Everything Is Happening At One Time

for oboe with electronics

(2014-2017)

Electronics

The electronic setup for this piece calls for two (stereo) loudspeakers placed close to and on either side of the performer. Ideally, the oboe part could be performed unamplified, with the volume of the electronics just under that of the oboe at *forte* in a middle register, such that the identity as to whether the source of sound is electronic or oboe-produced is unclear to the audience at times. It is intended that at times the electronics will cover or obscure the oboe sound, especially in the first movement. However, if an unamplified balance cannot be found in a given situation, a clean oboe track is provided in the mixer section of the accompanying software and this can be added to the electronic mix so that both oboe and electronics are broadcast from the loudspeakers. The piece can benefit by being performed in a resonant space.

The electronic component is a piece of software written in Pure Data (Pd) and detailed explanations as to how to configure this are included below in these notes. The electronics for each movement are activated by clicking the movements 1-3 buttons on the top of the patch at the very beginning of each movement and these may also be scrolled through (1-3) using any midi note or control input. For rehearsal purposes it might be easier to just activate each individually and a "reset all" button is provided for rehearsal purposes as well. There is a 3 second delay between clicking the movement 1 button and the start of the movement (or the piece).

Notes

Multiphonics are numbered according to the system outlined in Peter Veale and Claus Steffen Mahnkopf's 1998 book *The Techniques of Oboe Playing*[1]. Multiphonics – with suggested fingerings from the above volumes- have also been inscribed in the score, with the loudest probable pitches noted to represent the complex result of the multiphonic. It is however, intended that other notes and artifacts will sound, even unforeseen possibilities. It is especially intended that harmonics outside the equal-tempered scale and internal vibrations and beat frequencies within the harmonics be emphasized rather than suppressed. The included multiphonics where often chosen because they possess instability or less predictable qualities. If fingering or notated pitches seem in conflict with the sounding

[1] Veale, P., & Mahnkopf, C. (1998). *The techniques of oboe playing: Die spieltechnik der oboe : A compendium with additional remarks on the whole oboe family* (3rd ed.). Kassel: Bärenreiter.

result, refer to Veale and Mahnkopf's book and play the fingering accorded with the attendant number.

All movements are played attaca, one flowing into another.

Mvt 1:

This movement is intended to communicate a sense of movement akin to a running stream in the woods. The stream has a clear direction but is constantly interrupted and forced to change course by various obstacles, yet continues along, maintaining its sense of movements. For this reason, an exact metrical sense is not really important and therefore barlines have been removed. The metronome marking of (*crochet = 100, or as fast as possible*) might even be taken as a challenge to try and outrun the electronic synthesizer, which is by turns following and interrupting your throughout. Likewise, rest durations need not be exact (these are to be thought of as "interruptions" in the stream, and some ad lib flexibility is allowed here) a *fluid* sense of almost improvised yet always flowing movement is sought.

Due to the nature of the electronic sound in movement 1, care must be taken with the volume and manual adjustment may need to be made. The electronics heard from each speaker will be similar, but not identical. Much acoustic interaction and filtering with the acoustic oboe tone will occur when the movement is played together with the electronic part.

Mvt 2:

Like movement 1, movement 2 has no barlines. In this movement the feeling is one of expansiveness but also of a sense of massive blocks of sound, like the massive blocks of brutalist architecture. The image is one of internal movement resisted by itself. Dynamics are to be emphasized in this movement as much as possible.

The note to note pacing is determined by the performer, however in general a hierarchy is scored: short (crochet) notes are to be played as fast as possible and are usually associated with trills, followed by medium length notes (minim) and very long (semibreve) is scored. While the small arrows are associated with the fingerings given for the multiphonics, large arrows along the top of the staves, between multiphonic numbers indicate that if possible the two multiphonics joined by the arrows are to be connected. If a breath is needed amidst some of these streams of multiphonics, absolutely take one. An additional area of interpretive input from the performer comes in the form of trill speeds, which while keeping in the general hierarchical specifications given above are to be varied with a great deal of rubato. In general an emphasis on connectivity (where events are joined by arrows) and aggressive discontinuity (where rests are notated or events are not joined) is sought.

In contrast to the first movement, the electronics for this movement largely support the work of the performer, emphasizing and continuing resonances emerging from the

complexes of multiphonic and trilled textures. It is worthwhile noting that movement 2's electronics do not immediately end following the movement but continue for some time into movement 3.

Mvt 3:

Movement 3 is played at a loud dynamic throughout. The tempo gets slower throughout the movement, and the performer is assisted in performing this linearly by the presence of the visual metronome, whose flash corresponds to each performed note. However, it is best if this metronome is thought of as a calibration tool as it will be impossible to follow it strictly for each event (as it is constantly losing energy in a linear fashion). It is better to think of always making a gradual ritardando over the larger-scale structure of the piece and using the metronome to recalibrate along the way.

Various small bends are called for, and these are notated by small dashes up and down representing scoops and dips in the pitch, like small, quick glissandi. The distance of travel in terms of steps (such as half-steps, full steps, quarter-tones) of pitch is given in parenthesis above each bent note. I have composed this thinking that most of these might be performed with lip movements alone, but where finger glissandi are necessary that is also fine. The actual distance of the glissando may be adjusted as well, the point being that the entire event is thought of as a sound sliding, rather than a performed pitch that then slides.

The electronics for this movement are both interruptive and supportive, probably seeming more interruptive at the beginning at the movement where a direct transition from the second movement is sought. A structural image that guides this movement again one of expansiveness, but this time held back by its own internal structure. It moves, but in discrete steps, while all the time bends and angular, unpredictable repetitions in the electronics bely assert a tension with this gridlike limitation on movement.

Installing and Running Pure Data (Pd)

Pd is an open-source visual programming environment. One programs by connecting bits of code in small boxes in a similar manner to connecting modules within a modular synthesizer, except that here you can connect and assemble not only audio but logical processes. This approach might be familiar from MAX/Msp, which was largely invented by the same person Miller Puckette.

To run the software for this piece, download Pure Data:

https://puredata.info/downloads/pure-data

On that site you will see some options. You want the version labelled "Vanilla". Once selecting this there will be several other options listed. From these, it is simpler and better to download the version for your computer (Mac/Windows/Linux) rather than the "source tarball" that is labelled "Get Pure Data for all platforms". The patch for this piece should work on any operating system. Once downloaded simply decompress (unzip) the file and install.

