Some notable obstacles I overcame in this project were the insertion of brackets. I overcame this by using a running total of two strings rather than one, one to store the beat and one storing the total. Another obstacle I overcame were corner cases. My program would often work for regular cases but when I input something like an empty string, it was unable to handle it. I handled these cases separately from the rest of my program.

For the first function, I looped through the string checking whether or not the character itself is valid and if so, checking whether or not the characters before it in combination with it are valid. I checked the beginning and end separately because my loop does not work on these corner cases.

For the second function, I used the first function to check if it was a well-formed tune. If so, I would again loop through the string. I would do a check at the beginning of every possible new beat(a note letter or a slash) and check if the previous beat was playable. Again I checked the beginning and end separately because my code does not deal with corner cases. I checked that if the translated tune is a space, I would change badBeat to the beat in which translateNote returned a space to me.

Since my second function incorporates my first function, all my tests were done on my second function

List of tests:

“” empty string check

“F#/D5/” simple translation check

“B3C6A/Db4/F#6/” brackets check

“A4/Bb/C5/B#3/C9/” bad beat check

“Cb/F#A6G/C5#/” not well formed check

“F#5” not well formed check

“//” double slash check

“H5/” not well formed check

“G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/F#5/B3G5//G//G/” translation check

“D3/F#3/A3/D4//D3F#3A3D4/” translation check