Introduction to Artificial Intelligence
Assignment 2: Accompaniment Generation

Deadline

Week 14 (29.11.22 23:59)

Submission

- Only one Python file with the code has to be uploaded to Moodle. Input files are NOT allowed. Name of the file should be like this *NameSurname.py,* e.g *IvanIvanov.py*. No other symbols allowed
- You should provide best output MIDI files received by your program (one output for each input file). Name of the files should be like this *NameSurnameOutputN-DetectedKey.mid*, e.g *IvanIvanovOutput1-C#m.mid*. No other symbols allowed
- Report describing algorithm flow in plain English. Name of the file should be like this *NameSurname.pdf* (For example, *IvanIvanov.pdf*). No other symbols allowed

Programming Language

Python 3.7+

Requests

- The program must work, the code should be readable, well-structured and should contain English comments
- It has to be only one **.py* file, several output **.midi* files and report **.pdf* file
- A report should contain a manual for running your program
- A report should describe key detection algorithm
- A report should provide detected keys for all input midi files
- NO extension of a deadline. Works sent after the deadline will be evaluated with a penalty (at most 100%)
- Assignment is individual
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies. We will compare your works with works of previous year students

Grading Criteria

- 40% for the code correctness
- 10% for correctly identified music key
- 10% for readability of code and comments
- 10% for the beauty and quality of the generated music
- 30% for the well-structured and informative report

Task

You are given several monophonic midi files, which are representing melodies. By using an evolutionary algorithm (EA), you have to generate an accompaniment for each of those melodies. You are allowed to use any type of EA, however, you are obliged to use both crossover and mutation for evolving accompaniment. Accompaniment should be represented by a sequence of chords. Each chord should contain exactly three notes. For simplicity you have to consider only the next types of chords:

1. major and minor triads
2. first and second inversions of major and minor triads
3. diminished chords (DIM)
4. suspended second chords (SUS2)
5. suspended fourth chords (SUS4)

More information about these chords can be found in the next section. The initial melody cannot be modified as long as tempo, time signature, etc. The output MIDI file has to contain two musical batches: the initial unmodified melody and the generated accompaniment. It means that both of those batches will be played in duet. The instrument of the original melody can be substituted by any other, but this are optional; moreover, the accompaniment can also be played by any instrument. Keep in mind that the choice of other instruments may lead to problems in playing some batches. Nevertheless, any other additional modifications to the original melody will be considered as a violation of the rules.

Assume that each input melody will be written in a non-changing tempo (which may vary for different inputs) and time signature (4/4 aka C or common time). For accompaniment, you should consider that each *bar* contains 4 quarter duration chords (see Fig. 1). There are two bars in this figure divided by a vertical line in the middle. Both bars contain 4 notes of the same duration of quarter, for the accompaniment, there will be 3 simultaneously played notes instead of 1 shown in the picture. Time signature defines the number of beats contained in each bar (the upper numeral) and which note value is equivalent to the beat (the lower numeral). In case of necessity, some chords may be replaced by rests. Do not overuse them. Rest is a piece of the batch when nothing is played. Easiest note durations are represented in Fig. 2. For indication of rests other symbols are used. MIDI notation is not equivalent to music, but some libraries compensate this difference.