Once you have Pure Data installed, it is necessary to install some libraries on which this patch depends. **We will be installing the libraries named extra, creb, cyclone, and zexy**.

This is easily done by following the steps below:

1. Start Pure Data by clicking on the link that appears on your desktop of in your list of programs after installing the program.

2. From the console window which appears after you start the program look along the menu tabs along the top of the window until you reach one labelled "Help" on the far right.

3. Click this and scroll down to "Find Externals", click this.

4. Clicking this should open a new window. This is a plugin called Deken that is installed by default. From this window we can easily install libraries from the Pure Data code Repository. To do this:

    a. type the name of the library you want into the field at the top of the window

    b. click "show all"

c. select the version of the library that is most recent version for your computer system (Windows/Mac/Linux). Click that and let it install automatically (clicking permissions where prompted by your own computer. That should be it.

5. The libraries you need to install for this step are called extra, creb, cyclone, and zexy.

Configure Audio and Midi:

1. Click the DSP box on the right side of the main screen is checked

2. Set up sound settings:

Go to Media menu and then down to Audio Settings

Set your audio in and out for your sound card or computer

This patch reads audio from inputs 1 and 2 (the stereo audio ins) and can read audio in off only channel one just as well, so whether the input is mono or stereo in shoudn't matter.

The **SoundTester patch** built in into the **EverythingPerformance** patch will give you a numeric volume readout of any audio input detected and also provides you with a pink noise tone to test the speakers with. There is also a built-in test patch in Pd, which you can find by clicking on "Test Audio and Midi" in the Media menu.

3. Set up midi:

Within the **EverythingPerformance** patch, you can move from one movement to another by providing any midi input. Any midi note or control value will advance you to the next movement from where you currently are, scrolling through each in order, 1-2-3. Any midi input at all will do this and should not to anything else, so attaching a simple midi pedal (or stepping on a midi keyboard) can provide you with a simple hands-free way to move from movement to movement without pause in performance. You can also move from movement to movement by simply clicking the buttons for each movement in the **EverythingPerformance** patch, but in performance this is slower than using, for example, a midi foot controller.

To set up Midi in Pure Data, click **Midi Settings** and chose your device as input from the drop down menu. We need not worry about output. If your device does not appear it is likely an issue between your computer and device, as Pure Data generally automatically recognizes devices that the computer itself recognizes as connected. You can test whether the device is working by looking for a response from the button on the SoundTester module within the EverythingPerformance patch, or alternately within the build in test patch discussed above.

Now, to play the piece:

Click the patch named **EverythingPerformance.pd** to open, or if Pure Data is already running, open the piece from the File menu on the top left.

EverythingPerformance *must remain in the* ***EverythingIsHappening*** *folder with the other patches included there as these are dependencies which the performance patch needs to to work. If you move the patch, just move the whole folder. If you re-save it elsewhere it will not work properly without all the materials from the original folder.*

The performer starts the piece by clicking the Movement 1 button or sending any midi note or control signal. *If midi information has been previously input it is necessary to restart the patch to guarantee that all movements start from their default beginning states.* Please note there is a three second dealy between clicking the Movement 1 button and the start of the electronics.

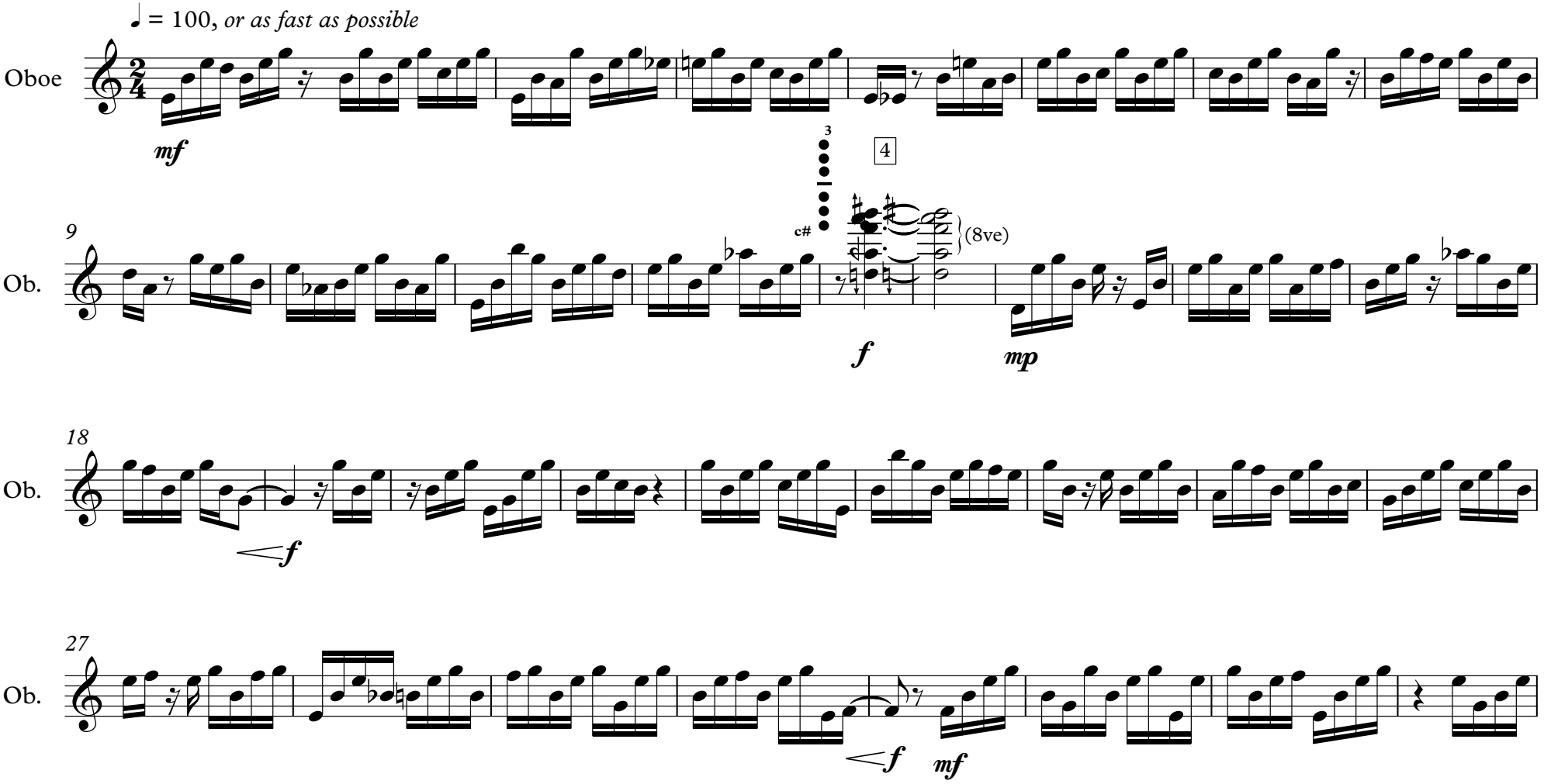
Following Movement 1, Movement 2 is activated in a similar manner, and Movement 3 after this. Movement 3 ends on its own.
No other buttons need be used and the rest of modules you see on the screen are simply there to provide performer feedback.

