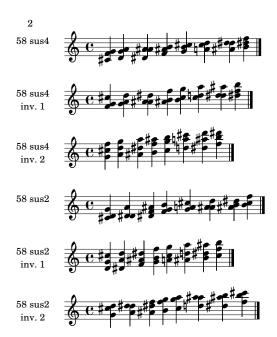


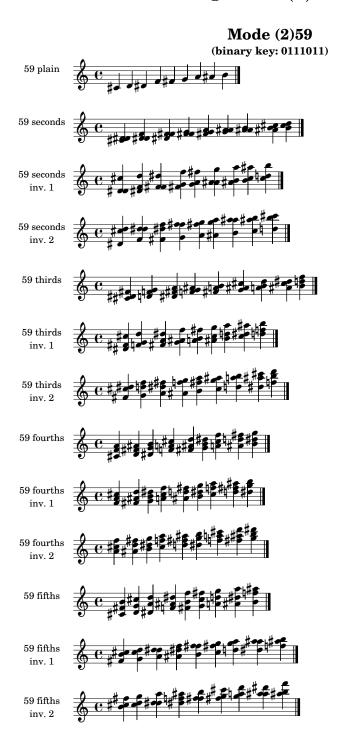


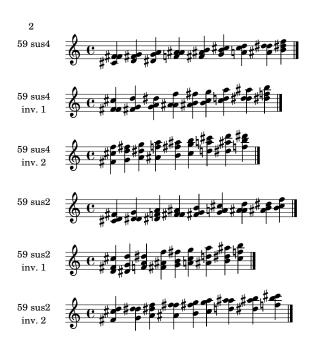


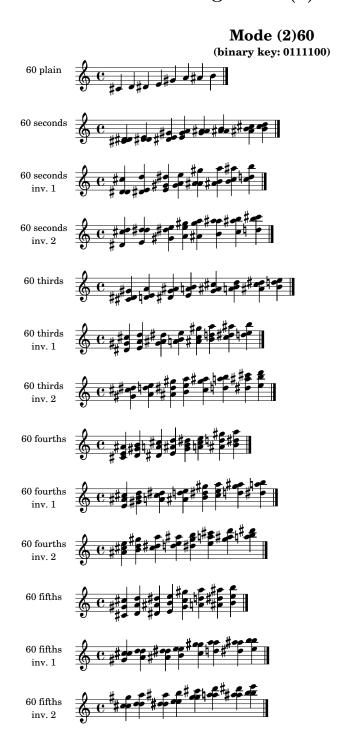


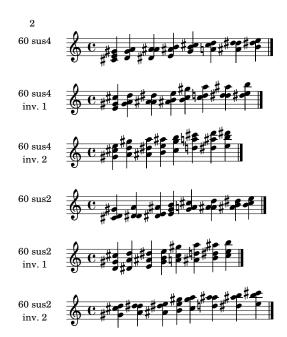


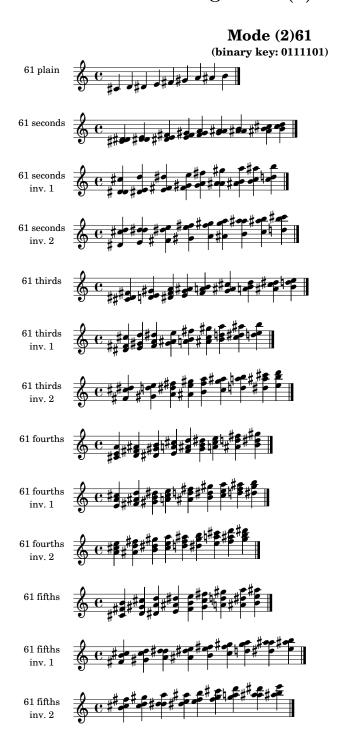


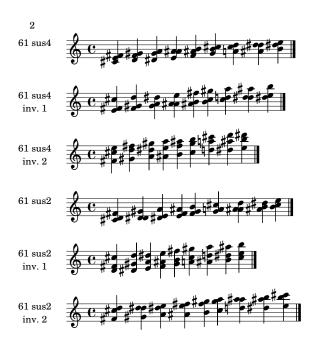


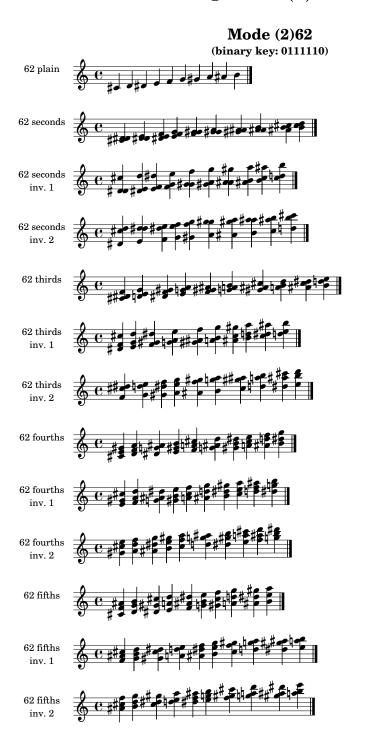




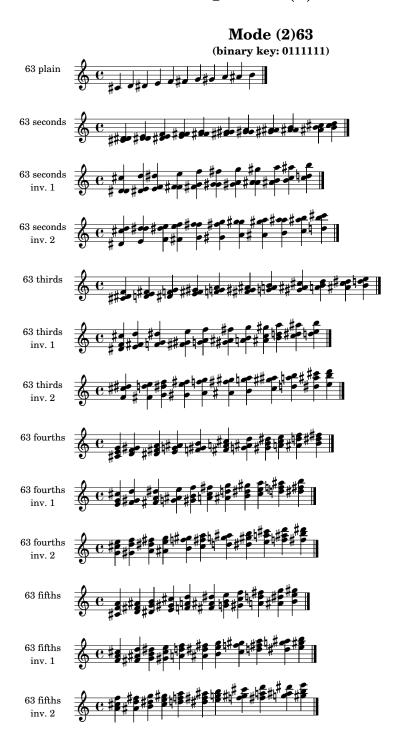






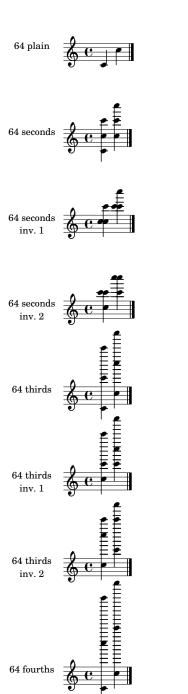


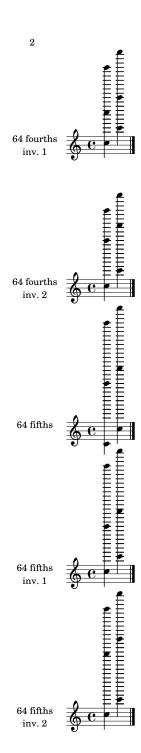


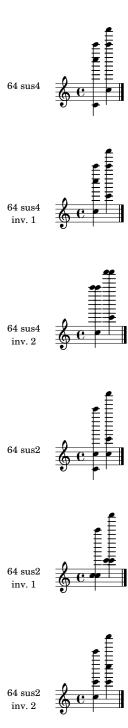




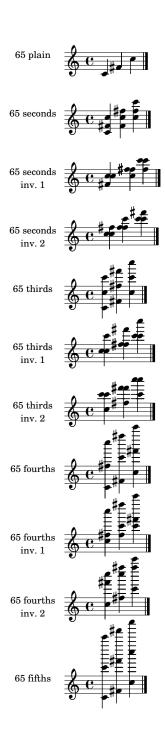
Mode (2)64 (binary key: 1000000)

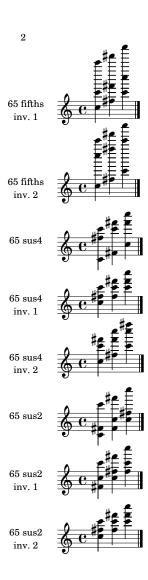




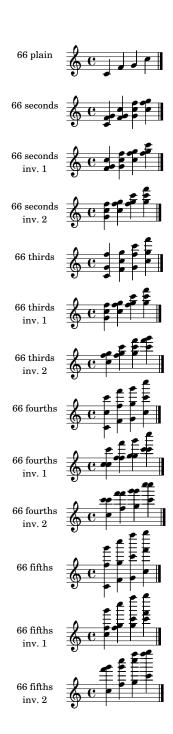


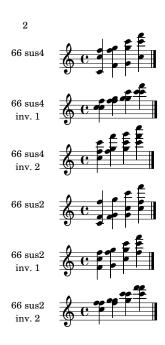
Mode (2)65 (binary key: 1000001)

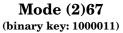


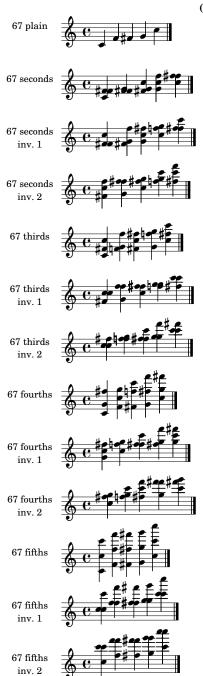


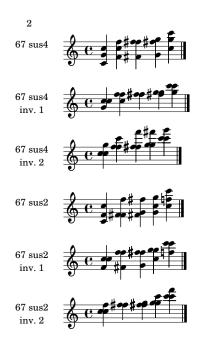
Mode (2)66 (binary key: 1000010)



