Alex Wasdahl

Project 3 Report

CS31 Spring 2022

Brief description of notable obstacles:

The biggest obstacles I found were as follows

* Parsing the requirements in general -- required a great deal of logical planning
* Keeping track of the proper beat count given the different types of beats
* Figuring out, in the convertTune function, how to combine the approaches of 1. Figuring out whether a string is convertible 2. Actually converting it to the instructions
* Checking within tune strings for various beat types in order to convert them into proper instructions

Description of the design of your program

* Program is designed along the following steps

Define preliminary functions isColor, makeLowercase, makeUppercase, toInt, toTwoDigitInt, and countChars

Define function hasProperSyntax

Returns true if tune is empty string

Otherwise if last char in string is letter or digit return false

Otherwise if first char is digit return false

Repeatedly

for each character in the tune parameter

if the character at index i is not a letter, digit, or '/' character return false

otherwise if char at i is a letter

if the letter does not correspond to a color, return false

if the character following i is a letter, return false

otherwise if char at i is a digit

if digit is followed by two more digits, return false

otherwise if the previous character is '/', return false

Otherwise, return true

Define function convertTune

If the tune does not have proper syntax, leave badBeat unchanged and return 1

Repeatedly

For each character in the tune parameter

if the char in tune is a /, increment numofBeats

if the char in tune is a digit

check to see if the following character is a digit

If so, convert the two characters to a two digit integer  
 otherwise, convert the first character into a one digit int

If the note is set to sustain for less than 2

set badBeat to beat number and return 3

If the tune is unfinished

Set badBeat to the number of chars in the tune + 1

Otherwise repeatedly

For every character in the length of the sustain note

If it is a /

then increment the number of beats in the sustain note

Otherwise

Set badBeat to the number of beats plus one and return 2

Reset the number of beats in the sustained note to zero

Otherwise continue

Now the syntax has been checked to be able to convert

If the first character is a /

Repeteadly

For each character in the tune

If it is not a / then break

Otherwise add x to the instructions

Repeatedly

For each character in the tune

If it is a color followed by a /, then add a lowercase letter for that color to instructions

if the char in tune is a digit

check to see if the following character is a digit

If so, convert the two characters to a two digit integer

Repeatedly

For each sustained note

Add uppercase letter for color to instructions

Oterhwise

convert the first character into a one digit int

repeatedly

For each sustained note

Add uppercase letter for color to instructions

Repeatedly

For each additional character beyond the sustained note

If the character is not a /

Break

Otherwise add x to instructions

Return 0

Call int main() {}

List of test data

1. True cases for hasProperSyntax
   1. ////////
      1. Tests all slashes
   2. “”
      1. Tests empty string
   3. G/
      1. Testing a capital letter color followed by /
   4. o5/
      1. Tests syntactically correct but not convertible string
2. False cases for hasProperSyntax
   1. b/r111/
      1. Tests three consecutive integers
   2. Qwertyuiop[asdfghjklzxcvbnm,
      1. Testing invalid characters
3. Cases for convertTune returning 0 (convertible)
   1. b04////o11///////////
      1. Testing various sustained notes of different lengths
      2. badBeat unchanged
      3. Instructions should be BBBBOOOOOOOOOOO
   2. /////
      1. Testing beat of only slashes
      2. Leaves badbeat unchanged
      3. Instructions should be xxxxx
4. Non-convertible cases for convertTune
   1. Returning 1 (not syntactically correct)
      1. g/o653
         1. Bad beat and instruction unchanged
         2. Has proper Syntax returns false
   2. Returning 2 (non slash beat present during sustain
      1. r3//y/b0//g2/
         1. badBeat = 3
         2. Has two errors– tests if program can output leftmost error
   3. Returning 3 (tune specifies sustain less than 2)
      1. y01///g///o/
         1. badBeat = 1
         2. Holding beat specifies length <2
   4. Returning 4 (tune ends prematurely)
      1. ///b17///
         1. badBeat = 7 (1 more than # of beats)
         2. Tune ends before length of beat held