**Project 3 Report**

Notable obstacles faced:

1. Checking for all the conditions required to validate the correctness of the song string syntax was a hurdle I had to overcome. Ensuring that there was no undefined behavior in my hasCorrectSyntax function was another obstacle I had to overcome.
2. Learning how to use the integer values for character types in comparisons was another obstacle I faced.
3. Extracting every note, accidental sign and octave digit from the song string was another obstacle I faced.
4. Counting the number of notes were present in one beat in order to determine how the notes should be encoded was a major obstacle I faced.

Design of my program:

Function hasCorrectSyntax:

An empty song string is syntactically valid

Repeatedly:

Check each character for invalid notes, octave digits and special characters that are not accidental signs

Check for incorrect positioning of octave digits and accidental signs. Accidental signs cannot follow digits.

Check for repeated octave digits and accidental signs

Check for spaces within song string (spaces are considered invalid syntax)

Not Repeated: Check if the string is terminated by a slash. If it is not, the song string does not have correct syntax.

Function encodeNote:

Assign the note an integer value that is incremented or decremented based on the accidental sign it is associated with

Assign the note a sequence number that refers to the specific encoded character required by the ButtonBass Software

Return the encoded character

Function isPlayable:

Check if the song string is syntactically valid. Syntactically invalid song strings are not playable.

Repeatedly:

Keep track of which beat is currently being checked

Store the notes, accidental signs and octave digits to check if they are playable

Notes below the second octave and above the sixth octave are not playable apart from certain exceptions (eg. Cb6, B#1)

Check if a note in the second octave is Cb2 as Cb2 is an unplayable note

Check for these notes and return the beat where these unplayable notes are present

Function encodeSong:

Check if the song string has a valid syntax. Syntactically invalid song strings cannot be encoded.

Check if the song string has a valid syntax but is not playable by the ButtonBass software. These song strings cannot be encoded.

If the song string is playable,

Repeatedly,

Count the number of notes in each beat

If there are no notes in a beat, a space is encoded

If there is a single note in a beat, the note, octave digit and accidental sign are stored and the note is encoded using the encodeNote function

If there is more than one note in a beat, the encoded notes (Using the encodeNote function) are placed within square brackets

Main function:

Test the functions hasCorrectSyntax and encodeSong for various song strings

Test Cases:

1. “” (Empty song string is valid)
2. /// (Beats with no notes are valid song strings)
3. A3B#4/D5Cb2/F5G3/ (Cb2 is an unplayable note)
4. A4B5/D7G6/C3/ (Octaves 6 and 7 are unplayable)
5. G/ (Single note with no specified octave)
6. ???? (Invalid characters)
7. A5/B3c4/ (Lower case notes are not valid syntax)
8. A4/B33D2/ (Repeated digits are not valid syntax)
9. B3/C#b4D/ (Repeated accidental signs are not valid syntax)
10. B3/C##4D/ (Repeated accidental signs are not valid syntax)
11. B3/Cbb4D/ (Repeated accidental signs are not valid syntax)
12. D3/F#3/A3/D4//D3F#3A3D4/ (Multiple notes in a beat are encoded in square brackets)
13. A3B/C (Song strings that do not end with a slash are not invalid)
14. B5C3//D/ (An empty beat is encoded as a single space character)
15. A3Bb5/D4Cb6/G/ (Cb6 is playable even though it is in the 6th octave)
16. F3G#5/B#1A3/ (B#1 is playable even though it is in the 1st octave)
17. G5A3/C6D2/ (C6 is playable even though it in the 6th octave)
18. #/ (A single accidental sign is not a valid song string)
19. 3/ (A single digit is not a valid song string)
20. A1B5/G3/// (Octaves below 2 are unplayable)
21. A B/ (Spaces between notes are not valid syntax)
22. b/ (A single accidental sign is not a valid song string)