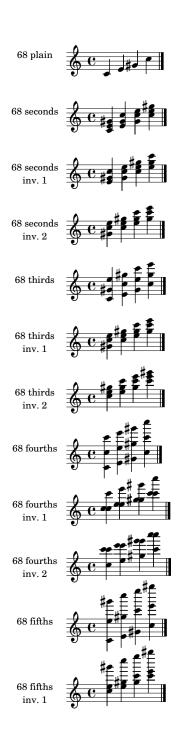


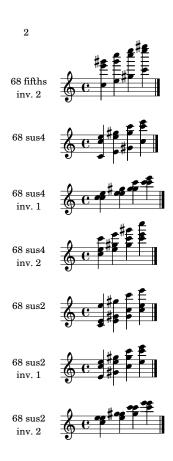




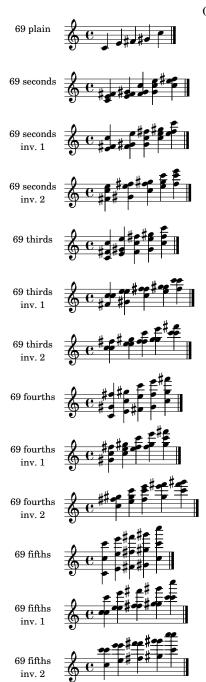


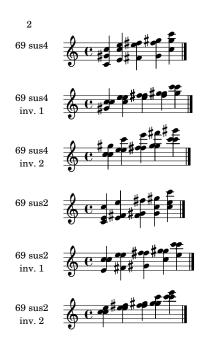
Mode (2)68 (binary key: 1000100)

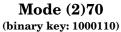


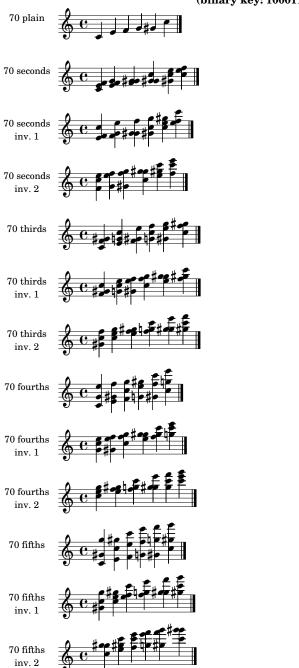


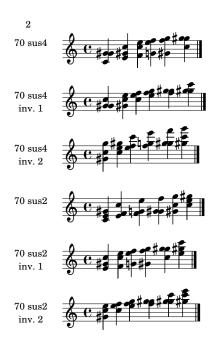
Mode (2)69 (binary key: 1000101)

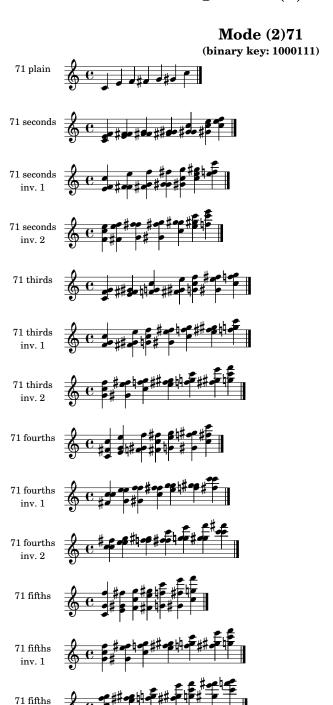


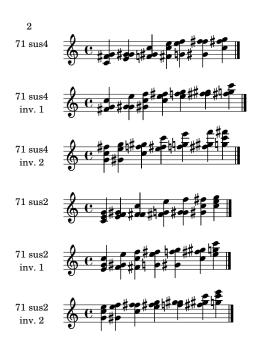




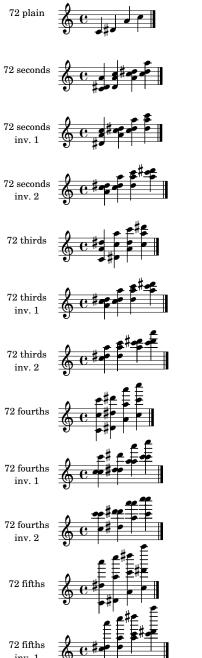


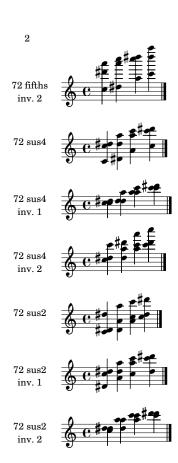


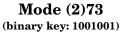


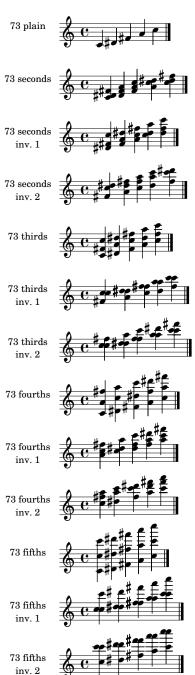


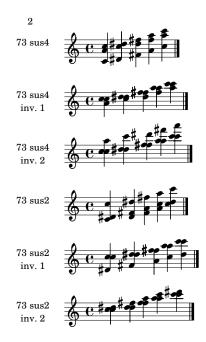
Mode (2)72 (binary key: 1001000)

