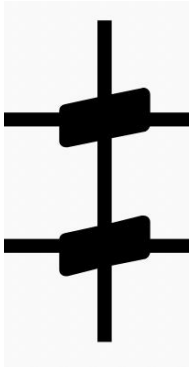
Strategy:

- Modify the .musicxml file directly
- There's a place that describes key signature.

key-step: note to alter
key-alter: amount (in semitones) to change key-step
key-accidental: description of actual accidental shape/properties

Example:
key-step: F
key-alter: 0.5
Key-accidental: quarter-sharp



```
<key>
  <key-step>F</key-step>
  <key-alter>0.5</key-alter>
  <key-accidental>quarter-sharp</key-accidental>
  <key-step>G</key-step>
  <key-alter>0</key-alter>
  <key-accidental>slash-quarter-sharp</key-accidental>
  <key-step>A</key-step>
  <key-alter>-2</key-alter>
  <key-accidental>flat</key-accidental>
  <key-step>B</key-step>
  <key-alter>0</key-alter>
  <key-accidental>slash-flat</key-accidental>
  <key-step>E</key-step>
  <key-alter>1</key-alter>
  <key-accidental>sharp</key-accidental>
  <key-step>C</key-step>
  <key-alter>-0.5</key-alter>
  <key-accidental>quarter-flat</key-accidental>
  </key>
```

Examples

- How we change key signature depends on the makam name in the title.
- A list of tuples, each a set of makam names and the resulting change
- Then, change key signature changes to xml (e.g. [“B -0.5”, “F 1”] to .xml)

```
def strks_to_xml(strks):
    """
    Helper. Takes strks, which is a string for key signature in the format:
    "note Δhalf-steps"

    and turns them into XML.
    """

    key_step = strks[0] #just the first letter. varname \approx strname
    key_alter = NUM_TO_KV[strks[2:]] #second til end is the number.
    key_accidental = NUM_TO_ACC[strks[2:]] #gives the right name

    xmlstr = "" #this will provide the full name
    xmlstr += "<key-step>" + key_step + "</key-step>\n"
    xmlstr += "<key-alter>" + key_alter + "</key-alter>\n"
    xmlstr += "<key-accidental>" + key_accidental + "</key-accidental>"

    return xmlstr #return the full XML.
```

```
NAME_TO_MSTRKS = [
    ({"Rast", "Evc", "Rehavi", "Pençgâh", "Sazkâr", "Sâzkâr", "Nevâ", "Nevâ Bûselik", \
        "Tâhir", "Tâhir Bûselik", "Hüseynî", "Hüseynî ʿAşîrân", "Gülizâr", "Yegâh", \
      "Beyâtî Araban", "Muhayyer", "Muhayyer Bûselik", "Gerdâniye", \
      "Gerdâniye Bûselik", "Dilkeş Haverân", "Evc Bûselik", "Irâk"}, ["B -0.5", "F 1"]), \
    ({"Acem Kürdî", "Acem Aşîrân", "Kürdî", "Muhayyer-Kürdî"}, ["B -1"]), \
    ({"Muhayyer Sünbüle", "Sabâ", "Sabâ Bûselik", "Sabâ-Zemzeme", \
      "Bestenigâr", "Şevk Efzâ", "Şevk-i Tarab", "Şevk ü Tarab"}, ["B -0.5", "D -0.25"]), \
    ({"Uşşak", "Beyati", "Beyati- Buselik", "Isfahan", "Berte-Isfahan", "Hisar"}, ["B -0.5"]), \
    ({"Nişaburek", "Eve Arâ", "Ferahnâk"}, ["F 1", "C 1"]), \
    ({"Suzidilara", "Mahur", "Mahur-Buselik", "Zavi*"}, ["F 0.25"]), \
    ({"Nikriz", "Şehnaz"}, ["B -0.25", "C 1"]), \
    ({"Hisar-Buselik", "Suzdi"}, ["F 0.5", "G 1", "D 1"]), \
    ({"Segâh", "Müstear"}, ["B -0.5","E -0.5", "F 1"]), \
    ({"Hüzzam", "Karciğar", "Beyati Araban-Buselik"}, ["B -0.5", "E -0.25", "F 1"]), \
    ({"Kürdili Hicazkâr"}, ["B -1", "E -1", "A -1"]), \
    ({"Hicazkâr"}, ["B -0.5", "E -0.25", "A -0.25", "F 1"]), \
    ({"Neveser"}, ["B -0.25", "E -0.25", "F 1", "C 1"]), \
    ({"Hicaz", "Hicaz Aşiran", "Hicaz-Buselik"}, ["B -0.25", "F 1", "C 1"]), \
    ({"Arazbâr", "Arazbâr-Buselik"}, ["B -0.5", "E -0.5"]), \
    ({"Buselik-Aşiran"}, ["F 1"]) \
    ]
```

Function Snippets

- Search for makam and modify the .xml file directly
- Through inspection we found that the key attribute is always stored within the 2nd component of the 2nd component of the 1st component of the 6th component of the root (ie entire file), hence indices [5][0][1][1]

```
for folder in os.listdir(filepath): #"." refers to current directory.
    #now, we know the string of folder, and can access all the files within!
    if folder[0] != ".": #see tech note below
        for musicfile in os.listdir(filepath + "/" + folder): #like "./CT 1-5
            """
            Quick tech note: "." indicates a file is HIDDEN.
            We don't wish to access hidden files, like .DS_Store and .mscbackup.
            Therefore, if a file starts with ., we are guaranteed that it's hidden.
            Of course, we don't have pathological file names that intentionally
            start with ".", so we assume start with "." -> don't consider it.
            """
            if musicfile[0] != "." and ".musicxml" in musicfile: #ignore if otherwise
                #now we are faced with JUST files. Now modify!

                #figure out which name is inside
                for name in ALL_NAMES:
                    #at least one must be true
                    if name in musicfile:
                        mstrks_cor = mstrks(name) #find the correct list of key sigs to modify

                #at this point mstrks must exist.
                insertks(filepath + "/" + folder + "/" + musicfile, \
                        mstrks_to_xml(mstrks_cor)) #gives us correct modification
```

```
#start with xml
print("FILE PATH:", filepath)
tree = et.parse(filepath)
root = tree.getroot()
print("ROOT:", root)

"""
this is where we need to hard code this number.
Within Musescore, our root is just the score-partwise.
Now with that being said, the 6th child (index 5) is called part id="P1".
Then, within that, the 1st child (index 0) is called measure number="1" etc
And the second child (index 1) is called attributes.
finally, the second child within that (index 1) is called key.
"""
desiredroot = root[5][0][1][1] #this is where the part of the key is stored
print("DESIR...oot)
```