Brian Tagle

604907076

a. One of the obstacles I overcame was having a lot of code that I could have simplified down. Especially early on I wrote a lot of code several times over and I could have simplified that code If I had just wrote a single function to house it. Later on in writing the program I did write some functions that make things simpler down the line but I didn’t go back and use the function in the code I had already written because my previous code worked and I didn’t want to risk changing it. One other obstacle I overcame was the conversion of chars to ints. This was a problem when I was using the given encodeNote function and the octave was way out of range because the ascii character value assigned to ints were not the ints themselves, so I had to make adjustments to make sure everything worked.

b. I used hasCorrectSyntax to look through a string and used a cluster of if statements to find notes in four different formats. These formats where: (note letter), (note letter, octave), (note letter, accidental sign), and (note letter, accidental sign, octave). Note letter, accidental sign, and octave could only be one character.

I used a refined hasCorrectSyntax titled isPlayableSong which did the same thing as listed above except it made sure note letter and octave were playable on the online keyboard.

I checked to make sure octave was legal using a function isLegalDigit which checked if the digit in question was 2-5

I checked to make sure note letter was legal using a function isLegalAlpha which checked if the alpha character was A-G

extractNote returns a string three characters long of a note that is always in the form (“noteLetter, accidental sign, octave”) if there is no accidental sign or octave number there is a space character in those spots.

encodeSong

if playable

set instructions = translated string

else if not playable and has correct syntax

find the beat with a bad note

else

do nothing except return 1

c. hasCorrectSyntax testing

"D5//Z/" should fail because of incorrect note letter

"C/C/G/G/A/A/G/" syntactically correct song, just make sure program works for an easy string

"G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/F#5/B3G5//G//G/" long syntactically correct song but make sure program works for strings of all sizies

"C/C/G/G/A/A/G" missing final beat marker, should fail because of this

"G/" single beat, make sure program works for strings of all sizes

"///" several beats but with but has no notes so program could have problems

"" an empty string, should be correct

" " a space character, not correct

encodeSong testing

"D5//D/" normal song string, should return 0 and change instructions

"D5//Z/" not syntactically correct so should return 1 and change nothing

“D5//D8/" syntactically correct but not playable. Should return 2 and find the number of the beat with the problem

“/G3B3DD5/G/A/A9/” making sure that the program correctly finds the beat number 5 as having the error. It might incorrectly do this because there is an empty beat as the first beat.

"G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/F#5/B3G5//G//G/" make sure program can correctly translate a long string that is playable

"///" make sure the program handles empty beats correctly

"" make sure the program handles empty string correctly