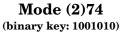


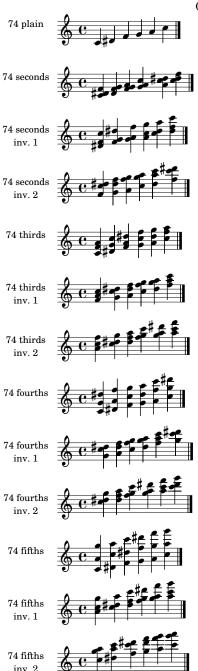


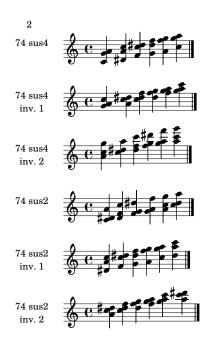




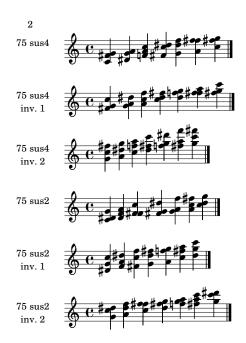


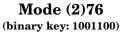


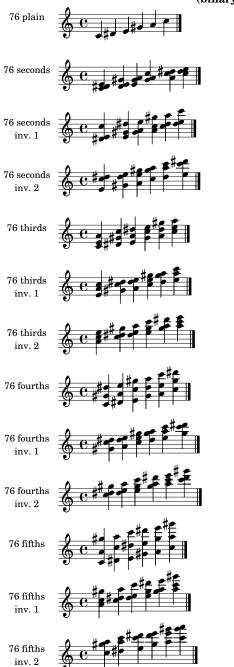


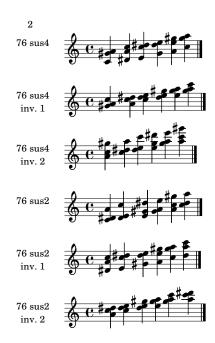


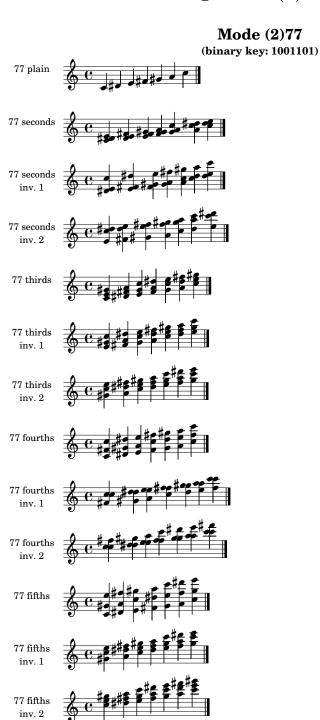


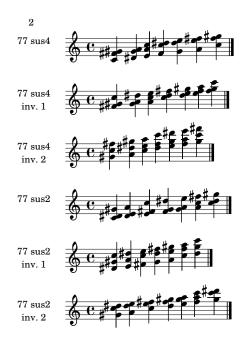




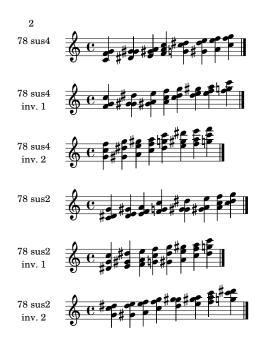




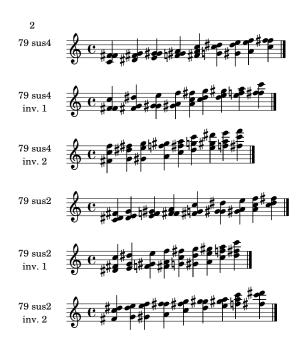




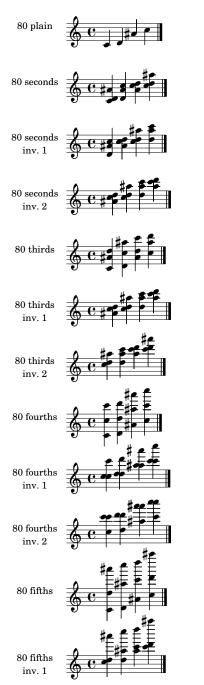


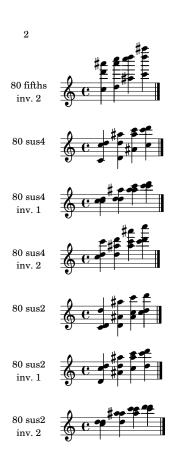


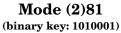


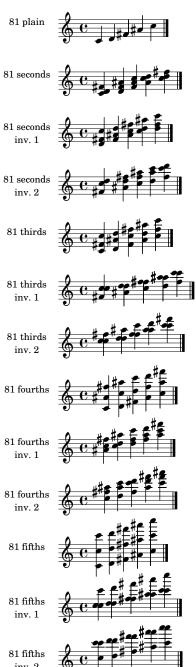


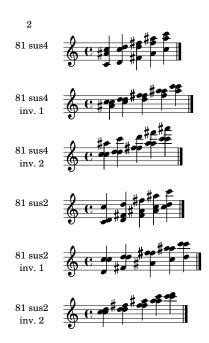
Mode (2)80 (binary key: 1010000)

