Li-Wei Tseng

304289977

Report

1. obstacles

One of the most notable obstacle is that the number in the string is regarded as a character when it is located. I use motion[i] – ‘0’ to convert the character to integer.

Since there can be two digits in the motion, I need to set up the cases when there is only one digit and when there is two digits. For the two digits part, I multiply the first digit by ten and add it to the second one, thus forming a base 10 number.

b. pseudocode:

1. bool isMotionMeaningful(string motion)

//determine if the string is meaningful.

repeatledly…

if the character of the string is ‘/’,

increment a variable add by 1.

or if the character of the string is alphabet,

check if it is ‘a’, ‘A’, ‘s’, ‘S’, ‘w’, ‘W’, ‘d’, or ‘D’ only which are accepted

if it is, check if the next character is ‘/’,

if it is, increment the variable add by 2.

if it is not, return false.

if the alphabet is not the accepted letter,

return false.

or if the character of the string is a digit,

check if its next character is also a digit,

if so, check if the character next to the two digits is accepted letter

if so, check if the character next to the letter is ‘/’

f so, increment the variable add by 4.

if not, return false.

if the letter next to the two digits is not accepted, return false.

or if the next character is not a digit,

check if the character is the accepted letter,

if so, increment the variable add by 2.

all other case, return false.

All other case, return false.

increment the index i with the variable add.

if the string repeatedly passes all the tests above, return true.

1. int translateMotion(string motion, string &instruction, int &badBeat)

//translate the string

pass the string to a function to test if it is translatable,

if so, pass the string to a function to translate it to the instruction,

return 0.

or if it is not translatable, pass it to a function to test if it is meaningful,

if not, return 1.

or if the string is meaningful,

repeatedly,

if the character is a digit, or its next character is also a digit,

run the string and break it to substring,

pass the substring to functions to test it,

if a beat prematurely, return 3 and set badBeat to the beat.

if a beat is not consists of only a slash, return 2 and set badBeat.

if the hold of a beat is less than 2, return 4 and set badBeat.

if the character is ‘/’,

just continue the loop.

1. bool translatable(string motion)

//test if the string is translatable

if the string is not meaningful, and its beat consists not only of a slash and ends prematurely, and the hold of the beat is less than 2, return false.

otherwise, return true.

1. bool isoneSlash(string motion)

//test if a beat consists of only a slash

repeatedly..

if the character and its next are digits,

change the character digits to integer digits,

count the slashes after the digits.

if the character is a digit,

change the character digit to the integer digit,

count the slashes after the digit.

if the number of digit is not equal to the number of slashes, return false.

if the string pass all the exams, return true.

1. bool isendPrematurely(stirng motion)

//test if a beat ends prematurely

repeatedly…

if the character and its next are digits,

change the character digits to integer digits,

count the slashes after digits, if it run out of bound, return false.

if the character is a digit,

change the character digit to the integer digit,

count the slashes after digits, if it run out of bound, return false.

if the string pass all the exams, return true.

1. bool holdLessThan2(string motion)

repeatedly…

if the character and its next are digits,

change the character digits to integer digits,

or if the character is a digit,

change the character digit to the integer digit,

if the number of integer is either 1 or 0, return false.

if the string pass all the exams, return true.

1. int badBeatF(string motion)

//find bad beat

repeatedly…

if the character and its next are digits,

change the character digits to integer digits,

if the number of digit is either 1 or 0, increment count, set the badBeat to count, and ends the program.

if not, count the slashes.

if the count of slashes is not equal to number, increment count, set the badBeat to count, and ends the program.

if the character is a digit,

if the number of digit is either 1 or 0, increment count, set the badBeat to count, and ends the program.

if not, count the slashes.

if the count of slashes is not equal to number, increment count, set the badBeat to count, and ends the program.

return badbeat.

1. string instruction(string motion)

//translate the string

repeatedly…

if the character and its next are digits,

change the character digits to integer digits,

if not, count the slashes.

change the characters following digits to upper case letter.

store it number of times to a temporary string.

or if the character is a digit,

change the character digits to integer digits,

if not, count the slashes.

change the characters following digits to upper case letter.

store it number of times to a temporary string.

or if the character is an accepted alphabet

change the character to lower case letter.

store it to a temporary string.

or if the character is ‘/’

store a ‘.’ to a temporary string

put the temporary string to another one.

return the final string

c. list of data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | test | return | Bad beat | instruction |
| 1 | s\w | 1 |  |  |
| 2 | s/wk | 1 |  |  |
| 3 | w/W/w/s/ | 0 |  | wws |
| 4 | w//s/3d///s/ | 0 |  | w.sDDDs |
| 5 | s/3d//w/ | 2 | 4 |  |
| 6 | s/3d// | 3 | 4 |  |
| 7 | 0d/ | 4 | 1 |  |
| 8 | 3d//0s/ | 2 | 3 |  |
| 9 | 3d//w/0s//2w/ | 2 | 3 |  |

I used 9 cases to test my program. The first two are not meaningful, so return 1. The 3-4 are the one that is translatable, and thus return 0 and set the instruction to the desired results. The 5th one’s beat is not just a slash, so return 2 and the bad beat is the 4th. The 6th one’s beat ends prematurely, so return 3 and the bad beat is the 4th. The 7th one’s hold of beat contains 0, so return 4 and the bad beat is the 1st . The 8-9 are the mixed cases, while the leftmost occurring problem is reported. Thus the program in 8th and 9th return 2 and set the bad beat to 3.