The sensitivity of the pitch detecting program in movement 1 can be adjusted if needed using the number box in "Rescale Sensitivity here".

To reset and start again just closing the patch and restart it, that way you can be sure that the memory buffer is clean, all timers reset, and you can start from scratch.

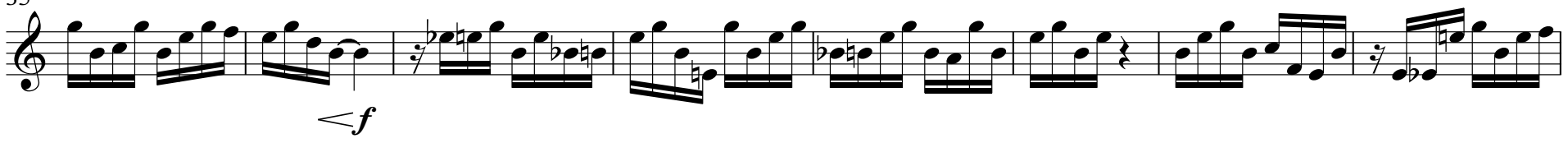
Volume for clean oboe amplification (if needed) and electronics are included at the top of the patch.

I.


♩ = 100, or as fast as possible
Oboe
mf
3
4
c#
(8ve)
9
Ob.
f
mp
18
Ob.
f
27
Ob.
f
mf

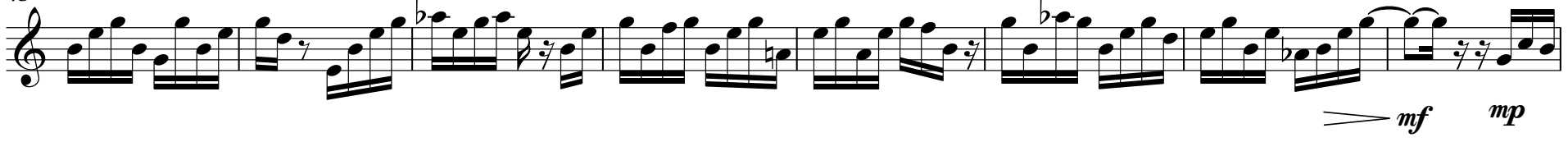
35

Ob.



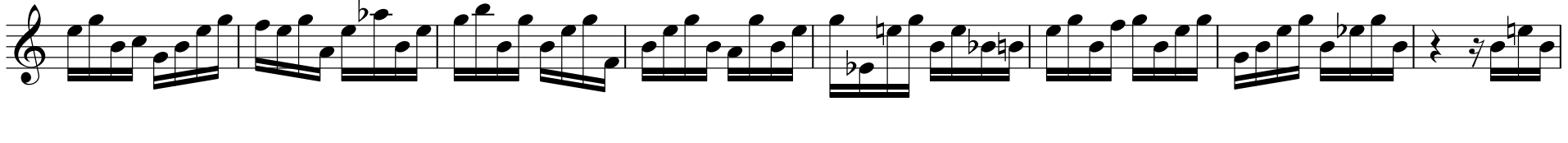
43

Ob.



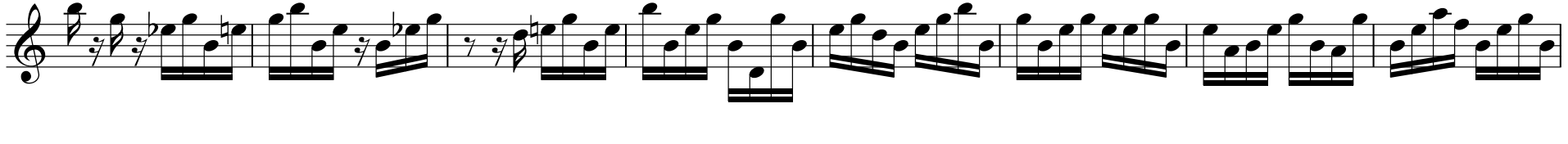
51

Ob.