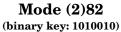


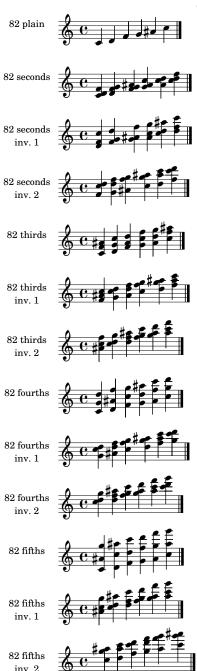


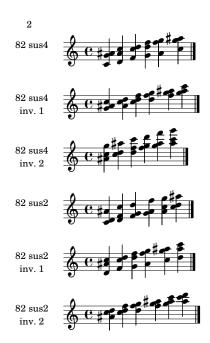


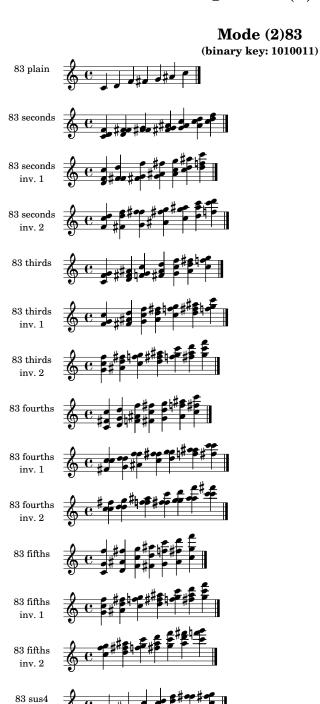


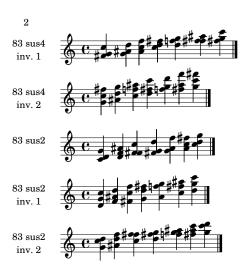


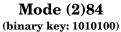


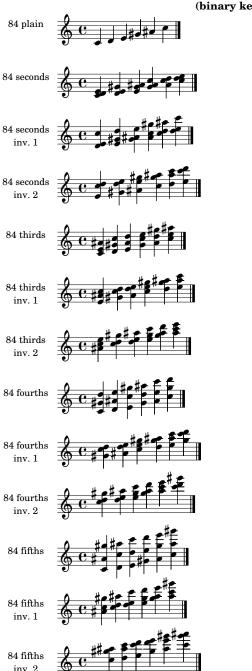


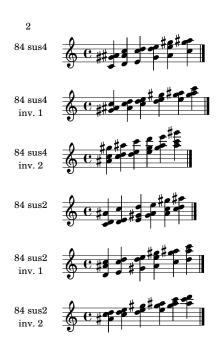


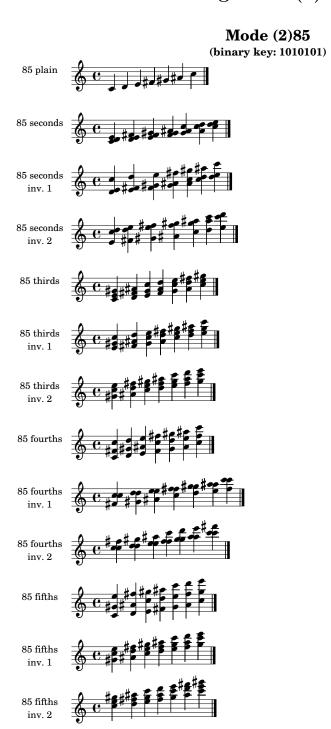


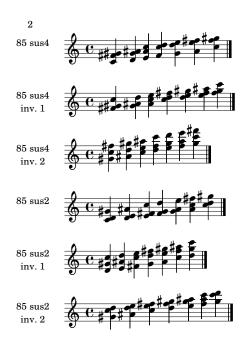


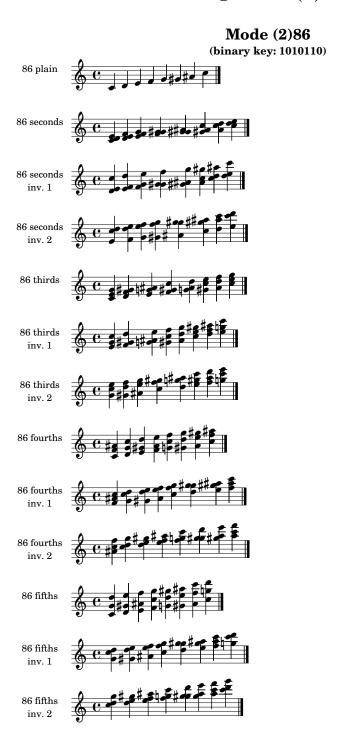


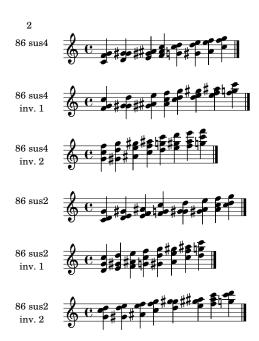




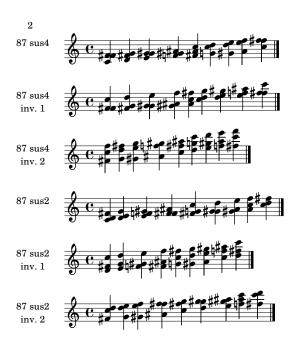


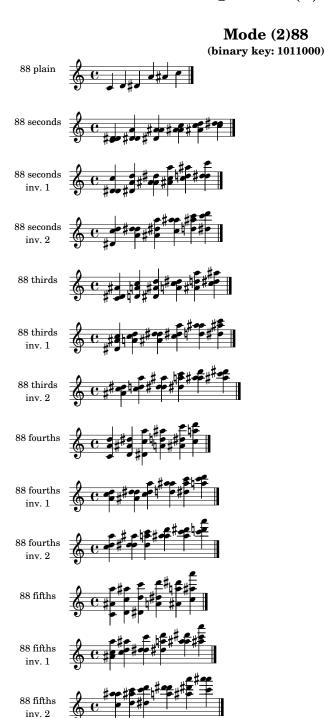


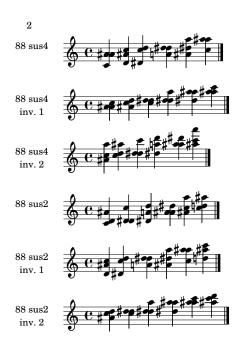


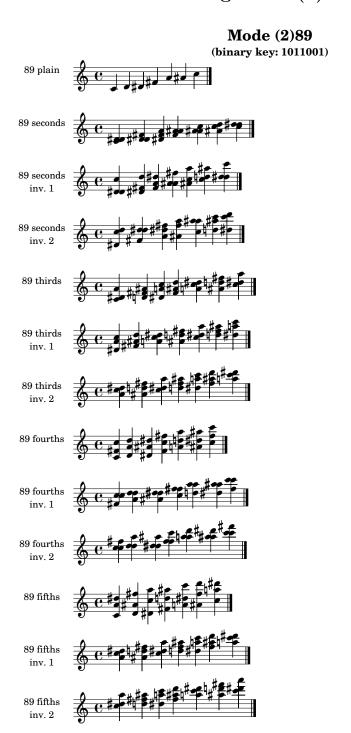


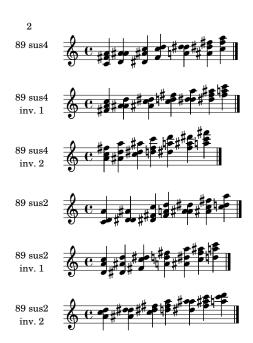




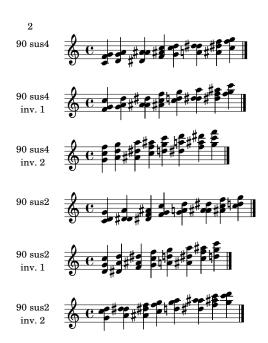


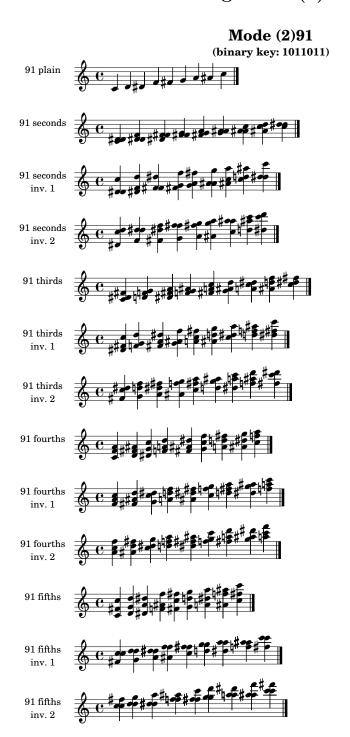


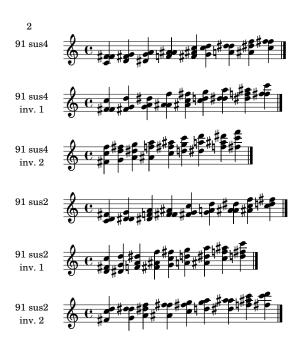


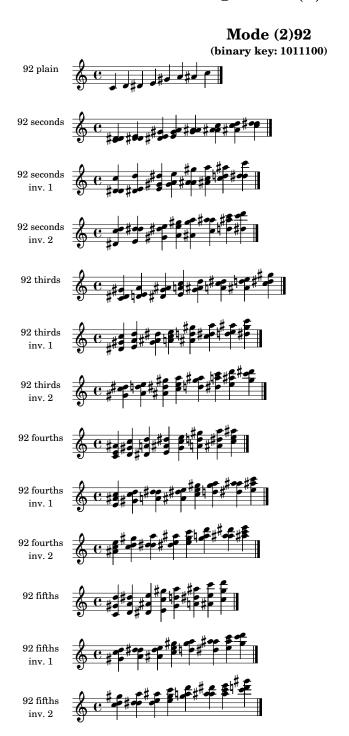


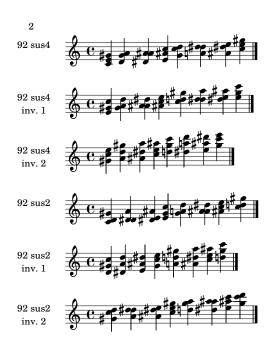


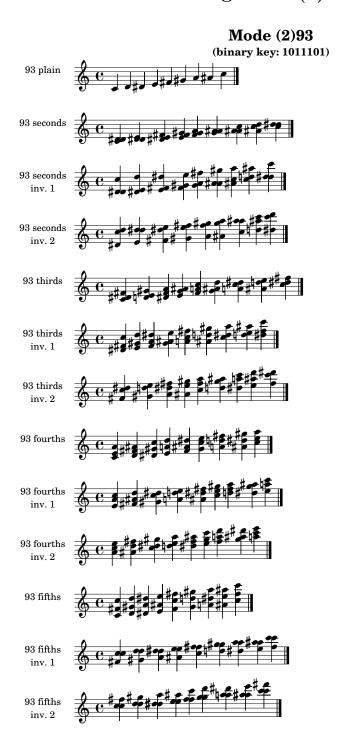


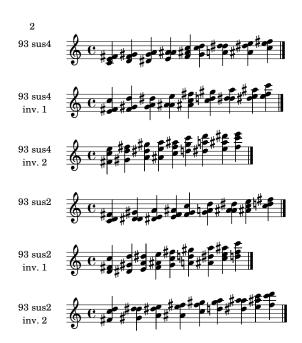


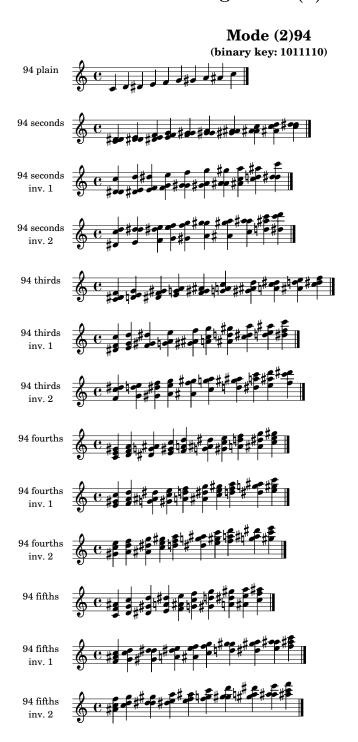


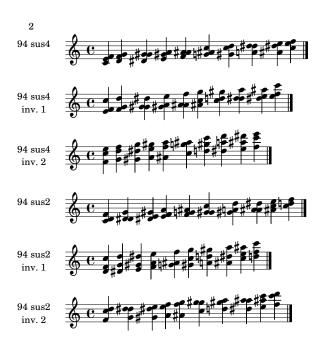


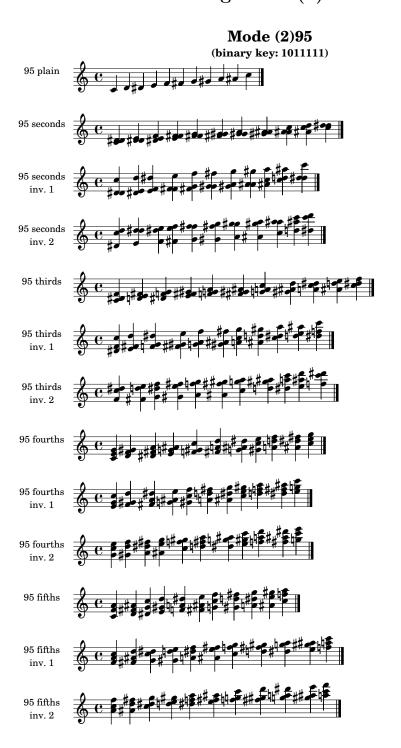






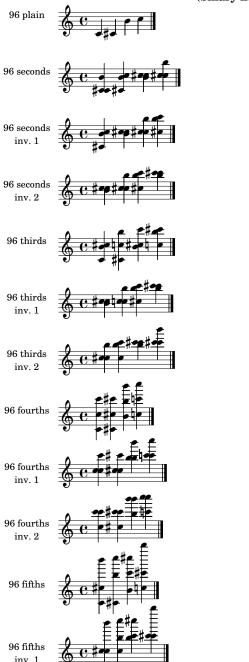


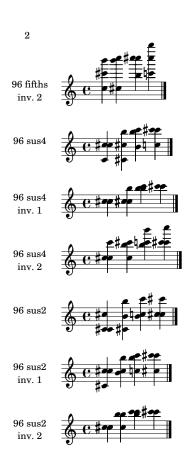


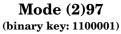


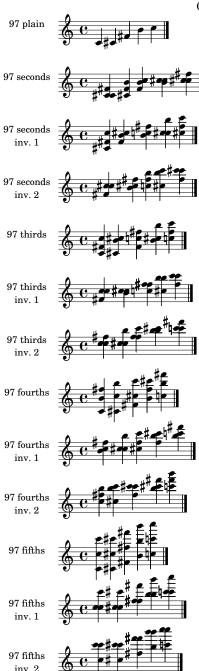


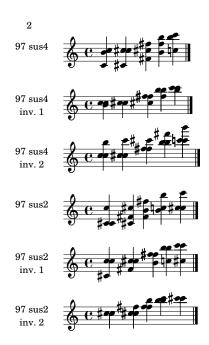
Mode (2)96 (binary key: 1100000)

