Adil Sultan

804275869

Project 3 Report

**1)** I didn’t for the most part didn’t have very many obstacles. I did struggle with how I could append the appropriate letter of directions to the instructions string during a freeze but after reading the stuff for this project I was able to figure it out. I ran out of time so I was not able to address a freeze that has a double digit duration. Another thing I ran out of time to address was if a beat ends prematurely to return and if there are two things wrong to return for the left most error.

**2)** This is how I designed my code:

hasCorrect syntax bool hasCorrectSyntax(string dance){

if (empty string)

return true;

//here for any case that is not acceptable return false.

for (int k = 0; k != dance.size(); k ++){

//if any letters other than the acceptable kind are use then return false

if (//if any letters other than the acceptable kind are use then return false)

return false;

// if a letter is not followed by a slash then return false

if (// if a letter is not followed by a slash then return false)

return false;

//if a digit in the string is not followed by either a digit or a letter then return false

if (//if a digit in the string is not followed by either a digit or a letter then return false)

return false;

}

//if the last character in the string is not a slash then return false

if (//if the last character in the string is not a slash then return false)

return false;

}

//if none of the preceding cases occur then the string has correct syntax so return true

return true;

}

int convertDance (string dance, string& instructions, string& badBeat){

int beatDuration;

char longBeat = '0';

int beatCount = 1;

// if the string doesn't have correct syntax then don't convert and return 1;

return 1;

if (hasCorrectSyntax(dance)){

int i;

//traverse the string

for (//traverse the string){

if you come across a slash that indicates the end of a beat, increment beat counter;

beatCount++;

//if the index of dance string is a lower case letter append it to the instructions string

instructions += dance[i];

// if the index is a slash and it followed by anything other than a slash then continue

continue;

// if the slash is followed by a slash then what we have is an empty beat append a '.' to represent this

instructions += '.';

// if the index is an uppercase letter then convert it to lower case and append

instructions += tolower(dance[i]);

//if the beat number is not greater than 2 then bad beat is equal to the number of that beat and return 3

if (isdigit(dance[i]) && isalpha(dance[i+1])){

if (dance[i] < '2'){

badBeat = beatCount;

return 3;

}

}

// if a freeze is in effect and ends prematurely then return 4 and set badBeat to that beat

int freezeCount = 0;

longBeat = dance[i];

beatDuration = longBeat - '1';

beatDuration++;

for (traverse the string from here{

if (dance[x]=='/')

freezeCount++;

}

if (beatDuration != freezeCount){

badBeat = beatCount;

return 4;

}

}

// if the index is a digit and it followed by a letter then append that letter the number equal to the number of the digit

{

longBeat = dance[i];

beatDuration = longBeat - '1';

beatDuration++;

for ( int x = 0; x < beatDuration; x++){

instructions += tolower(dance[i+1]);

}

}

}

}

return 0;

}

**3)**

List of test cases my project should be able to process correctly:

d/u//

d/u

d/3R///

d/u//r

d/z//

d/R/u/0d/

List of test cases it could potentially not be able to handle:

03l///u/r/ (**double digits for a freeze)**

3r//d/u/l/ (**Premature end to a freeze)**