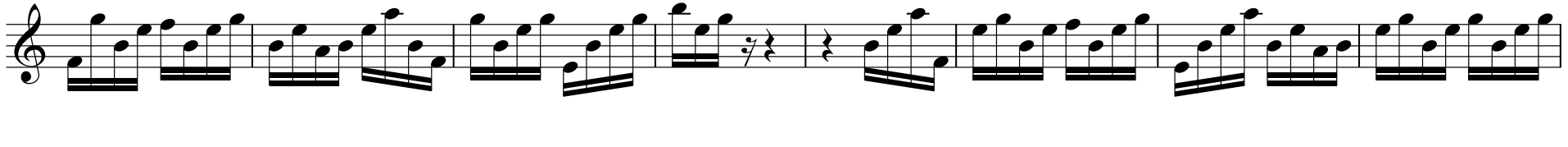
59

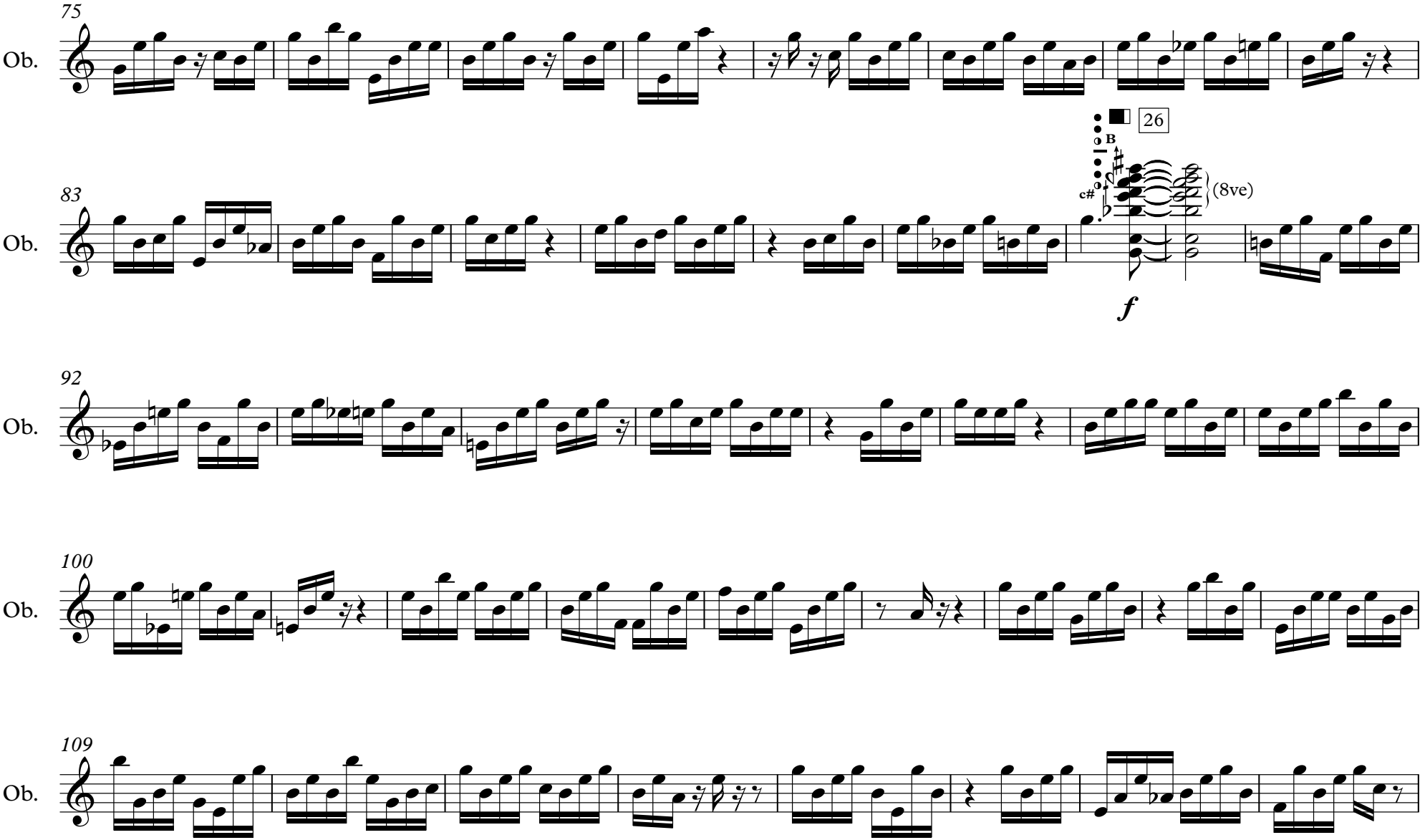
Ob.