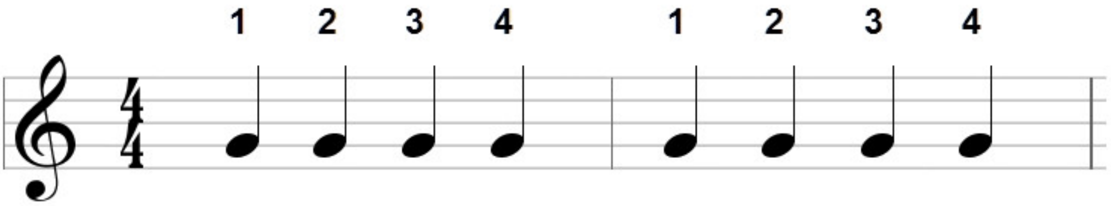


Figure 1. Bar, time signature, notes

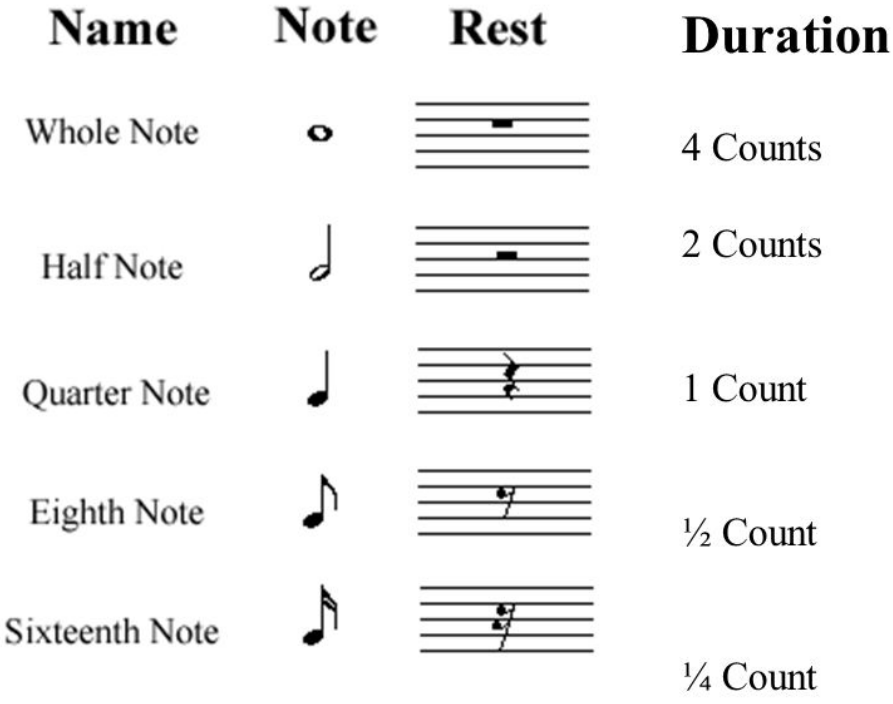


Figure 2. Note durations and symbols

Musical Theory

We are considering only Western music containing 7 notes in the *octave*. Technically, there are 12 notes, if we consider all possible pitches in octave (see Fig. 3). The order of notes in all octaves is always the same and repeated in each lower and higher octave.

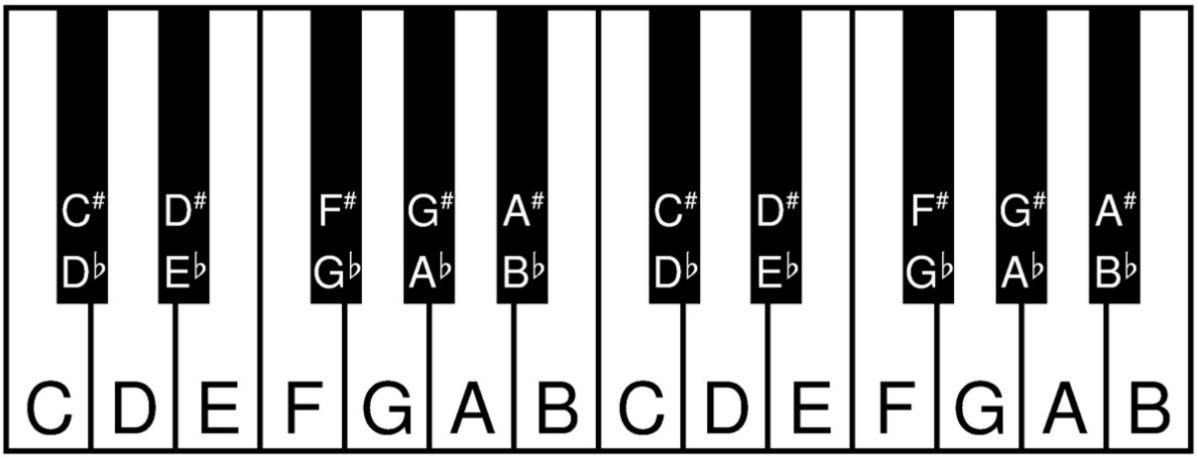
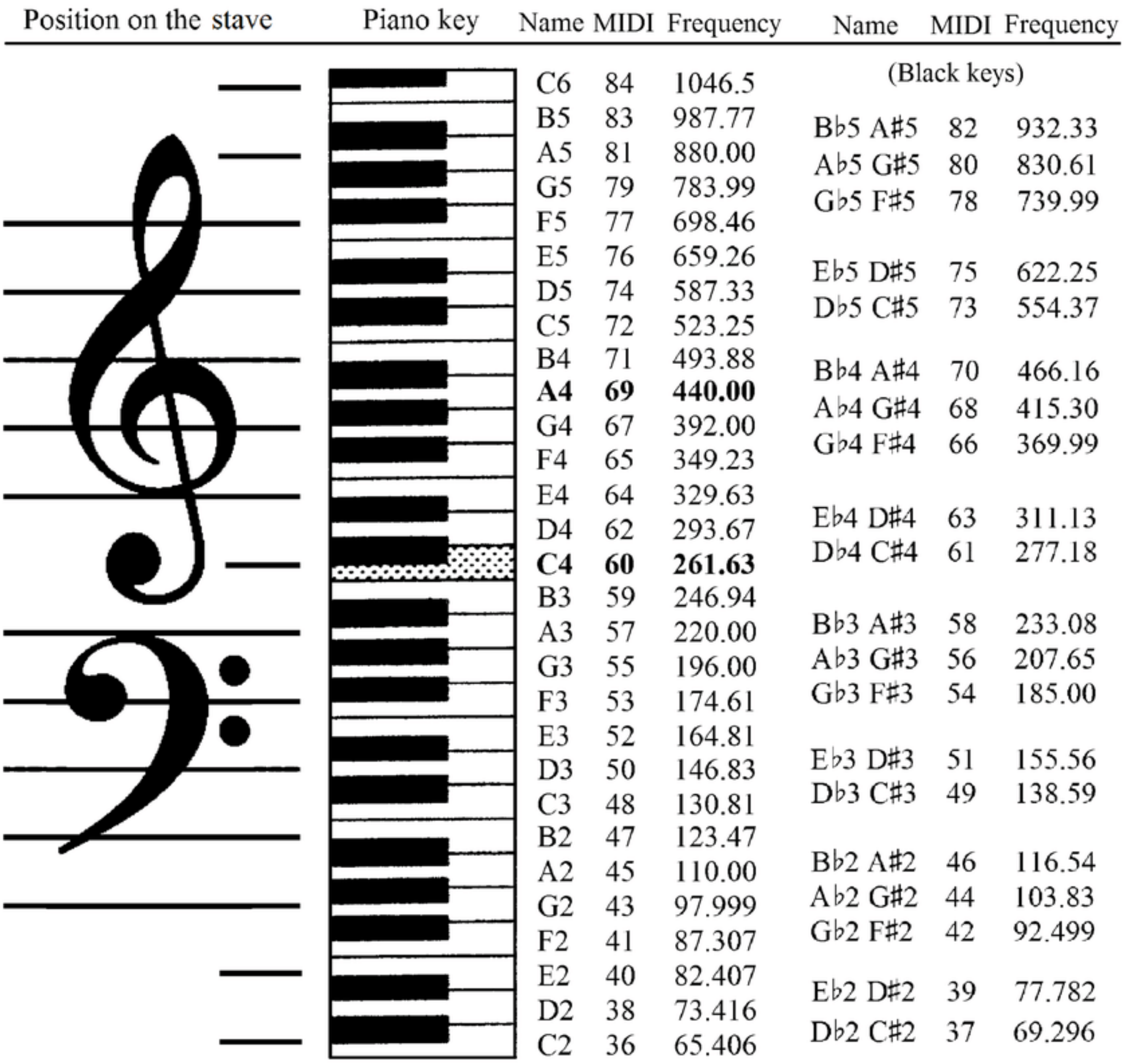