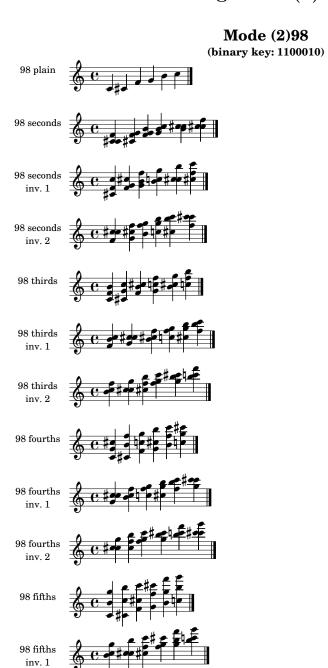




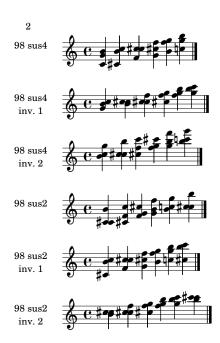




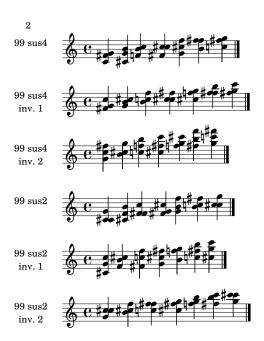


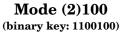


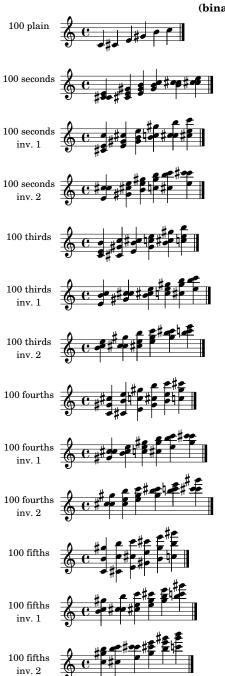
98 fifths

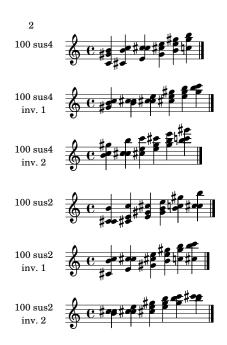


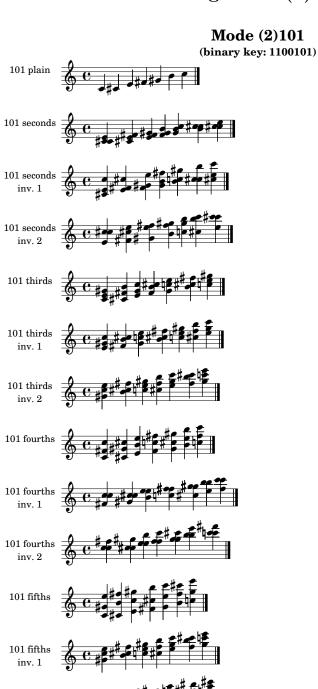


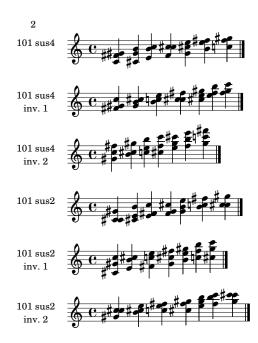


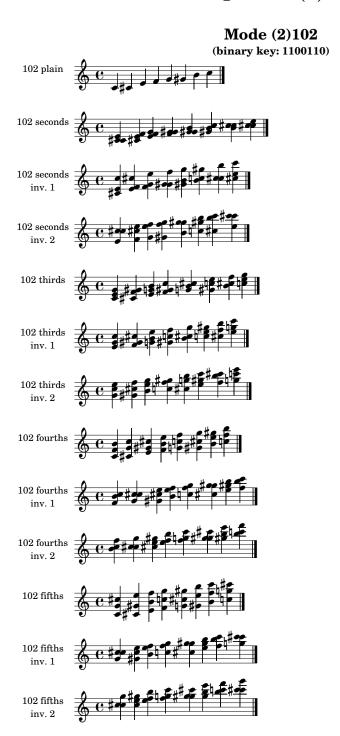


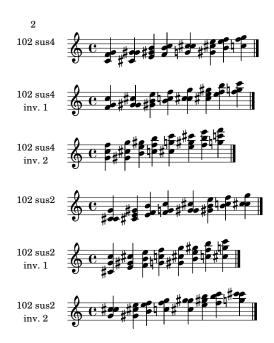


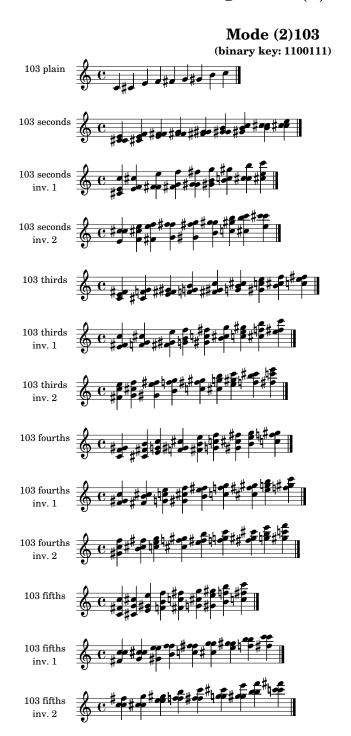




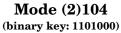


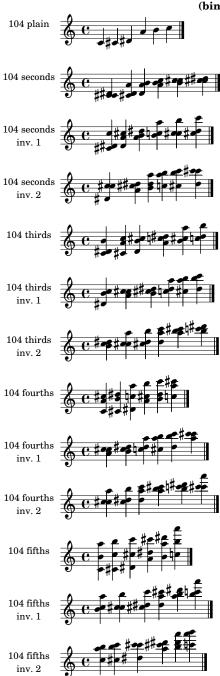


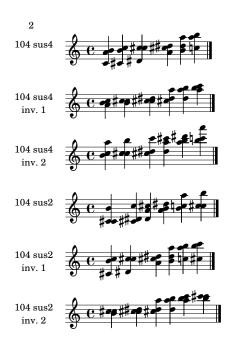


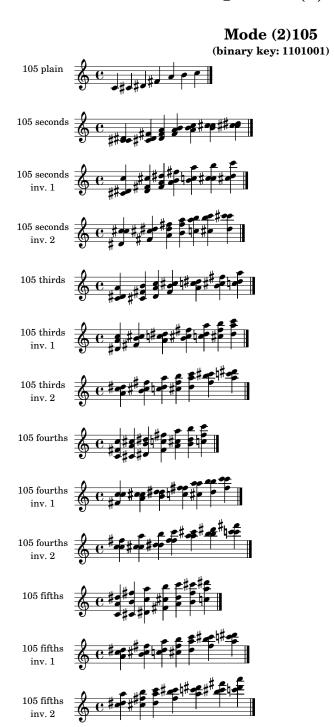


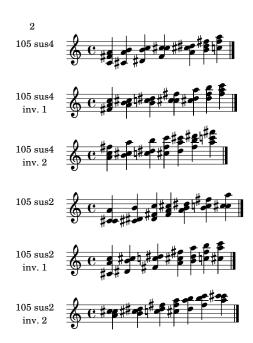


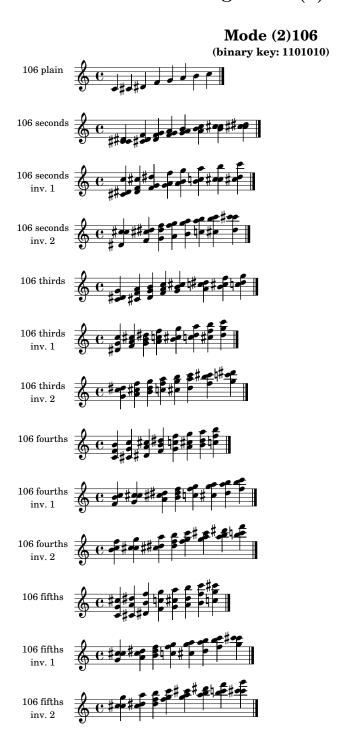


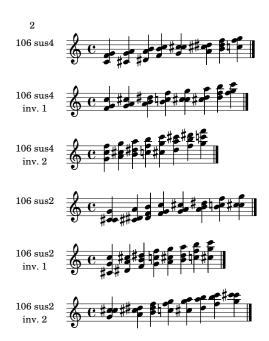


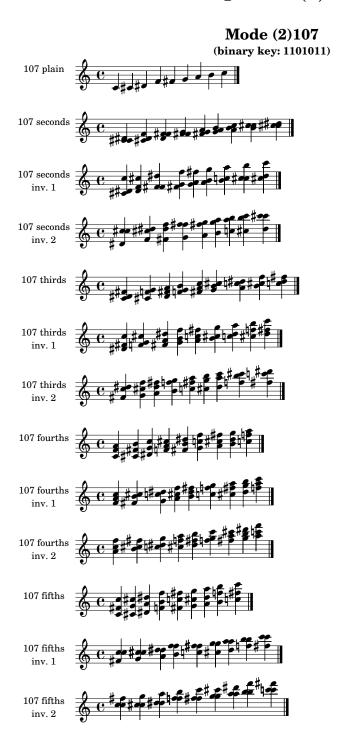


