**Project 3 Report**

1. A brief description of notable obstacles you overcame.

In the first function : isTuneWellFormed(string tune). I overcomplicated things and wrote a isBetween0and9(int i) function for some reason. This function was unnecessary and did not work as it was supposed to. It didn’t work because when I called the function I was actually passing a char such as ‘1’ or ‘8’. Therefore the function always returned false when called within isTuneWellFormed(string tune) but returned true when separately. This was fixed by just using the isdigit() function that takes care of everything. In the second function : int translateTune(string tune, string& instructions, int& badBeat). I fell for the common CS31 student error. My octave was changing to the ASCII code of the number instead of the number. This was fixed by subtracting ‘0’ which is 48.

b) A description of the design of your program. You should use [pseudocode](http://cs.ucla.edu/classes/fall14/cs31/pseudocode.html) in this description where it clarifies the presentation

2 extra functions were written :

1. bool isBetweenAandG(char c) - checks if the char is A or B or C or D or E or F or G
2. char oldKeyToTranslatedNote(string key) - takes a key string like “C#3” or “Bb5” and converts it to a translated note that button bass can play. Uses the given translateNoteFunction. The function works by taking the length of the string and assigning the variables octave, note letter and accidental sign accordingly. For example if the length is 3. Then, the first character is the note, the second is the accidental and the third is the octave. And so on…

*isTuneWellFormed(string tune)*

…

repeatedly look through every new note

if character is a note

set hold to 0

if character after note is not a forward slash and not a note

if character after note is a accidental or octave or digit

set hold to 1

else

return false

if character after note is accidental and character after that is octave

if character after accidental is not a note and not a forward slash

if character is a number

set hold to 2

else

return false

else if character is a forward slash

set hold to zero

else

return false

return true

…

*translateTune(string tune, string& instructions, int& badBeat)*

*…*

if tune is not well formed

return 1

else

repeatedly go through every beat

check for soundless beat

set beat to empty string

repeatedly go through every note in beat

set beat to all those notes

increment current beat

repeatedly go through every note in beat

if note is between A and G

increment number of keys

if number of keys is equal to 1

convert key to translated note

if note is playable

append instructions with translated note

else

return 2

set bad beat to current beat

else if number of keys is greater than 1 (or chord)

repeatedly pick up each key

convert key to translated note

if note is playable

append chord with translated note

else

return 2

set bad beat to current beat

append instruction with chord

else (empty beat)

append instructions with a space character

return 0

*…*

| Input Tune | Return value of translateTune | Instructions (“”) or badBeat | Reason |
| --- | --- | --- | --- |
|  |  |  |  |
| “” | 0 | “” | Empty input |
| “ ” | 1 |  | Illegal character |
| “ /” | 1 |  | Illegal character + slash denoting end of beat |
| “K/” | 1 |  | Notes not between A and G |
| “/” | 0 | “ ” | 1 silent beat |
| “//” | 0 | “ ” | 2 silent beats |
| “G” | 1 |  | Note without end slash |
| “G/" | 0 | “L” | Note |
| “A/A/A” | 1 |  | Multiple beats without end slash |
| “A/A/A/” | 0 | “QQQ” | Multiple beats |
| "Db" | 1 |  | Note with accidental without end slash |
| “Db/” | 0 | “!” | Note with accidental with end slash |
| “G#2" | 1 |  | Note with accidental and octave without end slash |
| “G#2/” | 0 | “4” | Note with accidental and octave with end slash |
| “Bd4/” | 1 |  | Using incorrect accidentals |
| “F%/“ | 1 |  | Using incorrect accidentals |
| “G92/” | 1 |  | Using consecutive numbers |
| “G#2/A4/Ab4/” | 0 | “40$” | 3 beats |
| “C0/" | 2 | badBeat = 1 | Note not playable at beat 1 |
| “Gb4/A4/Cb2/” | 2 | badBeat = 3 | Note not playable at beat 3 |
| “GA5C#6/” | 2 | badBeat = 1 | Bad Chord when one note is unplayable |
| “GABb” | 0 | “[LQ%]” | Playable Chord |
| “GABb/F#AB/” | 0 | “[LQ%][#QW]” | 2 Playable Chord |
| “FGA/G#2/Fb/” | 0 | “[KLQ]4J” | Chord + 2 Notes |
| “GAE//E#/” | 0 | “[LQJ] K” | Chord then silent beat then single note beat |