Figure 3. Octaves

Physically, each note is an abstraction over some frequency (see Fig. 4). There is a strong mathematical correlation between each note frequency. In MIDI each note has a value assigned from 0 to 128. In Figure 4 the list of notes is not full. Nevertheless, it is visible that each note's frequency is doubling in the next higher octave. These frequencies in Western music notation are true for all instruments. The same note played on different instruments will have the same frequency but a unique sound, which is called *timbre*.



Name	MIDI	Frequency
C6	84	1046.5
B5	83	987.77
A5	81	880.00
G5	79	783.99
F5	77	698.46
E5	76	659.26
D5	74	587.33
C5	72	523.25
B4	71	493.88
A4	**69**	**440.00**
G4	67	392.00
F4	65	349.23
E4	64	329.63
D4	62	293.67
C4	**60**	**261.63**
B3	59	246.94
A3	57	220.00
G3	55	196.00
F3	53	174.61
E3	52	164.81
D3	50	146.83
C3	48	130.81
B2	47	123.47
A2	45	110.00
G2	43	97.999
F2	41	87.307
E2	40	82.407
D2	38	73.416
C2	36	65.406

Name (Black keys)	MIDI	Frequency
B♭5 A#5	82	932.33
A♭5 G#5	80	830.61
G♭5 F#5	78	739.99
E♭5 D#5	75	622.25
D♭5 C#5	73	554.37
B♭4 A#4	70	466.16
A♭4 G#4	68	415.30
G♭4 F#4	66	369.99
E♭4 D#4	63	311.13
D♭4 C#4	61	277.18
B♭3 A#3	58	233.08
A♭3 G#3	56	207.65
G♭3 F#3	54	185.00
E♭3 D#3	51	155.56
D♭3 C#3	49	138.59
B♭2 A#2	46	116.54
A♭2 G#2	44	103.83
G♭2 F#2	42	92.499
E♭2 D#2	39	77.782
D♭2 C#2	37	69.296

