

Strings Review

Colour Chords

Lecture 13

In music, there are many ways to play chords. For example, the most typical chords in popular music include:

- C-major
- G-major
- A-minor
- F-major

However, there are many variations to these chords that are used to give a song a specific **"feeling"** (i.e. making a song sound "jazzy", "evil", "cheesy", and/or other abstract descriptors).

For example, we can make the four chords above sound "jazzy" by adding more notes to the chord, and therefore adding **"colour"** to them:

- C-major-seven (a.k.a. Cmaj7)
- G-dominant-seven (a.k.a. G7)
- A-minor-seven (a.k.a. Am7)
- F-major-seven (a.k.a. Fmaj7)

Note: The "colour" here is the addition of the dominant, major, and minor sevenths to the original chord.

There is an almost endless amount of chord and colour combinations to make a song have a specific type of "feeling", but for this problem, we will focus on only a select few.

Write a program that will do the following:

- Ask the user to enter a chord in the following format: `ROOT TYPE COLOUR` . For example, if the user wants to enter an A-minor-seven, they would enter the following (note the spaces between each component of the chord):

A \natural m7

Note: \natural is a symbol representing a "natural" note. A natural note is a note that is neither flat (\flat) nor sharp (\sharp).

It doesn't really matter if you don't know what this means—they are just options.

I have already included the valid components of chords in the file **is_valid_chord.py**, so you don't need to worry about knowing what they are or how they sound like:

```
ROOTS = "A B C D E F G"
TYPES = " $\natural$   $\flat$   $\sharp$ " # Natural, flat, and sharp
COLOURS = "maj m 7 m7 6 m6 dim sus4 dim 7 $\flat$ 13  $\sharp$ 9 7 $\flat$ 9"
INVALID_NOTES = "B $\sharp$  C $\flat$  F $\flat$  E $\sharp$ "
```

Note: INVALID_NOTES includes note notations that don't exist in western music.

- The program will determine whether the inputted components of the chord will form a valid chord. A valid chord is a chord that comprises a valid root from ROOTS , a valid type from TYPES , and a valid colour from COLOUR . If the root-type combination exists in INVALID_NOTES , the entire chord is invalid.
- The program's output will be in **exactly** the following format:

```
Enter a chord: [ROOT TYPE COLOUR] B  $\natural$  sus4
B $\natural$ sus4 is a valid chord!
```

```
Enter a chord: [ROOT TYPE COLOUR] F  $\flat$  6
F $\flat$ 6 is not a valid chord!
```

Tip: In order to test the natural, sharp, and flat symbols, just copy one from your code and paste it into your shell. That's how I've been doing it.

- **Optional:** Allow the user to enter various chord component combinations until they enter the word " DONE ":

```
Enter a chord: [ROOT TYPE COLOUR] C  $\natural$  maj
C $\natural$ maj is a valid chord!
```

```
Enter a chord: [ROOT TYPE COLOUR] B  $\sharp$   $\sharp$ 9
B $\sharp$  $\sharp$ 9 is not a valid chord!
```

```
Enter a chord: [ROOT TYPE COLOUR] C-sharp minor
```

C-sharpminorC-sharp minor is not a valid chord!

Enter a chord: [ROOT TYPE COLOUR] DONE

Note: You might get a `UnicodeDecodeError` when using the natural, sharp, and flat symbols. I'm pretty sure that there is a very easy way to fix that, but I'm too lazy to look it up. Just try running your program again, and it should (hopefully) go away. That's what happened to me, anyway.