a. Notable obstacles that I overcame was creating the code for translateTune to work for some cases. I had to keep in mind many different variables and wrote them down on paper in order to make sure I was able to follow my code. It was very difficult to create a function that worked. Another obstacle would be getting the '[' and ']' to work for some cases since it included factoring a variety of more variables which caused more stress and headaches.

b. For my code, my isTuneWellFormed first checks to see whether or not the tune ends in a '/' because it wouldn't be well formed if it didn't end in a '/'. Afterwards, the function goes through the entire string and checks to see whether the character at the location is a valid character, then checks for if there is an accidental sign after, and finally checks for whether or not a digit which represents the octave is there. Otherwise, it just checks to see if there is a digit after the letter or if a letter follows while also keeping in mind skipping iterations to the next start of a note. My translateTune function first checks whether the tune is well formed using the previous method. If it is not well formed, it returns 1. Then it checks whether or not it is playable which is described in a different method which makes sure to check for the special cases of notes. If the tune is playable but not well formed, it would set badBeat to the corresponding value of the beat where it fails using a different function called badBeatValue which keeps note of the number of beats and return 2. After this, I create a multitude of variables in order to make sure that I can create '[' and ']' in case there are chords as well as whether or not there is a chord. First it goes through the string and checks to see whether or not k-1 == length which is the case for my code since my other functions which grab the notes increase the incremental within the function itself. It then takes a substring and stores it in the string now and tests to see whether or not it is a chord by going through the splice of the string and seeing if there is more than one letter. Then it checks for a '/' which would be represented in the new code as a space character. It then checks to see if the splice is a chord and then if its the first note in the chord. If it is, it will add the character '['. It then obtains the note and then translates it and adds it to instructions. The function also checks whether or not the end of a chord has arrived and to place the ']' symbol. At the end it returns 0 since the function returns an int.

c. Test data to check:

for bool isTuneWellFormed(string tune)

G/ Just one note

C/

D/

A3C#E//E//F#3A3D4/A3C#E/ chords as well as rests

B5D#C/E//Gb/

C0C0DC0DD/E#FbB#Cb/B#9/ same notes in a chord

B5B5B5/C#3C#3/D5C3/

///       only empty beats

//

/

for translateTune(string tune, string& isntructions, int& badBeat)

"C/C/G/G/A/A/G/" Tune with only 1 basic note in each beat

"D/E/C/A/B/C/"

"D3/F#3/A3/D4//D3F#3A3D4/" Tune with a chord in it as well as accidentals and integers

My code fails with the below:

"G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/F#5/B3G5//G//G/" Complex tune with chords and beats with no notes

(misses some ']' signs and has extra spaces)

"DADDA/" Tune with one beat and multiple notes that are the same as a chord.

(misses the last ']' at the end)