Figure 4. Notes, their MIDI values and frequencies

Each music composition is usually written in a specific *key*. In Western notation, the key contains exactly 7 notes called scale degrees (see Fig. 5). Each of those scale degrees has an abstract name, do not confuse it with note name. Sometimes, music keys are altered into other keys for changing the mood of the composition. The key can be represented by two things: *scale* and *tonic*. The latter is the note in the first position of a key. In Figure 5 the key is *C Major* or simply *C* (omitted *M*), where the tonic is *C* (*Do* in notation common for Russian music culture inherited from Italian notation) and scale is *Major*.

In melodies with a non-changing key, it is easier to make a chord generation than in melodies with an alteration of the key. However, even in melodies with non-changing key not each sequence of chords belonging to the key will sound aesthetically pleasing. One of the directions for finding better chords is to avoid *dissonances* between the chord and melody note at least in the moment of the beginning of both. In other words, dissonant notes should not be pressed simultaneously, but non-simultaneous sounding of both may be applicable in some cases. Notes that are dissonant can sound harsh or unpleasant when played at the same time. Therefore, one of the ways is to use chords, containing the melody notes but in lower octaves. This may be a part of a fitness function. The melodies with alterations of a key require more analysis for a valid accompaniment generation.

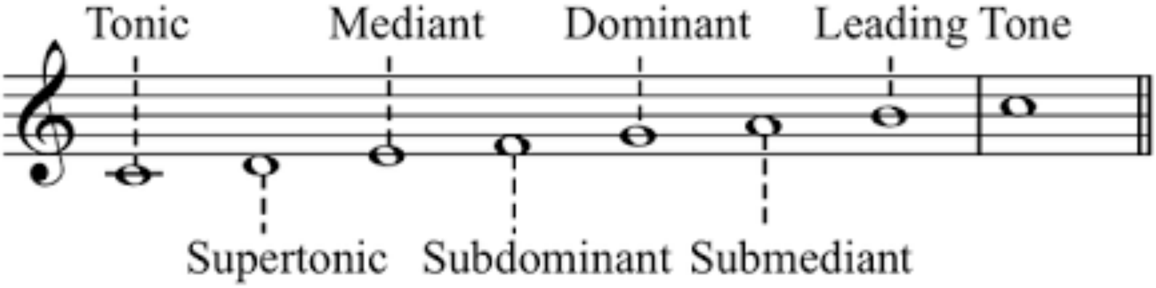
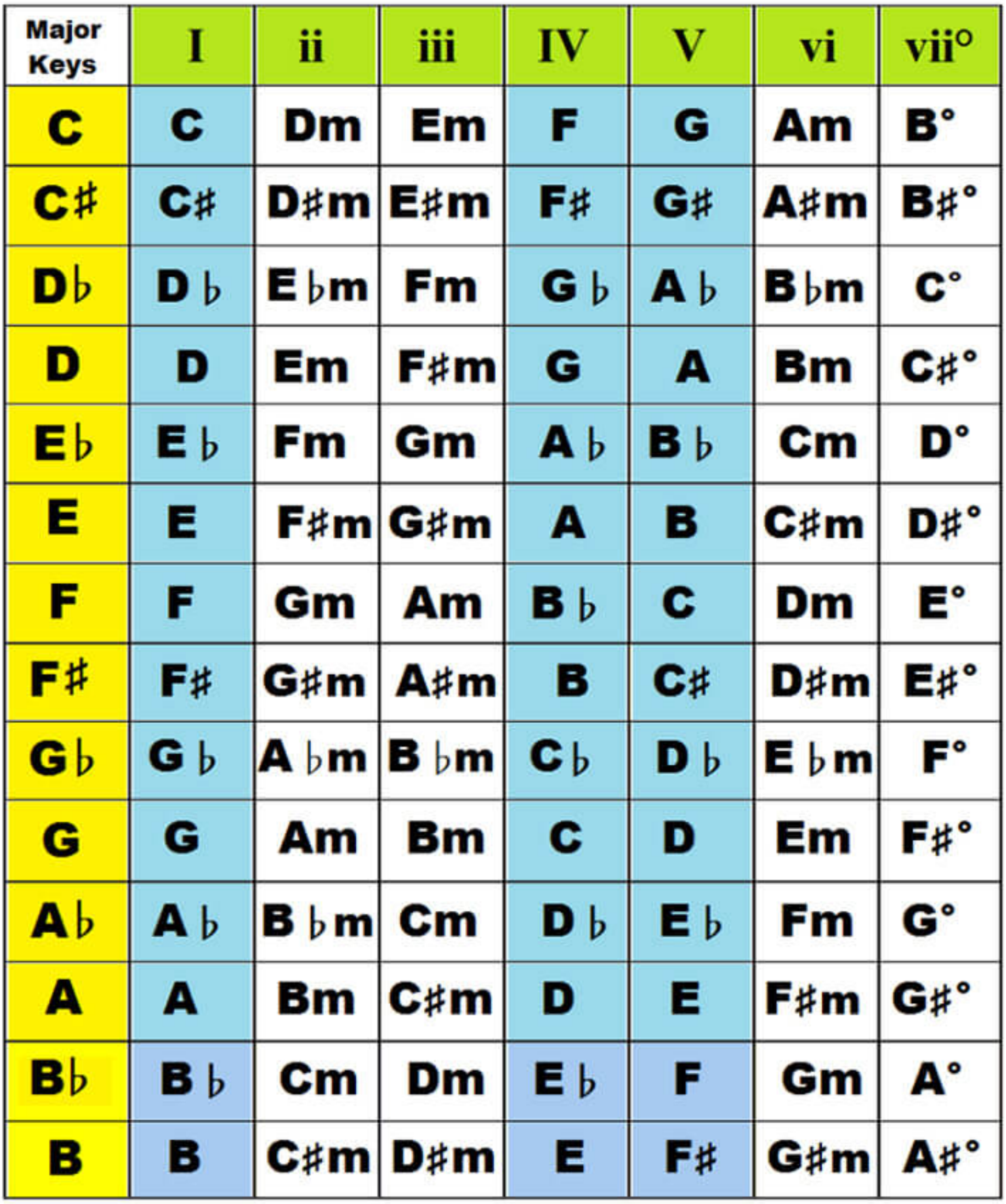


