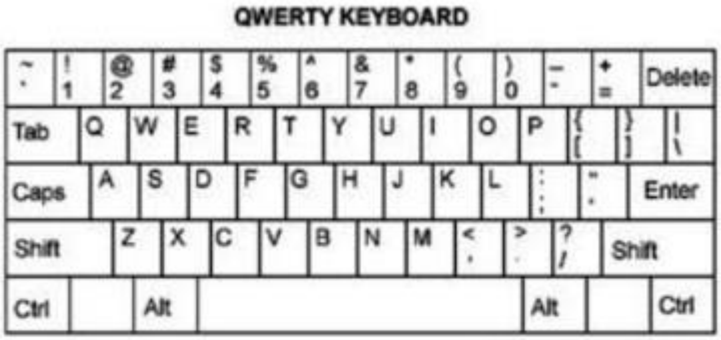
Keyboard Calculations



This problem is only concerned with UPPER CASE letters on a QWERTY keyboard (No spaces or special characters are included). All letters on a QWERTY keyboard are adjacent to other letters. For example, the letter Q is adjacent to both the letter W and the letter A. And the letter D is adjacent to letters E, R, S, F, X, and C.

The following table indicates the adjacent relationship of all letters on a QWERTY keyboard.

Letter	Adjacent letters	Letter	Adjacent letters
A	Q, W, S, Z	N	B, H, J, M
B	V, G, H, N	O	I, K, L, P
C	X, D, F, V	P	O, L
D	E, R, S, F, X, C	Q	A, W
E	W, S, D, R	R	E, D, F, T
F	R, T, D, G, C, V	S	W, E, A, D, Z, X
G	T, Y, F, H, V, B	T	R, F, G, Y
H	Y, U, G, J, B, N	U	Y, H, J, I
I	U, J, K, O	V	C, F, G, B
J	U, I, H, K, N, M	W	Q, A, S, E
K	I, O, J, L, M	X	Z, S, D, C
L	K, O, P	Y	T, G, H, U
M	N, J, K	Z	A, S, X

In this problem you will complete four different methods. The `kbDistance`, `averageDistance`, `numDirectionChanges`, and `wordDifficulty`. `kbDistance` calculates the distance between any two letters on a QWERTY keyboard, `averageDistancee` determines the average distance between consecutive letters in a word, `numDirectionChanges` determines the number of (left – right) direction changes in typing a word on a QWERTY keyboard, and `wordDifficulty` determines the difficulty of entering a word on a QWERTY keyboard.

Keyboard Calculations

In this problem, the distance between two letters is defined as the minimum number of jumps from one letter to another adjacent letter. For example, the distance between E and R is one. The distance between W and C is three. The distance between Q and M is 8.
Note: By definition the distance between the same letter is 0

The following code shows the results of the `kbDistance` method.

The following code	Returns
`KeyBoardCalculations.kbDistance(“A”, “A”)`	0
`KeyBoardCalculations.kbDistance(“E”, “R”)`	1
`KeyBoardCalculations.kbDistance(“W”, “C”)`	3
`KeyBoardCalculations.kbDistance(“Q”, “M”)`	8

The average distance between consecutive letters in the word is found by summing the distance between consecutive letters in word. To find the average distance for the word COMPUTER, sum the following values:

- 6 (distance between C and O)
- 2 (distance between O and M)
- 3 (distance between M and P)
- 3 (distance between P and U)
- 2 (distance between U and T)
- 2 (distance between T and E)
- 1 (distance between E and R)

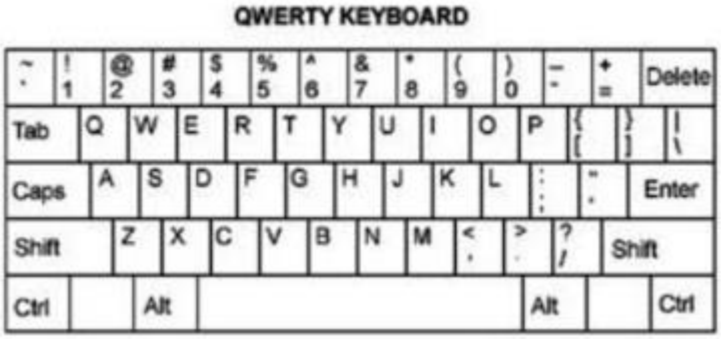
For a total of 19. Divide the total by 7 for an average of 2.714285

The following code shows the results of the `averageDistance` method.