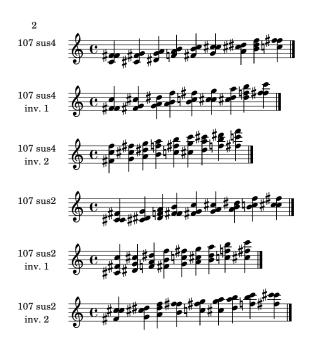


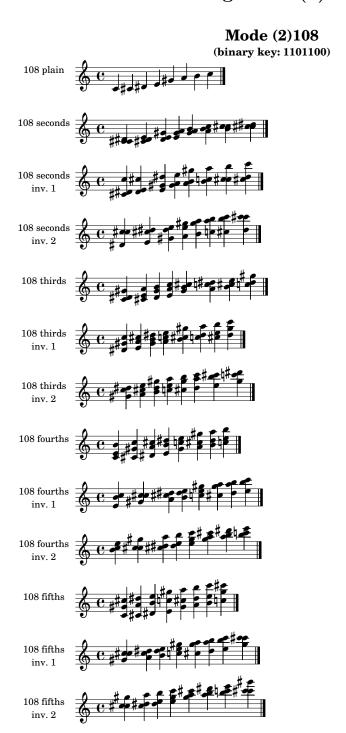


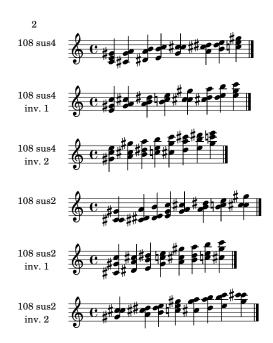


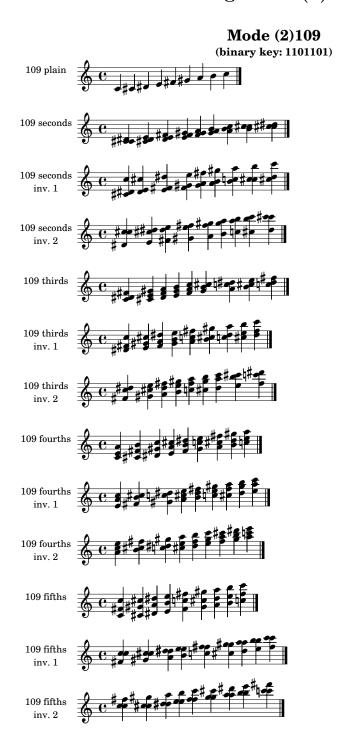


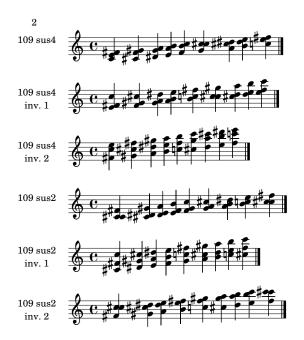


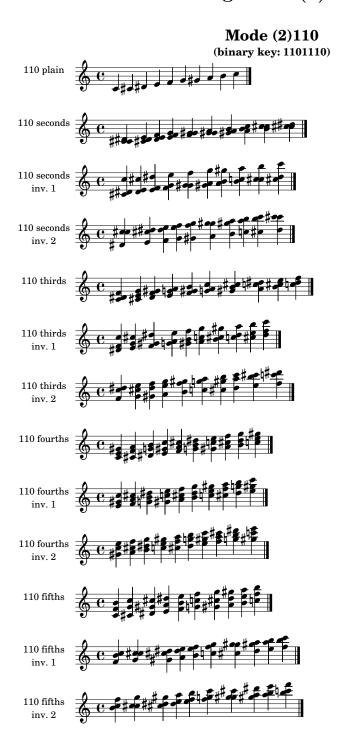




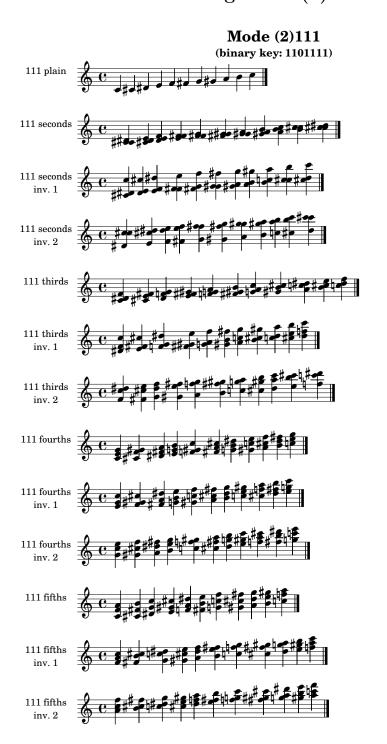


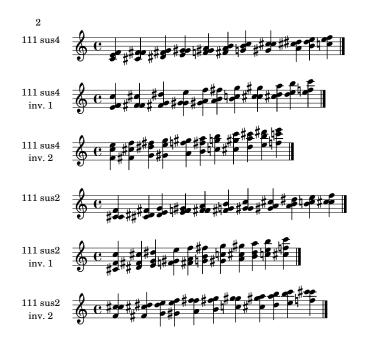


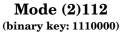


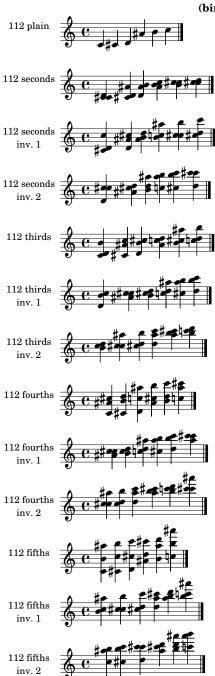


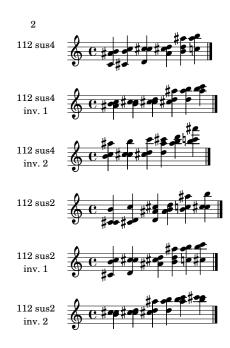


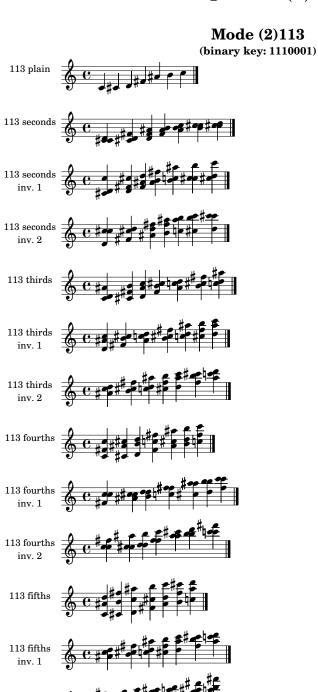


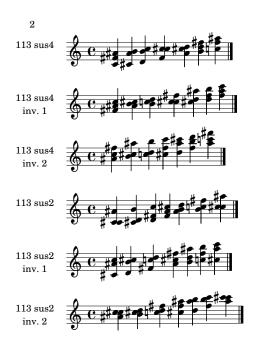


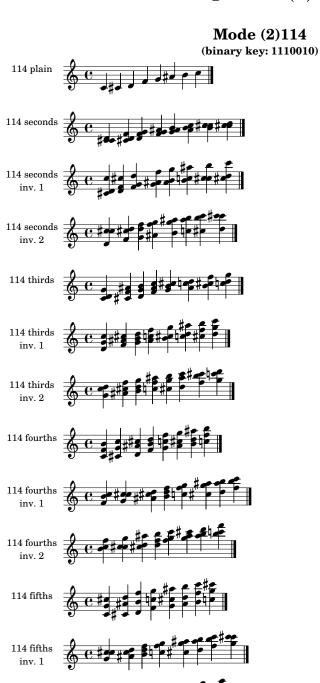


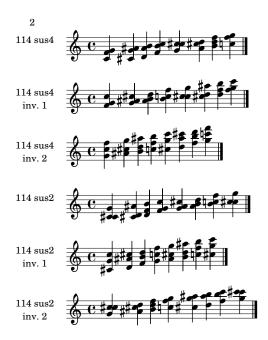


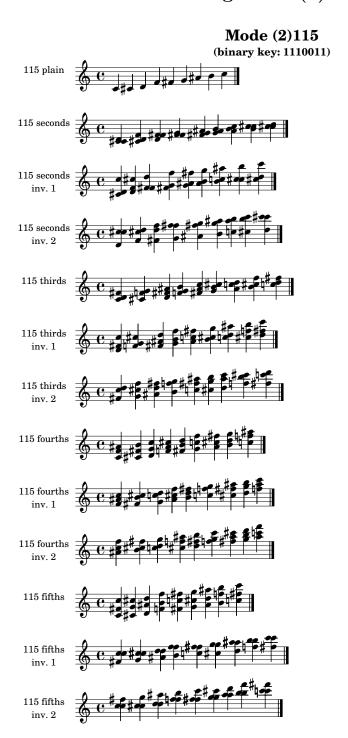


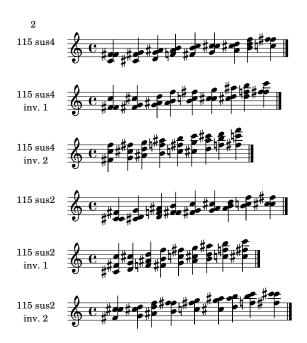


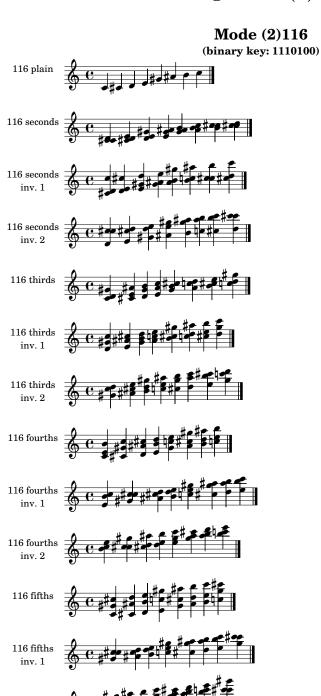


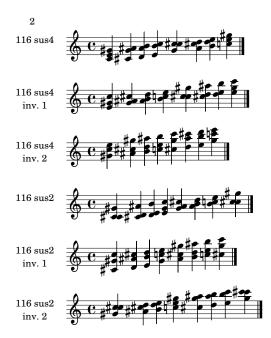