Figure 5. Scale degrees

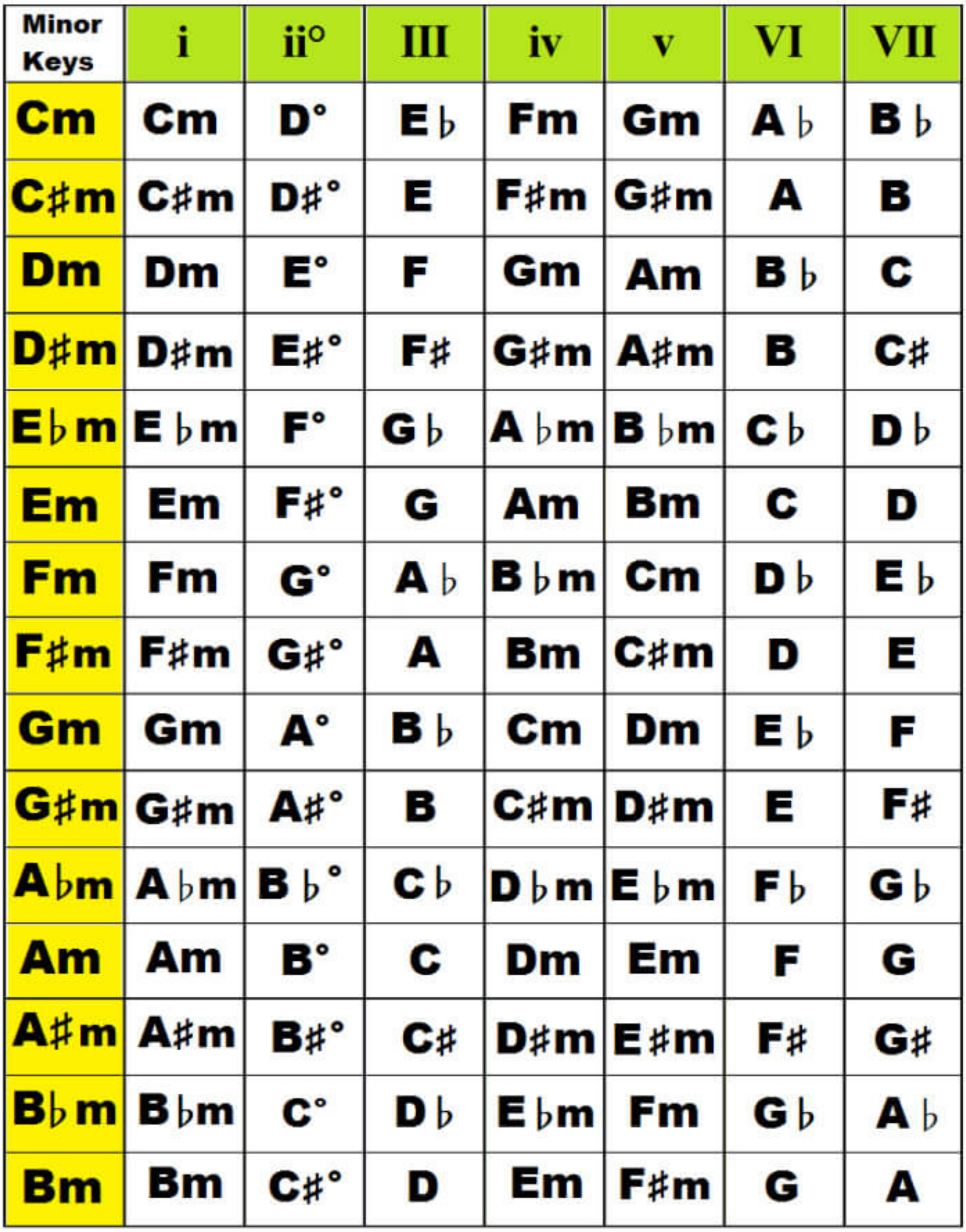
Scale is a set of musical notes ordered by pitch (fundamental frequency). As a limitation, we will use only *major* (*M* or usually omitted) and *minor* (*m* is never omitted) scales. Musically the former may lead to cheerful emotion, the latter to sadness. Mathematically they differ a little, both can be represented as integer offsets between scale steps (tonic, supertonic…leading tone, next octave tonic). For major it is 2, 2, 1, 2, 2, 2, 1, for minor it is 2, 1, 2, 2, 1, 2, 2. In music an offset of 2 is called *tone* and offset of 1 is called a *semitone*.