The following code	Returns
`KeyBoardCalculations.averageDistance(“WAS”)`	1.0
`KeyBoardCalculations.averageDistance(“KING”)`	1.666666… = 5 / 3
`KeyBoardCalculations.averageDistance(“SAMPLE”)`	3.8 = 19 / 5
`KeyBoardCalculations.averageDistance(“COMPUTER”)`	2.714285… = 19 / 7
`KeyBoardCalculations.averageDistance(“AVERAGE”)`	3 = 18 / 6
`KeyBoardCalculations. averageDistance(“ALABAMA”)`	6.666666… = 40 / 6

Keyboard Calculations

A word contains a change in direction every time the middle letter of three consecutive letters in the word is farthest to the right or farthest to the left.


QWERTY KEYBOARD
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Delete
Tab Q W E R T Y U I O P { [}] | \
Caps A S D F G H J K L : ; " ' Enter
Shift Z X C V B N M < , > . ? / Shift
Ctrl Alt Alt Ctrl

The following table shows several calls of the `numDirectionChanges` method.

`The following code`	`Returns`
`KeyBoardCalculations.numDirectionChanges("MIK")`	`0`
`KeyBoardCalculations.numDirectionChanges("MJU")`	`0`
`KeyBoardCalculations.numDirectionChanges("WZA")`	`0`
`KeyBoardCalculations.numDirectionChanges("NHY")`	`0`
`KeyBoardCalculations.numDirectionChanges("KING")`	`0`
`KeyBoardCalculations.numDirectionChanges("JUM")`	`1`
`KeyBoardCalculations.numDirectionChanges("WAZ")`	`1`
`KeyBoardCalculations.numDirectionChanges("TFV")`	`1`
`KeyBoardCalculations.numDirectionChanges("WAS")`	`1`
`KeyBoardCalculations.numDirectionChanges("SAMPLE")`	`2`
`KeyBoardCalculations.numDirectionChanges("COMPUTER")`	`4`
`KeyBoardCalculations.numDirectionChanges("AVERAGE")`	`5`
`KeyBoardCalculations.numDirectionChanges("ALABAMA")`	`5`

Keyboard Calculations

There are six levels of difficulty for entering a word on a QWERTY keyboard. The six levels of difficulty are determined by according to the following three properties:

1) The number of letters in the word
2) The average distance between consecutive letters in the word, and
3) The number of changes in (left-right) direction.

The six levels of difficulty are:

Difficulty level	Conditions (all three condition must be satisfied)
elementary	word has less than four letters average distance between consecutive letters is less than 2 word has at most one change in direction
basic	word has less than seven letters average distance between consecutive letters is less than 3 word has at most two changes in direction
so-so	word has less than nine letters average distance between consecutive letters is less than 4 word has at most three changes in direction
average	word has less than ten letters average distance between consecutive letters is less than 5 word has at most four changes in direction
demanding	word has less than ten letters average distance between consecutive letters is less than 6 word has at most five changes in direction
challenging	Otherwise

The following code shows the results of the `wordDifficulty` method.

The following code	Returns
`KeyBoardCalculations.wordDifficulty(“WAS”)`	`elementary`
`KeyBoardCalculations.wordDifficulty(“KING”)`	`basic`
`KeyBoardCalculations.wordDifficulty(“SAMPLE”)`	`so-so`
`KeyBoardCalculations.wordDifficulty(“COMPUTER”)`	`average`
`KeyBoardCalculations.wordDifficulty(“AVERAGE”)`	`demanding`
`KeyBoardCalculations.wordDifficulty(“ALABAMA”)`	`challenging`