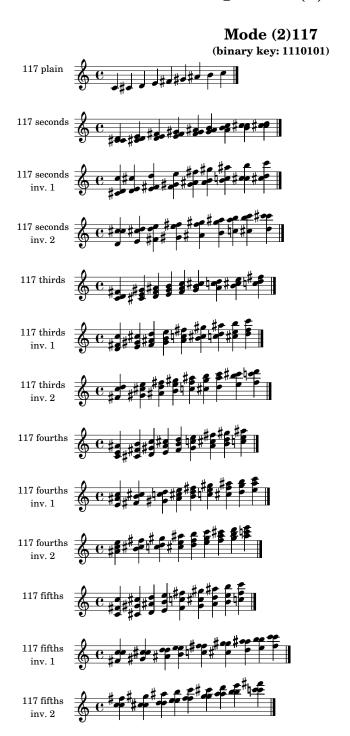


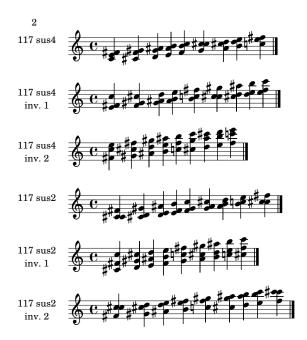


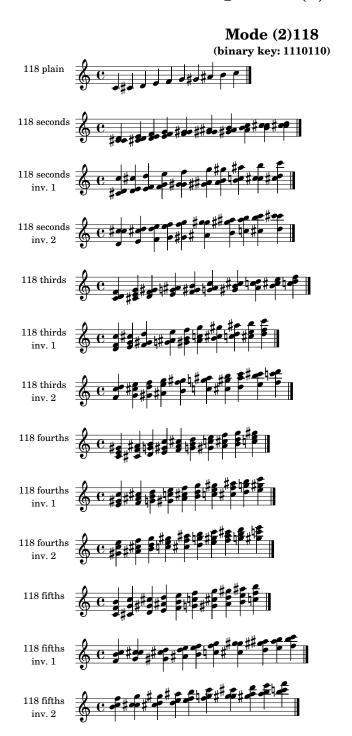




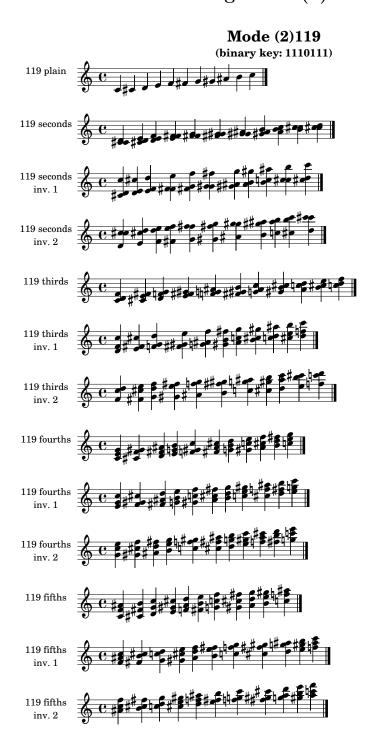


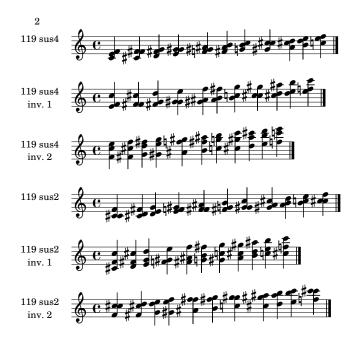


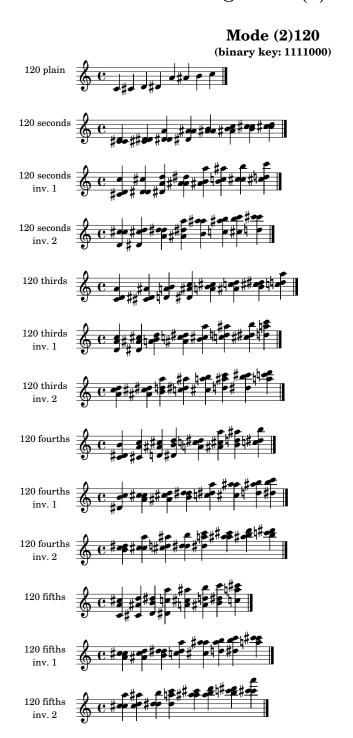


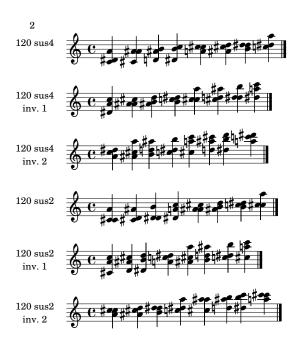


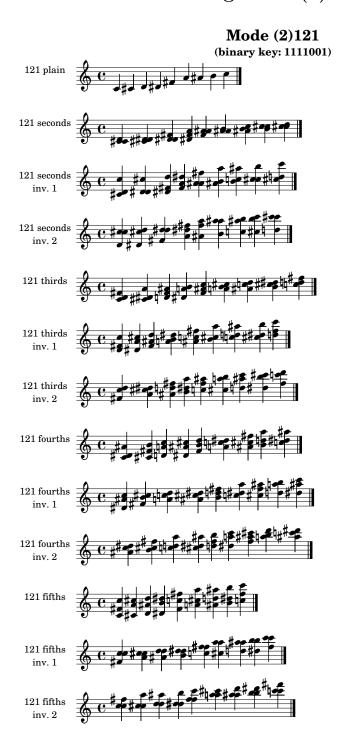




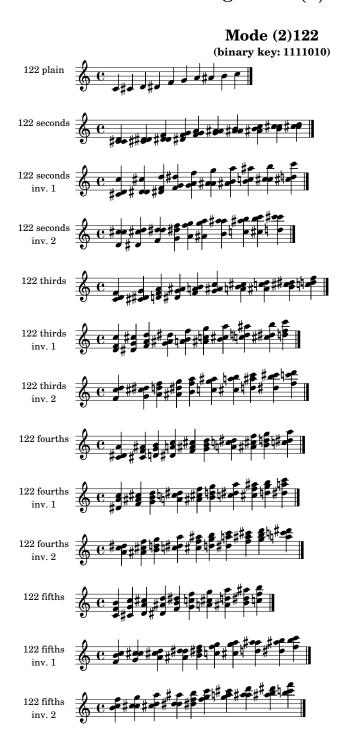




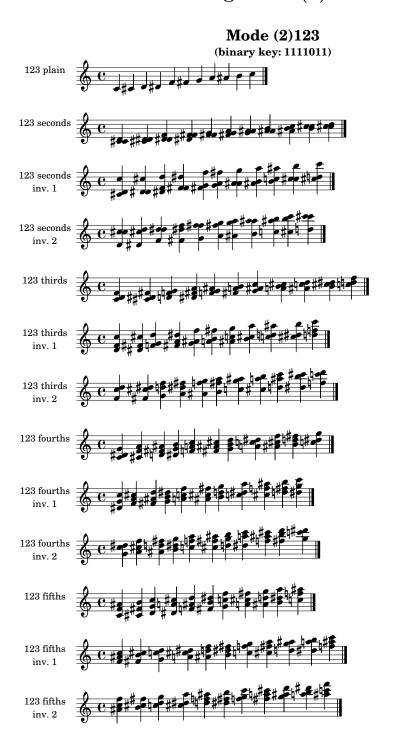




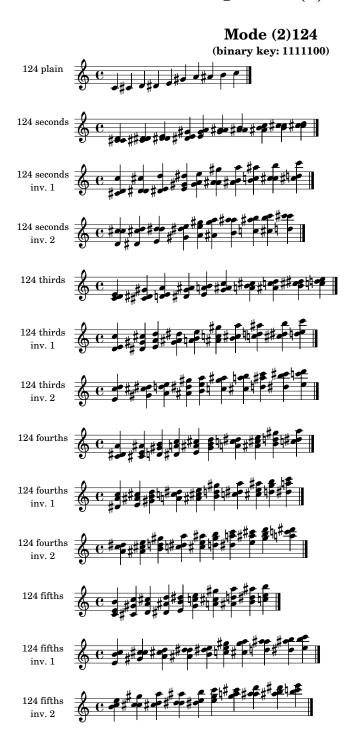




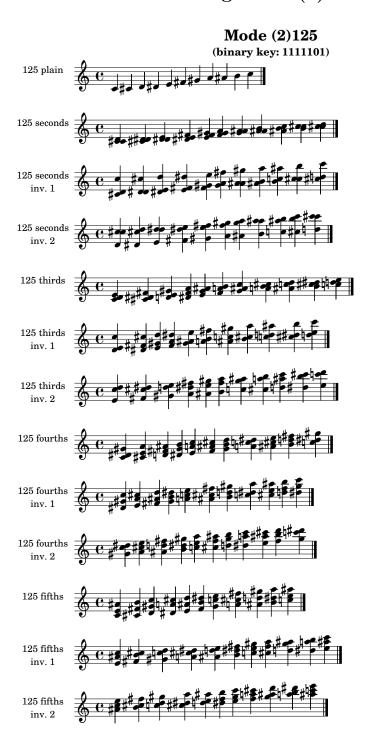


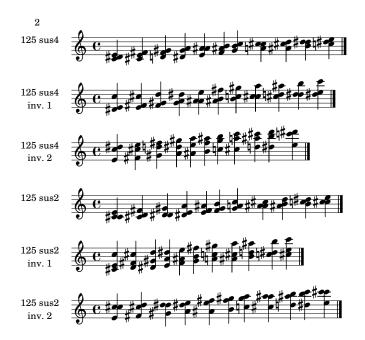


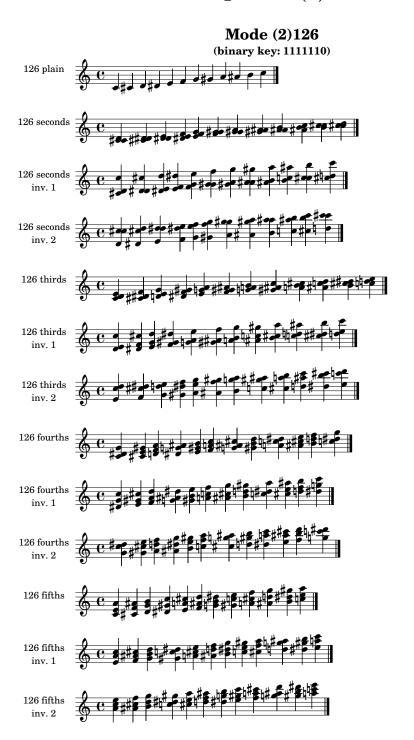




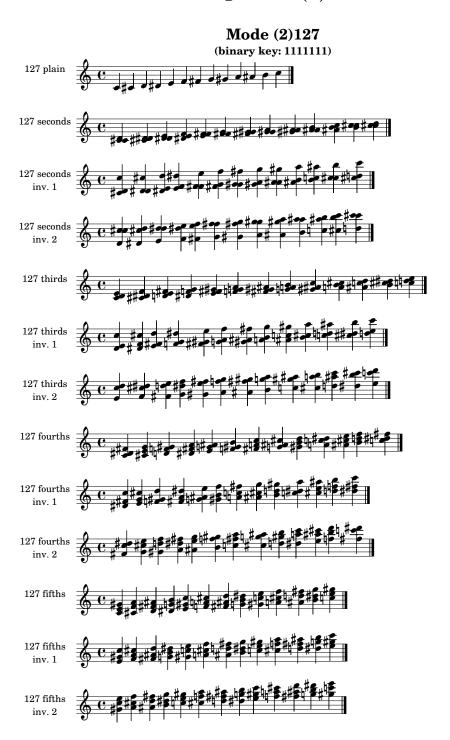








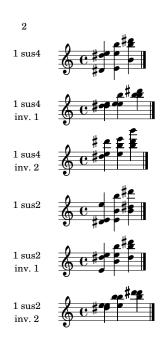


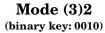


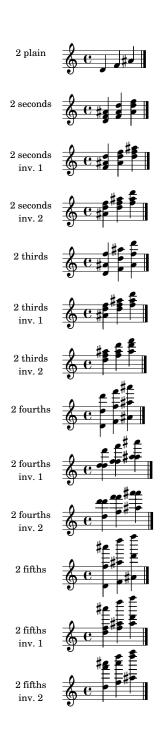


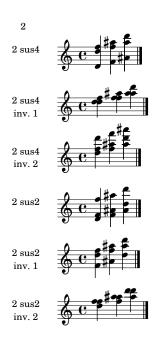