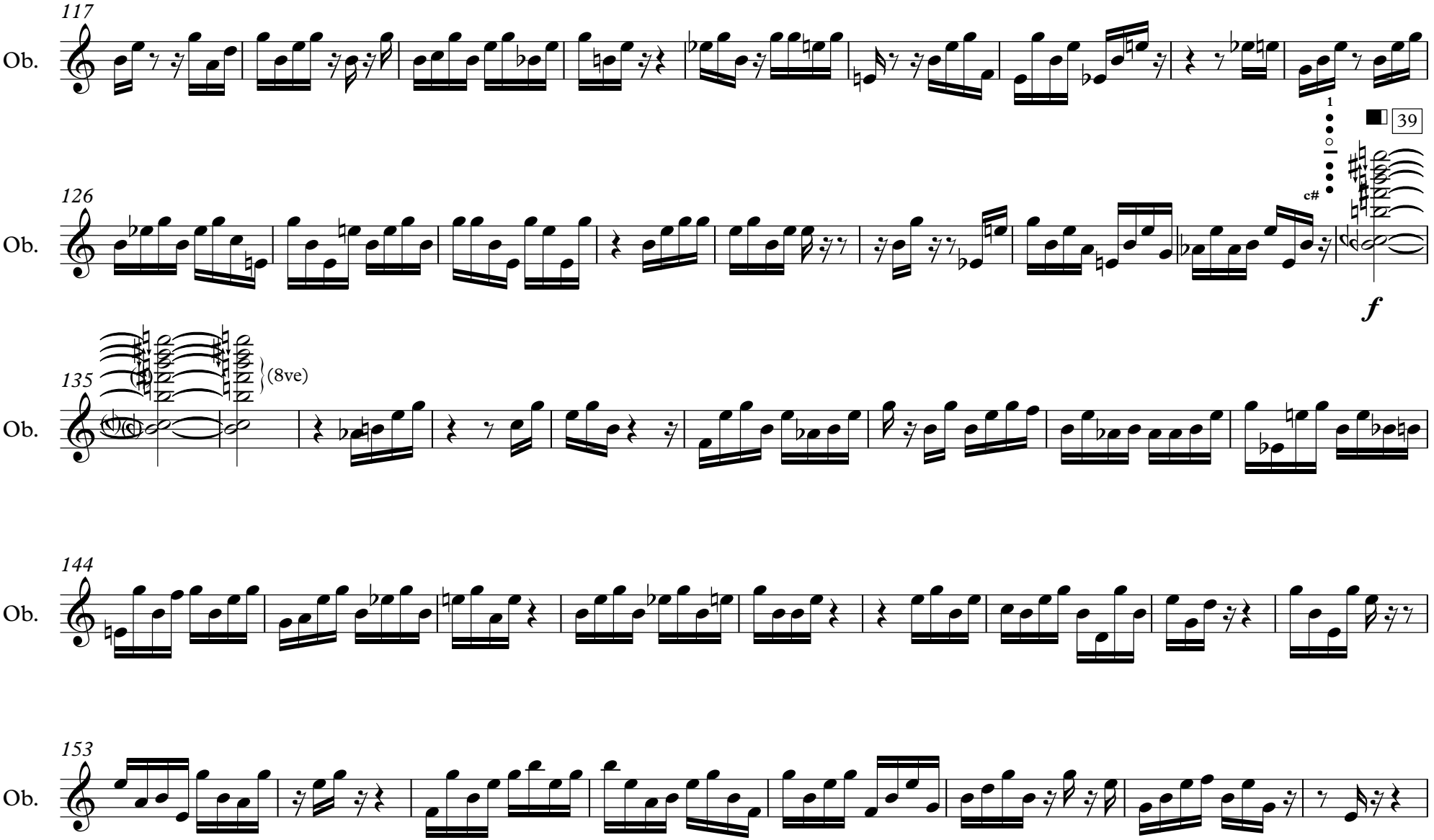


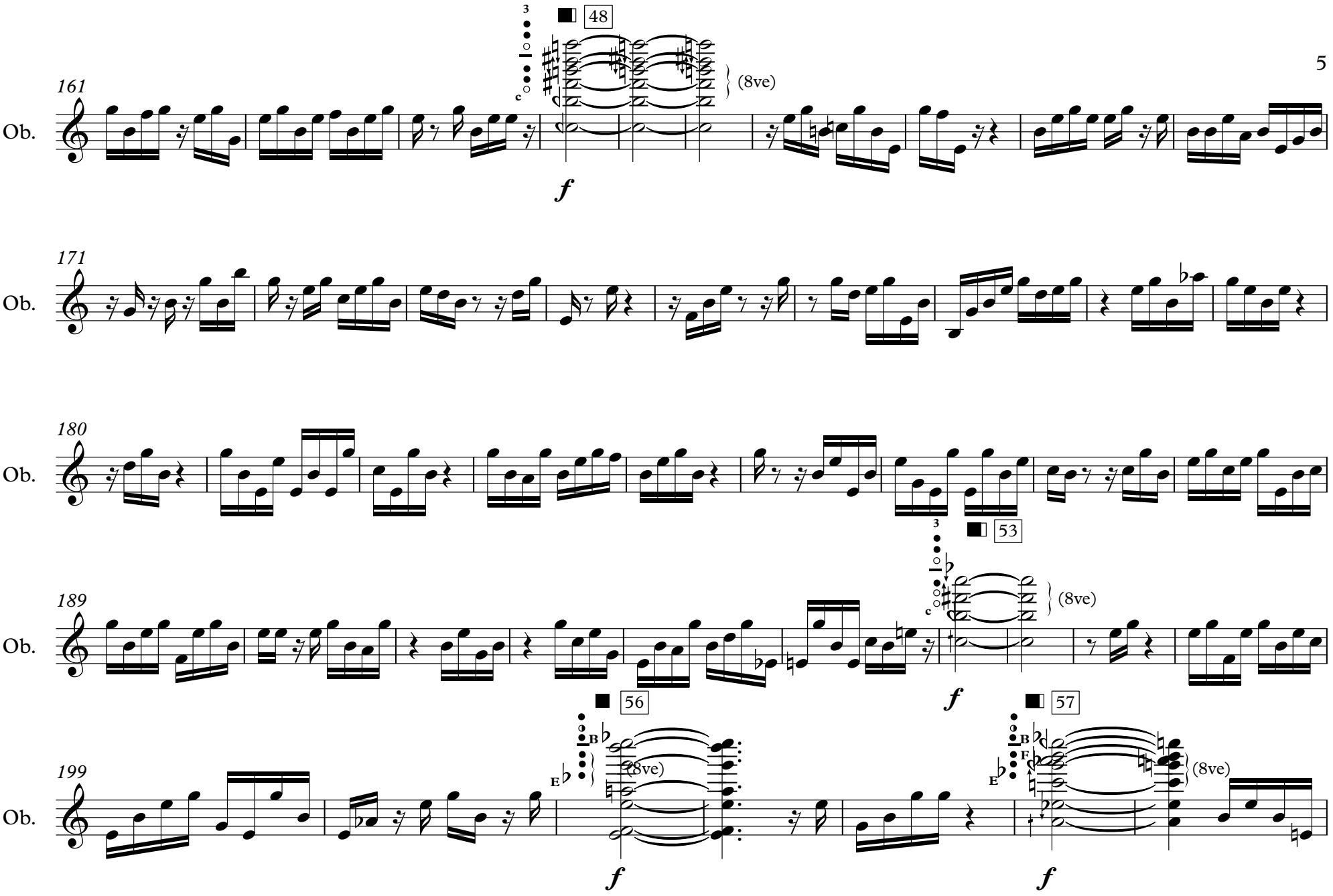
67

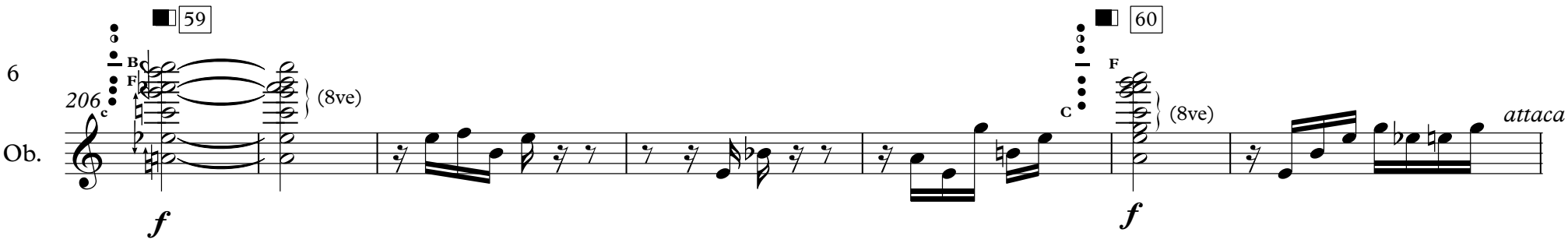
Ob.



