CS 31 Project 3 – Report - Jahan Kuruvilla Cherian 104436427

1. The biggest obstacle I faced in this project was figuring out how to add square brackets around multiple notes to indicate a chord. To overcome this I basically checked for if there was a beat (‘/’) and the number of notes (a counter for the number of notes) was greater than 1 then you append a square bracket around the end, but to add it in front of the section string without adding to the entire string, I used the .insert() for the string between the relevant string (got by doing the size of the string – (the number of notes +1) to get the position before the start of the chord.
2. The design of my code follows:

***Function to translate and individual note***

***If the octave ever goes above 9 then print an error message to cerr***

***If the letters are A-G then convert them to their respective buttonbass counterparts***

***If it is an accidental sign then convert it to its respective buttonbass counterpart***

***If it’s a space then just do nothing***

***Go through the string of the respective characters on the buttonbass keyboard in order and pick out the relevant character based on the octave, letter note and accidental sign***

***Function to check if the tune is well formed***

***Loop through the whole tune***

***Check for uppercase letters from A-G***

***If this is true then check for accidental and octaves between 0-9 but if this is not true then it is not well formed.***

***If there are any other characters other than A-G (upper case) and # or b or / and digits, then it is not well formed.***

***If the string is empty then it is well formed***

***If there is at least 1 beat and note then it is well formed***

***If the tune is just beats then it is well formed***

***If the tune does not end with a beat then it is not well formed***

***If the tune contains a space, then it is not well formed***

***Function to check if the tune is playable***

***If the tune is well formed***

***Loop through the whole tune***

***If the octaves are not 2,3,4 or 5 then it is not playable***

***If the tune contains Cb2 then it is not playable***

***If the tune contains B#1 convert to C2 and then it is playable***

***If the tune contains Cb6 convert to B5 and then it is playable If the tune contains C6 convert to B#5 and then it is playable***

***If it isn’t well formed then it is not playable.***

***Function to translate the tune***

***If the tune is playable***

***Intialise the translation string to empty***

***If the tune starts with just a beat then add a space character before the start of the translation for n number of n slashes.***

***Loop through the tune***

***If a letter comes up***

***For the case of letter with accidental translate and add to the translated string***

***For the case of letter with accidental and octave, translate and add to translated string***

***For the case of letter with an octave translate and add to the translated string***

***For the case of just a letter note translate and add to the translated string***

***If the tune contains a beat and more than one note, then add square brackets around the translation for that set of notes.***

***If the tune contains more than one / character in sequence, translate as spaces equal to n-1 for n / characters.***

***Return 2 as defined in the specification***

***If the tune is not well formed***

***Return 1 as defined in the specification***

***If the tune is well formed but not playable***

***Loop through the tune***

***If we find invalid octaves or Cb2***

***Find the beat at which this occurs***

***Return 0 as defined in the specification***

***Function to test the translation***

***If the result is 2***

***Print out that the tune is well formed and playable and print out the translated string***

***If the result is 1***

***Print out that the tune is not well formed***

***If the result is 0***

***Print out that the tune is well formed but not playable and print out the position of the badbeat.***

***Otherwise***

***Print an error message that should never be reached.***

***Main function***

***For an infinite loop***

***Test the translation***

1. Test cases:
2. String of only numbers such as 2345678/ to test if the well formed function works to disregard digits without letters.
3. String of letters that aren’t notes such as XLO/ to test if the well formed function works to disregard letters not in A-G.
4. String of valid notes but with lower case letters such as a#3/bb5//d/ to make sure the well-formed function is false for lower case incidents.
5. Tune but with random letters thrown in the middle of the string such as G3B3DD5//G/A/A3B/C5/B3D5//G//G//CE5//C5/D5/E5/ASFRVE//F#5/B3G5//G//G/ so that well formed returns false because the string has an non valid letter.
6. Tune but with an octave higher than 9 to make sure the well formed function returns it false such as A10/ or Bb34E56/
7. Tune with randomly scattered characters within the tune such as Eb5/F#4//.;[]?!/ to make sure the well formed function returns false for non valid characters.
8. Tune with well-formed notes but in octaves above 5 or below 2 such as A2/ or G7/ to make sure well formed returns true, and we get not playable and get the correct badbeat position.
9. Well formed and playable notes but all alone so as to check that the translation is valid as defined by the buttonbass software, such as E#5/Fb4//D/G/
10. Well formed and playable notes but with multiple notes together so as to check that the translation is valid as defined by the buttonbass software and that the square brackets are put in the correct places, such as E#5/Fb4ABb5G2E#3//D/G/
11. An empty string so as to make sure both playable and well formed functions return true and the translation is just a space.
12. A string with only beats (‘/’) to make both playable and well formed functions return true and the translation is the relevant number of spaces.
13. A tune that does not end with a beat, to make sure the beat is not well formed and so it doesn’t get translated such as Ab3/C/G
14. The 4 special cases wherein the function is well formed but with Cb2 is not playable and with B#1, C6 and Cb6 they are converted to their respective counterparts and are playable and therefore correctly translated.
15. Try tunes with multiple accidental characters in the same note, such as F##/ or G3#b/ to make sure the well formed function.
16. Start of with a forward slash and then go on to start with multiple slashes to make sure that n number of spaces are created for an n number of slashes to begin with. For example: /G/ would produce ‘space’’translation’ and ///A/ translates to ‘space’ ‘space’ ‘space’ ’translation’