Mode (3)1 (binary key: 0001)

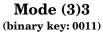


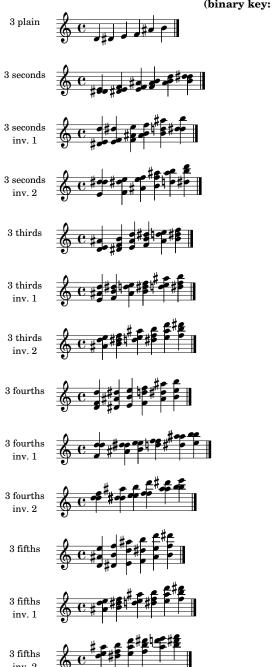


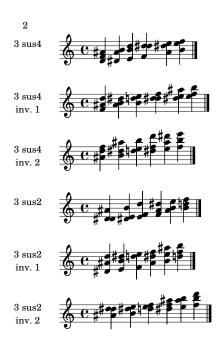




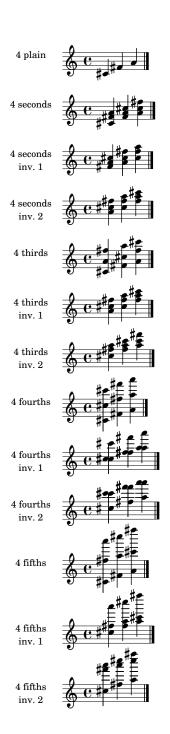


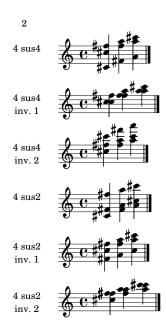


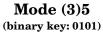


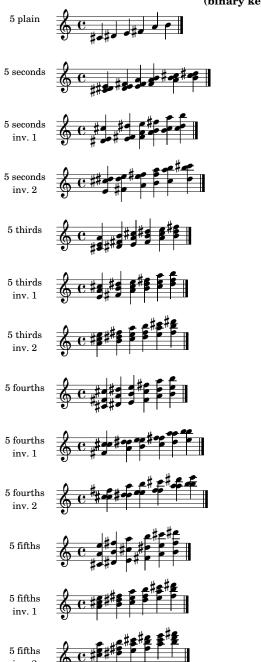


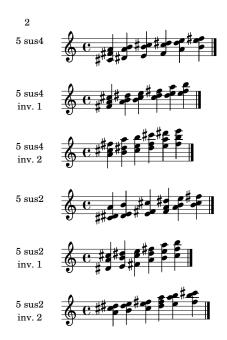
Mode (3)4 (binary key: 0100)

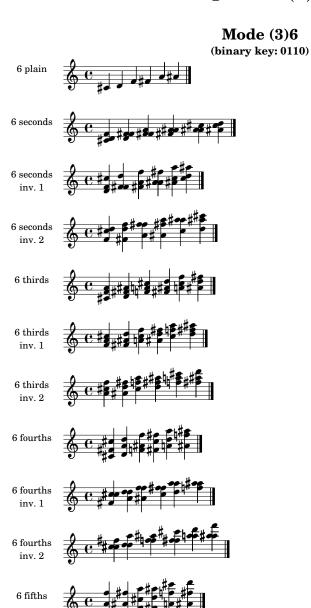






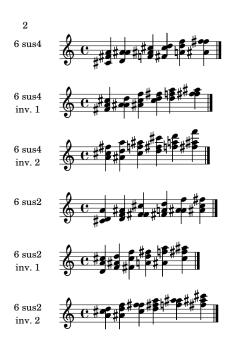


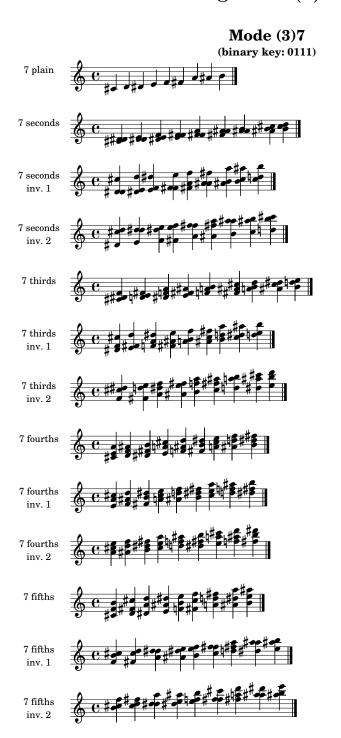


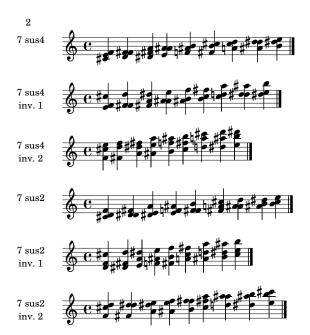


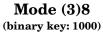
6 fifths

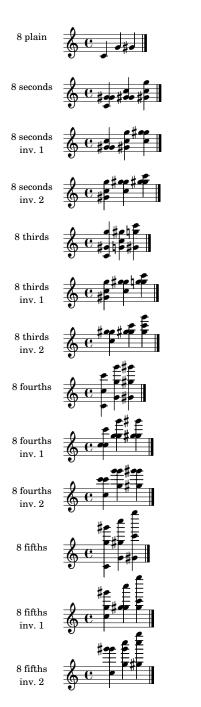
6 fifths

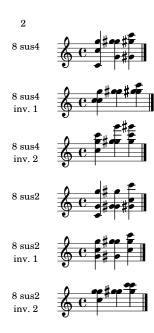


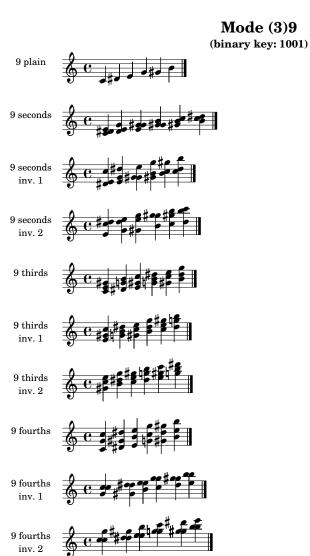












9 fifths

9 fifths

9 fifths