For simplification we consider only natural major and minor keys, and will refer to them as major and minor respectively. Assume that the number of considered keys is equal to the number of notes in octave multiplied by the number of considered scales 12*2=24 (see Fig. 6 and 7). Here we skip *enharmonic* keys because they include the same physical notes and sequence of MIDI values in the same order, e.g., *C#* and *D♭*, *F#* and *G♭*, *D#m* and *E♭m*, *G#m* and *A♭m*, *A#m* and *B♭m*. Those enharmonic keys are present in these Figures, exclude them from consideration.



Major Keys	I	ii	iii	IV	V	vi	vii°
C	C	Dm	Em	F	G	Am	B°
C♯	C♯	D♯m	E♯m	F♯	G♯	A♯m	B♯°
D♭	D♭	E♭m	Fm	G♭	A♭	B♭m	C°
D	D	Em	F♯m	G	A	Bm	C♯°
E♭	E♭	Fm	Gm	A♭	B♭	Cm	D°
E	E	F♯m	G♯m	A	B	C♯m	D♯°
F	F	Gm	Am	B♭	C	Dm	E°
F♯	F♯	G♯m	A♯m	B	C♯	D♯m	E♯°
G♭	G♭	A♭m	B♭m	C♭	D♭	E♭m	F°
G	G	Am	Bm	C	D	Em	F♯°
A♭	A♭	B♭m	Cm	D♭	E♭	Fm	G°
A	A	Bm	C♯m	D	E	F♯m	G♯°
B♭	B♭	Cm	Dm	E♭	F	Gm	A°
B	B	C♯m	D♯m	E	F♯	G♯m	A♯°

Figure 6. Major keys



Minor Keys	i	ii°	III	iv	v	VI	VII
Cm	Cm	D°	E♭	Fm	Gm	A♭	B♭
C♯m	C♯m	D♯°	E	F♯m	G♯m	A	B
Dm	Dm	E°	F	Gm	Am	B♭	C
D♯m	D♯m	E♯°	F♯	G♯m	A♯m	B	C♯
E♭m	E♭m	F°	G♭	A♭m	B♭m	C♭	D♭
Em	Em	F♯°	G	Am	Bm	C	D
Fm	Fm	G°	A♭	B♭m	Cm	D♭	E♭
F♯m	F♯m	G♯°	A	Bm	C♯m	D	E
Gm	Gm	A°	B♭	Cm	Dm	E♭	F
G♯m	G♯m	A♯°	B	C♯m	D♯m	E	F♯
A♭m	A♭m	B♭°	C♭	D♭m	E♭m	F♭	G♭
Am	Am	B°	C	Dm	Em	F	G
A♯m	A♯m	B♯°	C♯	D♯m	E♯m	F♯	G♯
B♭m	B♭m	C°	D♭	E♭m	Fm	G♭	A♭
Bm	Bm	C♯°	D	Em	F♯m	G	A

Figure 7. Minor keys

Knowledge about the key helps to find required chords because chords are based on notes allowed in the key. Figures 6 and 7 are containing information about all major and minor chords applicable in each key. Therefore, the key has to be detected for the evaluation of generated chord sequence.