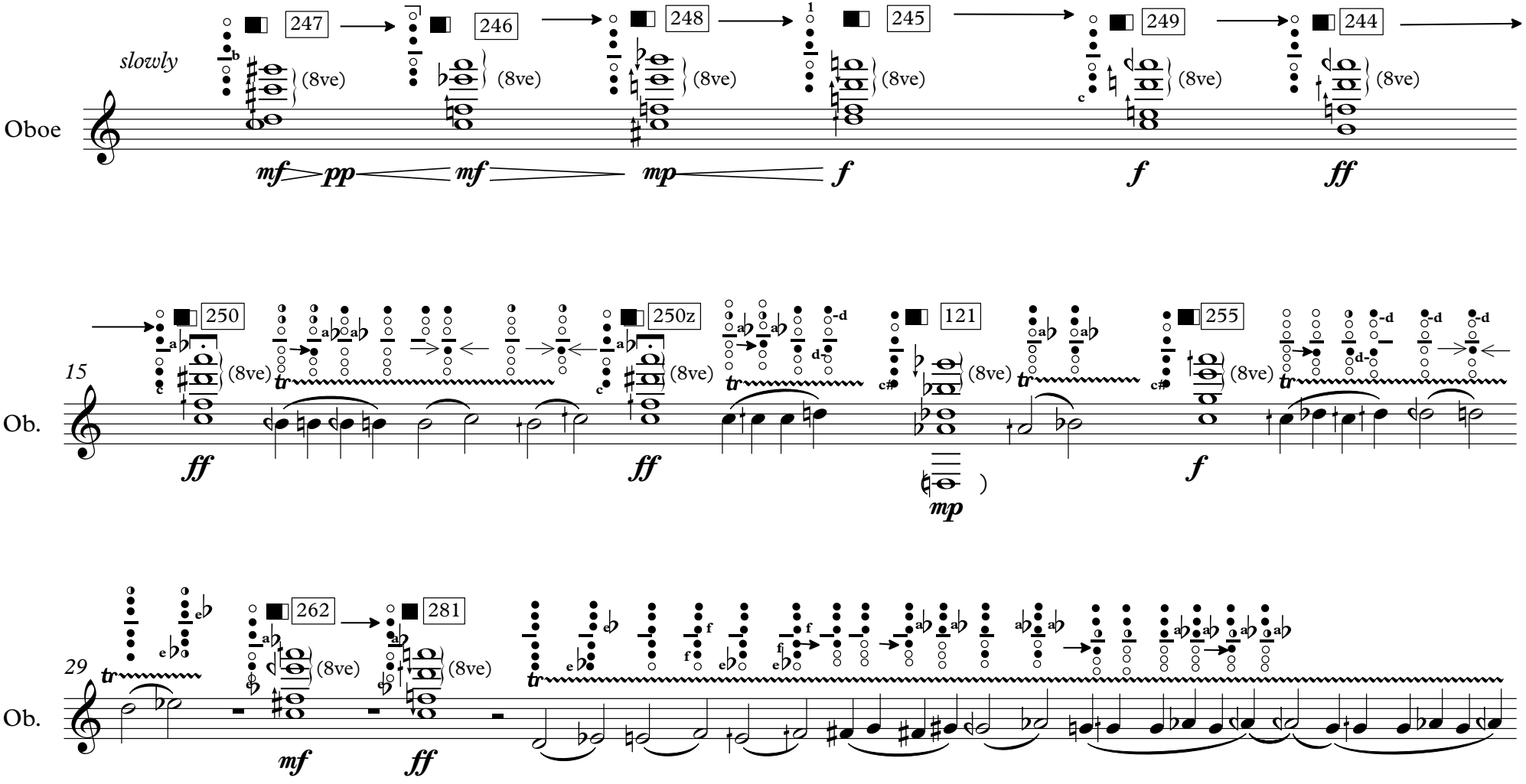

75
Ob.
83
Ob.
26
B
c#
(8ve)
f
92
Ob.
100
Ob.
109
Ob.

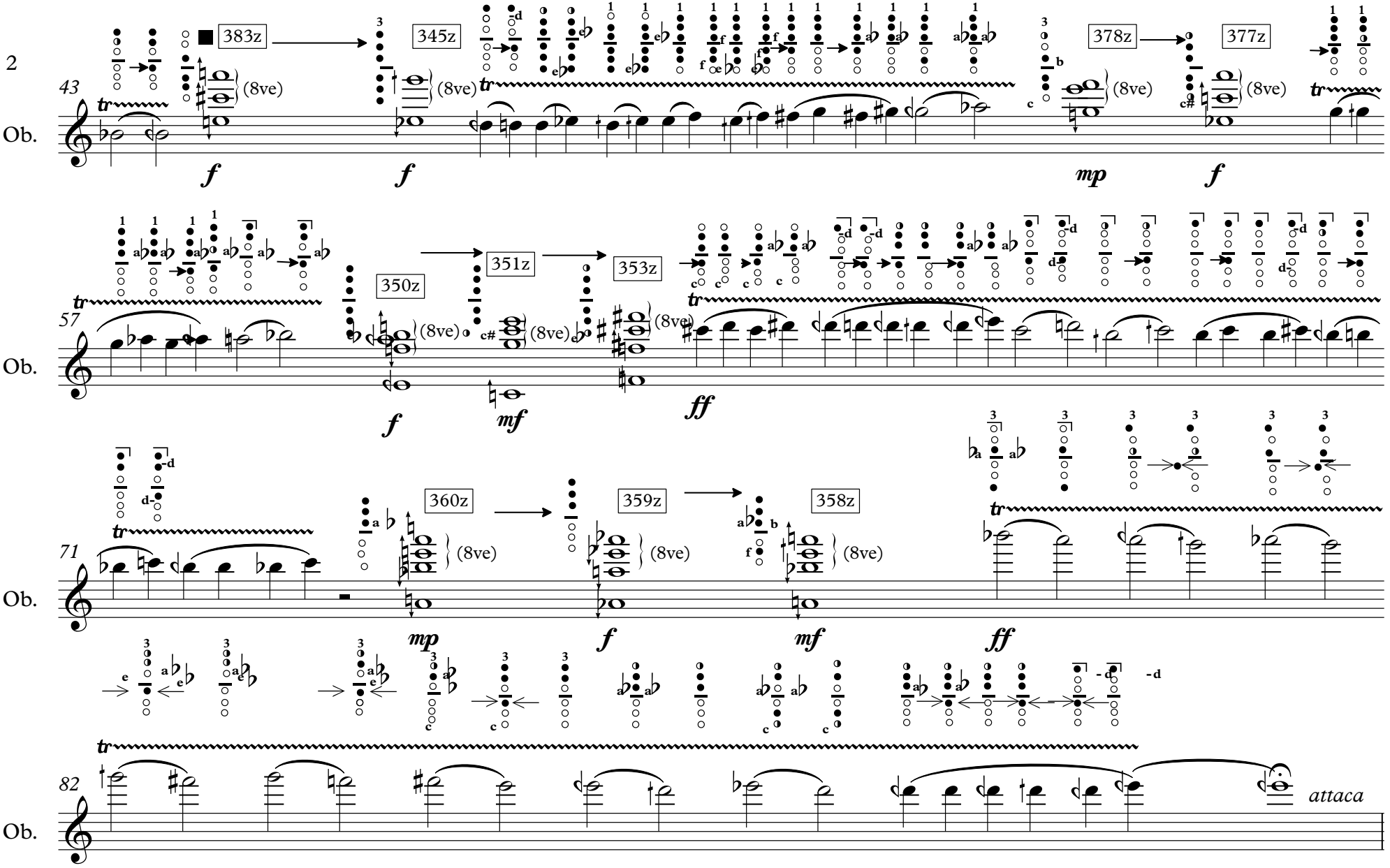

117
Ob.
1
39
126
Ob.
c#
f
135
Ob.
(8ve)
144
Ob.
153
Ob.


161
Ob.
48
(8ve)
f
171
Ob.
180
Ob.
189
Ob.
53
(8ve)
f
56
(8ve)
f
57
(8ve)
f
199
Ob.

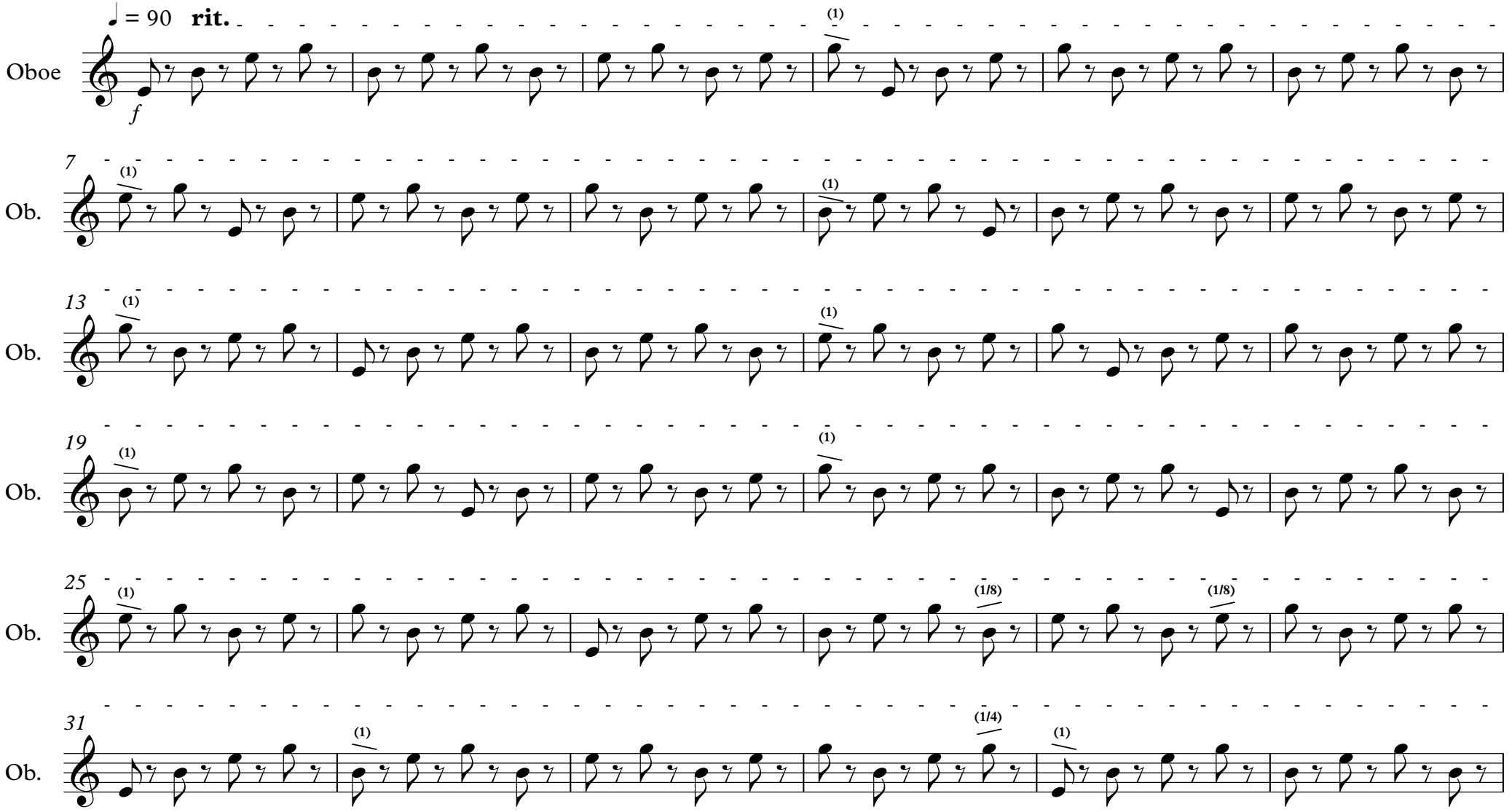

6
Ob.
59
206
B
F
(8ve)
f
60
F
