1. I had a fair bit of difficulty interpreting the project spec simply in terms of picking out the important main parts from the trivial details that wouldn’t be important until much later in the process. At my discussion however, the TA suggested printing it out on paper and Highlighting the important parts. While I didn’t literally print it out, I did copy it into a word document and highlighted sections in different colors that were related in some way to each other (main parts, syntax definitions, suggestions, certain special cases that needed to be watched out for, etc.). I also had trouble starting the program from the beginning and keeping it organized enough to come back to it after a day or so without being completely lost, but I read the FAQ and all the sub articles which got me rolling.
2. hasCorrectSyntax

…

Empty is true

Last char not slash is false

Check each char

If any individual char is not valid, false

Make sure each note is in letter, accidental, octave format

…

isPlayableSong

…

Check for correct syntax

Check each char

For each note letter, accidental, and octave, store into variables

When the next char is a slash, check the note for playability

If unable to play, return false

…

encodeSong

…

If the song is playable,

Check each char

Two slashes in a row is a space

If char isn’t a slash,

Store note, accidental, or octave

If next char is a slash,

Add note to instructions

If next char is a note,

Add note to bracket note for later use

…

My program uses hasCorrectSyntax to clarify isPlayableSong. encodeSong calls isPlayableSong to check song validity before running.

1. All of the following cases were tested using “assert” in the main function of my code

*The following are given on the spec sheet (return 0)*

“empty”

//

C/C/G/G/A/A/G/

D3/F#3/A3/D4//D3F#3A3D4/

G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/F#5/B3G5//G//G/

DADDA/

*The following will test variations in capitalizations, ending with slashes, and improper note characters (return 1 if bad syntax or 0 if good)*

D5//Z/ bad

D5//D8////////L/ bad

DADDa/ bad

DADDA bad

//////////////A/ good

DAZZA/ bad

DAZZA bad

D/A/D/D/A/ good

dADDA/ bad

*The following will test the badbeat output (return 2)*

D5//D8/ 🡺 3

/F#3Ab2D0/ 🡺 2

F#3Ab2D4///Gb3// / 🡺 6

/F#3Ab2D4///Gb3//Fb1/ 🡺 7

A/B/C/D/E/F/G//C#8/ 🡺 9

A/B1/C/D/E/F/G//C#8/ 🡺 2

*The following will check bracketed notes and spaces between them (They should all encode properly) (return 0)*

D5//D/

D5//D////F#4Bb5/

/ABCDEFG//A#B#C#D#E#F#G#//Ab5/

Fb2G5D3/A3/Db4/

*The following will test double accidentals and double octaves as well as out of place accidentals and octaves (return 1)*A##/

Abb/

A33/

A99/

#A3/

bA3/

3A/