Report

1. Notable obstacles:

I initially tried to use for loop incrementation to determine if the input was translatable. Eventually I realized using substrings would be much easier. Another obstacle was finding a way to convert a string of number into an integer.

I didn’t have time to implement the return integers for badBeat because my inttranslatesong depends on the Boolean function istranslatable which contained the cases that would return integers.

1. Pseudocode:

hasCorrectSyntax:

for (current character less than song size)

if (it is not a valid character)

return false

while (substring of song is not of length zero)

if (color is not terminated)

return false

else if (part of the song doesn’t terminate a sustained note properly)

return false

else if (current character is a rest)

evaluate the rest of the string

else if (a color and termination are given)

evaluate the rest of the code

else if (part of the code is a double digit sustained note)

if (the following character is a slash)

evaluate the rest of the string

else return false

else if (part of the code is a single digit sustained note)

if (the following character is a slash)

evaluate the rest of the string

else return false

else return true

isTranslatable:

for (current character)

if (it’s not a valid character)

return false

while (substring is not of length zero)

if (color is not terminated)

return false

else if (a sustained note is not terminated)

return false

else if (current character is a rest)

evaluate the rest of the song/string

else if (current character is a terminated non-sustained color)

evaluate the rest of the song/string

else if (double digit sustained note)

find the number of following slashes

convert the two digits to an integer

if (the number of slashes is greater than or equal to the 2-digit int)

evaluate the rest of the song

else return false

else if (single digit sustained note)

find the number of slashes the follow

convert the digit to an int

if (the number of slashes is greater than or equal to the digit int)

evaluate the rest of the code

else return false

else return true

isCorrectChar:

if (the char is one of the valid characters)

return true

else

return false

translateSong:

if (the song is translatable)

while(the substring length is greater than zero)

if (there’s a rest)

add an ‘x’ to instructions

evaluate the rest of the song

else if (there’s a non-sustained color/note)

add that color to instructions

evaluate the rest of the song

else if (double digit sustained note)

for (current character in range of possible slashes)

if (it’s a slash)

add one to numSlashes

if (numSlashes hasn’t reached it’s digit limit)

add the uppercase letter to instructions

else

evaluate the rest of the song

else if (single digit sustained note)

(same as double digit case)

else return 1

1. Test data:

r/ to verify hasCorrectSyntax

g3// to verify that it is syntactically correct but not translatable.

g03/// to verify double digit code

g3/// to verify single digit code

g3///// to verify substring is working properly

g3// should return 3 but since I