A)

The major error I made was i wrote my program in a way such that it tried to examine elements of the string array that were bigger than the size of the string, i.e. for example if the string were "ABC/" my code would sometimes try to examine the 5th element of the string (which only has 4 elements). I solved this before completing "isTuneWellFormed"

My "translateTune" function was a complete mess until i split it up into subfunctions. Additionally, setting "badBeat" was difficult because I was unable to think of a way to easily figure out what beat I was in, until i read the instructions more closely to notice that all the beats are set off by '/' marks, so i used those to determine beats

Another issue I faced was that while writing my code, i would sometimes get an error that said "not all control paths return a value." I understood what the error meant, but it took a long time to identify what cases of combinations of of my boolean statments that i did not write a "return" statement for.

I could not figure out how to implement brackets.

B)

The program is designed with several functions. The main function I wrote more or less just takes in the string input. Depending on the translateTune function, it will either show the translated tune, tell the first bad beat, or say that the input is not well formed.

translateTune is the essential function. first, it sees if the tune is well formed or not by using the value returned from isTuneWellFormed. isTuneWellFormed checks all of the different kinds of characters in the string, and makes sure that the characters in the string are typed in the proper order (for instance, a sharp cannot come right after a number). If the tune is well formed, then translateTune checks if it is playable or not using the isPlayable function. The isPlayable function also checks for badbeats and sets badbeat as necessary. if the tune is fully playable, then convertTune actually builds the converted tune.

convertTune is incomplete. what it does do is apply proper formatting to the first and last entry of the tune, and it translates all the spaces and all the notes in the song using processNote.

the function processNote passes on to translateNote, the function provided to us, and then it takes the character that translateNote created, and adds it to the string that gets put into instructions at the end.

C)

"" the blank statement, which should be tested because in the instructions a tune with zero beats should pass

A/ the simplest test case with one note in a beat

Ab/ a test case of a single accentted note

Ab3/ a test case of a single accented note in a different octave than the original octave

A/B/C/D/E/F/G/ all the notes played one at a time in sequence in beats

Ab/Bb/Cb/Db/Eb/Fb/Gb/ any accented note played one at a time in sequence in beats

Ab3/Bb5/Cb4/Db3/Eb4/Fb3/Gb5/ any accented note with numbers played one at a time in sequence in beats

ABCD/ Multiple notes in one beat, couldn't implement brackets

ABCD//ABCD/ multiple notes in one beat, an empty beat, multiple notes in one beat, couldn't implement brackets

//// four empty beats