A *major triad* is represented by integer notation as {0, 4, 7}. A *minor triad* is represented by integer notation as {0, 3, 7}. Here assume that 0 stands for the lowest note in the chord and the next numbers are offsets in MIDI, do not confuse it with scale steps. It means that only the middle note position is different. The first inversion of a major or minor chord is putting the lowest note of the chord one octave above. The second inversion of a major or minor chord is the same as the first inversion, but also with the original middle note of the chord put one octave above.

From the VIIth step of major scales and IInd step of minor scales major and minor triads cannot be constructed. Instead, diminished (DIM) chords have to be used, which by integer notation can be represented as {0, 3, 6}. In Figures 6 and 7 instead of DIM notation "°" sign is applied.

SUS2 chords can be represented by integer notation as {0, 2, 7}. However, it should be forbidden to generate such chords from the IIIrd and VIIth steps of major keys and the IInd and Vth steps of minor keys. Not following this may lead to unpleasant chords in the context of given keys.

SUS4 chords can be represented by integer notation as {0, 5, 7}. However, it should be forbidden to generate such chords from the IVth and VIIth steps of major keys and the IInd and VIth steps of minor keys.

Also, for chord generation more high-level approach may be applied in addition to key detection. It is called a *chord progression*. Based on the key and one of the existing sequences of chord numbers you may receive a valid sequence of chords. For example, chord progression *I-IV-V-V* in *C Major* will turn into chord sequence *C-F-G-G*. The problem is to understand which of the existing chord progressions should be applied for a given melody. Popular chord progressions may be found on the Internet. It has to be mentioned that chord progression does not define the duration of chords and the number of their repetitions because it is nothing more than a skeleton. Because of it, the previous example *C-F-G-G* can turn into *C* played with a whole duration, *F* played with a half duration, *G* played with a half duration and *G* played with a whole duration. So, the durations are not defined by chord progressions. Therefore, even correctly generated chord progression may not provide valid accompaniment.

Example

In Figure 8 the note representation of "Aqua – Barbie Girl" MIDI file is given. The MIDI file contains the next sequence of notes (note durations and rests are intentionally omitted):

G#5 E5 G#5 C#6 A5 F#5 D#5 F#5 B5 G#5 F#5 E5 E5 C#5 F#5 C#5 F#5 E5 G#5 F#5

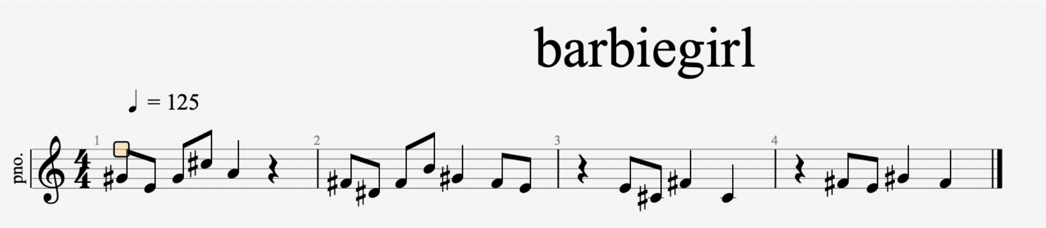


Figure 8. Input example in note format

The unique ordered set of these notes without octave number is {*C# D# E F# G# A B*}. After comparison with all considered keys, we receive 2 candidate keys, satisfying those notes: *C#m* and *E*. Those 2 keys are called *relative* to each other, this means that both of them are containing the same notes, but the scale steps are different:

- for the *C#m* key the tonic note is *C#*
- for the *E* key the tonic is *E*.

Generally speaking, the melody can end with any note of the key, but it is more popular to end with *stable notes* (tonic, mediant or dominant). It is a popular approach to find the key tonic by considering the last note (or chord root) of the composition. Ending by tonic is the most popular case, but it's not the rule. In the given example, we can see that the last melody note is F#, which does not give the answer about the tonic because it is not *C#* or *E*. Therefore, we should consider mediant (IIIrd step) and dominant (Vth step) notes. For the *C#m* key, it is *E* and *G#*, respectively. For the *E* key, it is *G#* and *B*, respectively. None of them is equal to the melody's last note. Still, the last note is not obliged to be one of the stable notes, it can be one of the *unstable notes* (supertonic, subdominant, submediant or leading tone). Ending with unstable notes is less popular.

