**Notable Obstacles I Overcame:**

Among one of the most notable obstacles I overcame, was the process of creating a function to test if the tune in question was well formed. In order to do this, I had to create a while loop that parsed through each of the characters of the tune to make sure that they were syntactically correct. This entailed not only checking for letters(A-G), accidentals, and numbers, but also checking whether or not they were placed correctly. For example a number and an accidental cannot stand-alone or start off the tune. By far the most difficult challenge I overcame was the process of inserting the brackets around the characters in the translation when several notes were contained in a single beat. I solved this by using the “insert” function and by having a counter that kept track of how many notes there were in each beat of the translation. If there was more than one note in the beat, I inserted a bracket before the note at the beginning of the beat and after the note at the end of the beat.

**Design Description and Pseudocode:**

My program is designed by using multiple functions other than main. The first function is *isLetter*. *isLetter* returns a Boolean value of true or false. It tests whether or not a character is an uppercase letter A through G.

Pseudocode:

Function *isLetter* takes a character and returns a bool.

Checks if the character is uppercase A through G

If it is return true, if not return false

My next function was called *isTuneWellFormed.* This function checks to make sure that the tune is completely syntactically correct, but not necessarily playable.

Pseudocode:

Function takes a string and returns a bool.

if tune is a blank string return true

if the last character in tune is not a slash return false

visit each character in string

if character is a slash increment appropriately

but if character is a letter A-G then check

if its followed by an accidental

and then by a number

Increment appropriately

Otherwise followed by just a number

Increment appropriately

If neither increment by 1

If the first char is not a letter or a slash return false

end when finished checking

My third function was called *translateTune* and it takes 2 strings and an integer and returns an integer. In this function, I appended the translated notes from Professor Smallberg’s function onto a blank string called translation, while simultaneously making sure to add brackets when more than one note was in a beat.

Pseudocode:

Keep a count of translated notes in a single beat and number of slashes

If there is a slash

and more than one note in a beat add brackets by using insert

reset translation counter

increment appropriately

otherwise if there is a letter followed by an accidental and a number

append the translation of tune onto *string translation*

check for badBeat

increment appropriately

otherwise if the letter is just followed by an accidental

translate tune

append

check for badbeat

increment appropriately

increment translation counter

otherwise if it is just a letter

translate tune

append

check for badBeat

increment translation counter

increment appropriately

all other cases…

translate

append

check for badBeat

increment translation counter

increment appropriately

outside loop set instructions equal to translation, return 0

The final function is Professor Smallberg’s *translateNote* function.

**Test Cases:**

if the tune starts with a slash: (/AB/)

purpose: to make sure code translates correctly

if the tune does not end with a slash: (/AB)

purpose: to make sure it fails to translate and returns appropriate value

if two slashes in a row: (AB//C/) or (//AB/C/)

purpose: to make sure the spacing in the translation is correct.

if three slashes in a row: (AB///C/)

purpose: to check the spacing of the translation

if letter outside range of A-G: (VB/A/)

purpose: to make sure it fails to translate and returns appropriate value

if lowercase letter: (aB/F/)

purpose: to make sure it fails to translate and returns appropriate value

if number outside 2-5 keeping exceptions in mind: (D1/E/) or (D6/E/)

purpose: to make sure it fails to translate and returns appropriate value. Also, to check if the badBeat value is correct.

if number starts off tune: (3D/A/)

purpose: to make sure it fails to translate and returns appropriate value

if accidental starts off tune: (#F/AB/)

purpose: to make sure it fails to translate and returns appropriate value

if two numbers in a row: (A33/F/)

purpose: to make sure it fails to translate and returns appropriate value

if two accidentals in a row: (A#bB/)

purpose: to make sure it fails to translate and returns appropriate value

if stand alone accidental or number: (A/b/) or (A/#/)

purpose: to make sure it fails to translate and returns appropriate value

if empty tune: ()

purpose: to make sure it provides blank instructions

if tune just contains space character: ( )

purpose: make sure it does not translate

if multiple notes in one beat: (A/BCA/)

purpose: to make sure brackets are added around chords

Borderline octaves: (A2/ B5/)

Purpose: make sure these borderline values are considered playable

Borderline letters: (AG/)

Purpose: make sure these borderline values are considered playable

If number before accidental: (A3#/)

Purpose: make sure it does not translate and produces appropriate return value

Number that does not follow a letter but is before a letter: (A/3B/)

Purpose: to make sure it does not translate and produces appropriate return value

Accidental before letter (A/#B/) also try (A/A#B/) to see difference; one translates one doesn’t

Purpose: to make sure it does not translate and produces appropriate return value in the first example but does translate in the second example

Ending tune with more than one slash: (A/B//)

Purpose: to make sure spaces are properly handled and placed at end of translation

If tune contains a character besides a letter, accidental or number: (A\*B/)

Purpose: To make sure it does not translate and produces appropriate return value

If space in string: (A B/)

Purpose: To make sure it does not translate and produces appropriate return value

If space after final slash in string (A/B/ )

Purpose: to make sure it does not translate and produces appropriate return value

If tune contains multiple chords: (AB/ CDD/BB/)

Purpose: to make sure brackets are added properly to each chord

If tune contains multiple single note beats: (A/B/C/D/)

Purpose: to make sure the translation is correct and without brackets

**Also tests mentioned in the spec:**

notes with no digit character. These are considered to be in octave 4, so D and D4 are equivalent, as are F# and F#4. (A/B/C/)

notes with a digit character 2, 3, 4, or 5, except that Cb2 is not a playable note. These notes are considered to be in the indicated octave number. (A/Cb2/)

Cb6, representing Cb in the sixth octave, equivalent to B5. (A/Cb6/)

C6, representing C in the sixth octave. (A/C6/)

B#1, representing B# in the first octave, equivalent to C2. (A/B#1/)