C
(8ve)
f
attaca

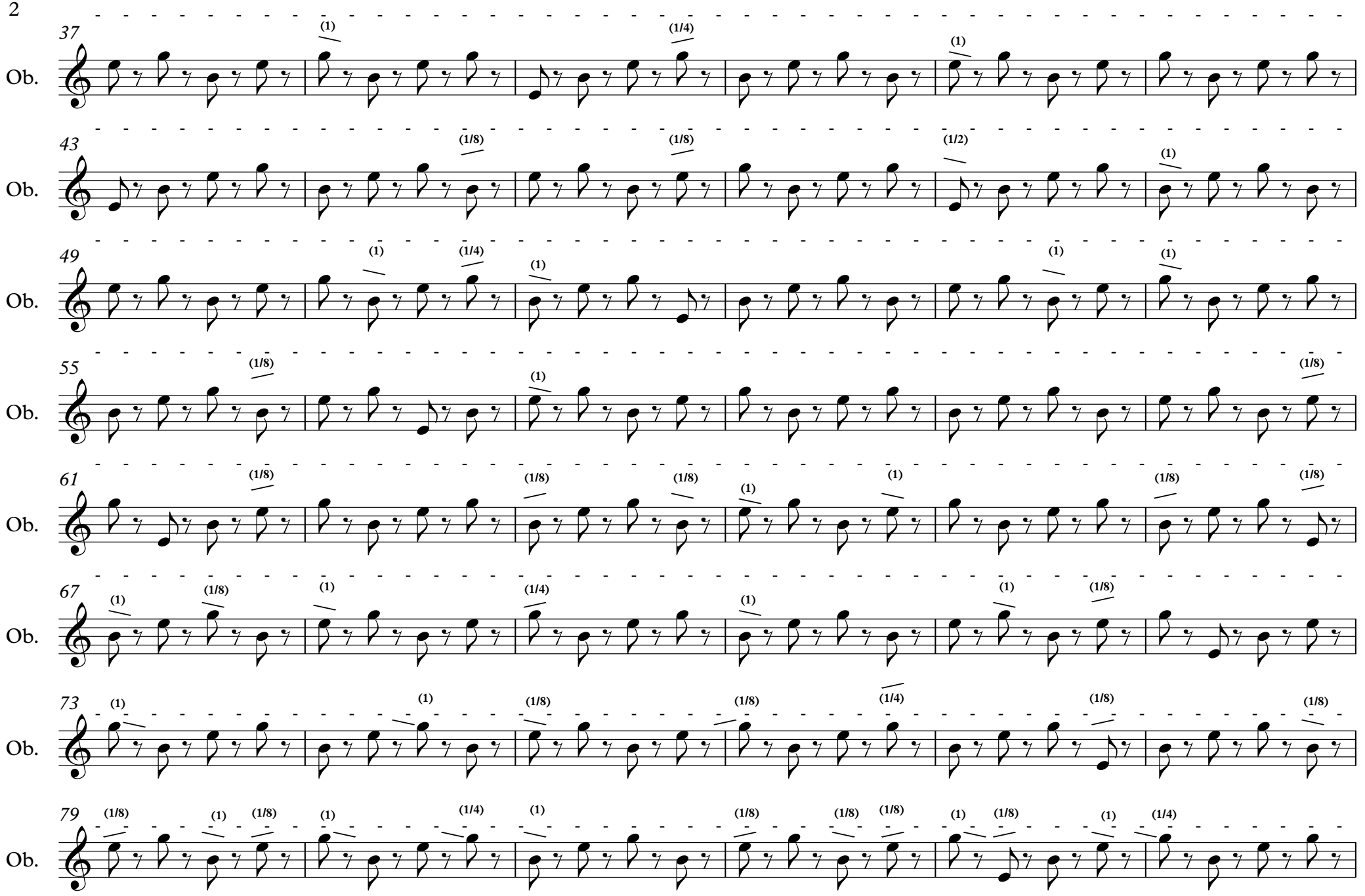
II.

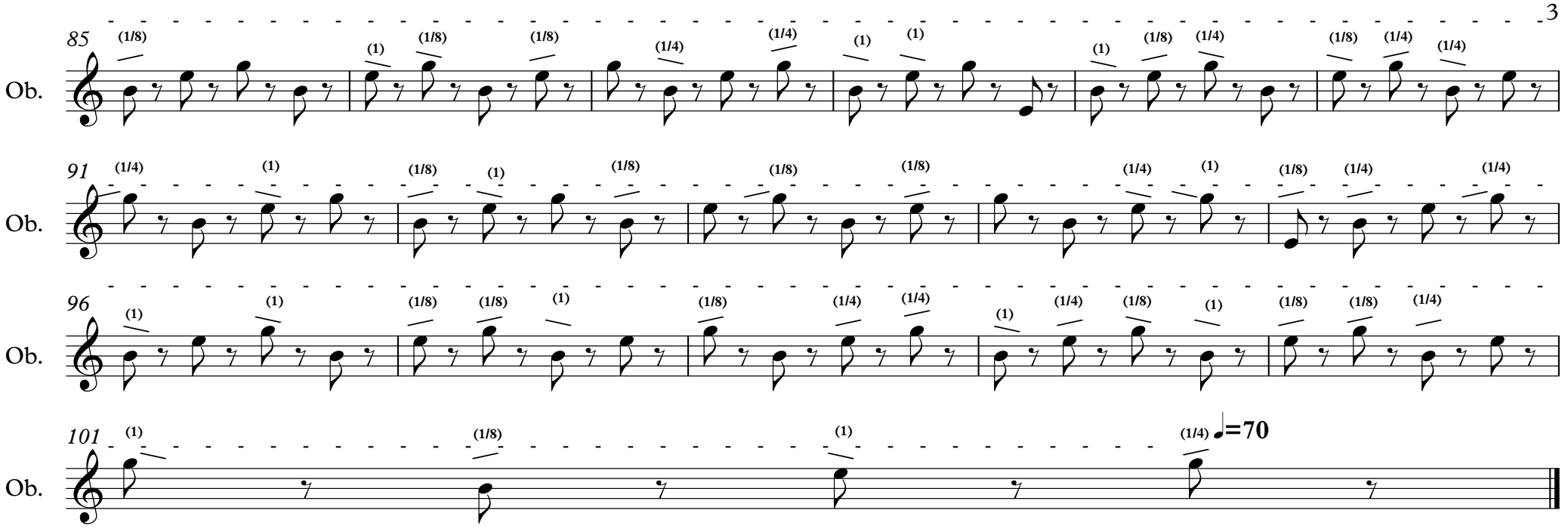

slowly
Oboe
247
246
248
245
249
244
(8ve)
mf
pp
mf
mp
f
f
ff
15
Ob.
250
250z
121
255
ff
ff
mp
f
29
Ob.
262
281
mf
ff


43
Ob.
383z
345z
378z
377z
(8ve)
f
f
mp
f
57
Ob.
350z
351z
353z
(8ve)
f
mf
ff
71
Ob.
360z
359z
358z
(8ve)
mp
f
mf
ff
82
Ob.
attaca

III.




37
Ob.
(1)
(1/4)
(1)
43
Ob.
(1/8)
(1/8)
(1/2)
(1)
49
Ob.
(1)
(1/4)
(1)
(1)
(1)
55
Ob.
(1/8)
(1)
(1/8)
61
Ob.
(1/8)
(1/8)
(1/8)
(1)
(1)
(1/8)
(1/8)
67
Ob.
(1)
(1/8)
(1)
(1/4)
(1)
(1)
(1/8)
73
Ob.
(1)
(1)
(1/8)
(1/8)
(1/4)
(1/8)
(1/8)
79
Ob.
(1/8)
(1)
(1/8)
(1)
(1/4)
(1)
(1/8)
(1/8)
(1/8)
(1)
(1/8)
(1)
(1/4)


85
Ob.
91
Ob.
96
Ob.
101
Ob.
=70