We have to dig deeper. In general, it can be said that tonic is the most stable scale step. The second place can be given to the dominant note. The ratio of tonic and dominant frequencies is approximately 2:3 (see Fig. 4 for a comparison of any tonic and dominant notes of any key). At the same time the subdominant, which is considered an unstable note, has a ratio of 3:4 between tonic and subdominant frequencies. Nevertheless, its level of stability may be considered equal to dominant, because the ratio between the subdominant and the tonic of the next octave is equal to 2:3. Keep in mind that subdominant cannot be called stable, but in some exceptional cases may be extremely helpful for key detection. The third level of stability goes to mediant, which has a ratio between tonic and mediant equal to 4:5. Other scale steps are even less stable.

We already checked that the last note of the melody (*F#*) is not tonic for both *C#m* and *E* keys. Next fewer stable notes are dominant and subdominant. We checked that dominant notes for *C#m* and *E* keys are not equal to *F#*. However, if we check subdominant notes, we see that for *C#m* it is *F#* and for *E* it is *A*. So, we saw that the last note of the melody is a subdominant note of *C#m*, which makes the latter more probable candidate for being chosen as a key.

While ending with tonic is a very popular approach, beginning compositions from the tonic note is a less popular, but applicable approach. In our example, the melody begins with *G#*, which is not the tonic for both candidate keys. But at the same time, it is dominant and mediant for *C#m* and *E* keys, respectively. We already know that the dominant note is more stable, therefore *C#m* key is a more probable candidate for being chosen as a key.

Another less reliable approach is to consider the number of repetitions of melody notes, more stable notes and subdominant are more plausible to appear. The number of those notes in melody for *C#m* is 17 and for *E* is 10 notes, which makes the former a more probable candidate for being chosen as a key.

Thus, we may conclude based on several approaches that this melody is written in the *C#m* key. You may also consider other approaches. Keep in mind that in this example all 7 notes of the key were contained. The situation becomes different with a smaller number of notes, which requires considering more candidate keys.

After the accompaniment generation, you may receive different results. In Figure 9 you may see the next sequence of chords:

C#m C#m F#m F#m B B E E C#m C#m F#m F#m Bsus4 Bsus4 B B

In this output most of the chords were built upon the root notes from the 3rd octave, only *B* and *Bsus4* were having root notes from the 2nd octave. As it can be seen, the repetition of chords is applicable, but you are also allowed to make chord inversions or change octaves or anything else for increasing the variety. Any valid solution may be applied. In the output all the chords belong to the *C#m* key, except *Bsus4*. This is a sustained chord that does not belong to major or minor scale as it was mentioned earlier in the previous section. However, it is possible to apply it, but was not mandatory. Bsus4 chord contains *B*, *E*, *F#*, which suits 2 melody notes played simultaneously: *F# E*. This sounds more interesting than *B* chord, containing *B*, *D#*, *F#*, containing only 1 note intersection.

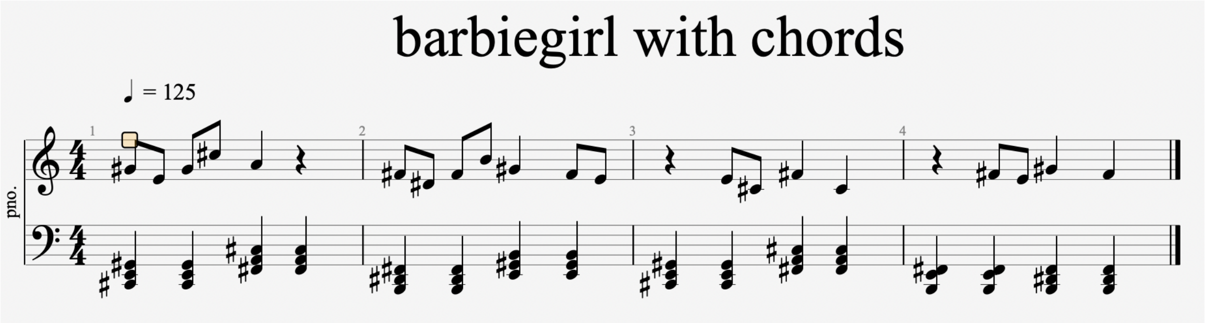


Figure 9. Output example in note format

Recommendations

For simplification as accompaniment, you are recommended to use only chords containing three notes. Though you are allowed to use more complicated chords, this will increase the search area. The same is true about the number of chords in each bar. You are recommended to use only 4 quarter duration chords in each bar, but for achieving more varied results you may use other durations. And again, this will significantly increase the search space and also create some limitations.

For a better understanding of musical theory, you are recommended to spend more time watching some extra videos and reading articles. Knowing the subject area is very important for achieving good results. This will help you to come up with a better fitness function.

Useful Links

1. Python midi library - https://mido.readthedocs.io/en/latest/
2. Online midi sequencer - https://onlinesequencer.net