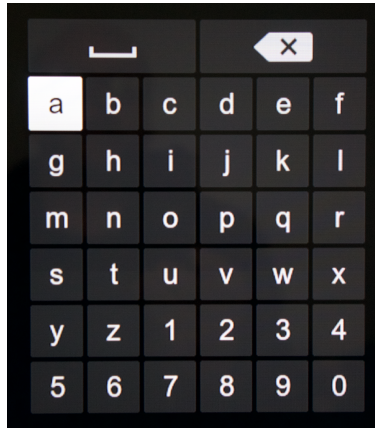# CS161: Introduction to Computer Science
Homework Assignment 10
Due: Monday 4/25 by 11:59pm

Many cable packages come with a search interface that let's you search for a show or movie by typing the name one letter at a time using your remote control. For example, consider the keyboard shown below:

Suppose we want to know if any of the Harry Potter movies are showing. Starting with the cursor at "a" and using the "up", "down", "left", and "right" buttons on our remote control, we would push the following sequence of buttons:

- h ("down", "right", "select")

- a ("up", "left", "select")

- r ("right" 5 times, "down" 2 times, "select", "select")

- y ("down" 2 times, "left" 5 times, "select")

- '␣' ("up" 5 times, "select")

- p ("down" 3 times, "right" 3 times, "select")

- o ("left", "select")

- t ("left", "down", "select", "select")

- e ("up" 3 times, "right" 3 times, "select")

- r ("down" 2 times, "right" , "select")

In total, we had to push 52 buttons just to type the 12-character title "Harry Potter". You can imagine how tiring and tedious this becomes!

One idea to reduce the number of button pushes is to examine different arrangements of the keys on the keyboard. Here are some examples of different arrangements we could try:

a b c d e f g h i j k l m n o p q r s t u v w x y z   0 1 2 3 4 5 6 7 8 9

Here, the keys are all laid out in a single row. Another idea is to arrange the keys using a QWERTY layout:

```
q w e r t y u i o p
a s d f g h j k l z
x c v b n m   0 1 2
3 4 5 6 7 8 9
```

Assuming the cursor started at the upper-left key ("q"), this arrangement would require only 45 buttons to type "Harry Potter". In general, there are lots of different arrangements we could experiment with to find the one that requires (on average) the fewest number of key presses.

In this assignment, you will implement two classes: `Key` and `Keyboard`. The `Key` class represents a key on the keyboard. The `Keyboard` class will use a two-dimensional array to represent the keyboard itself. Your `Keyboard` class should be general enough that it can represent different arrangements. Finally, you will experiment to see which arrangements are the best – i.e., which arrangements require the fewest button pushes.

The starter code for this assignment contains a `Controller` class which you can read about at the end of this writeup.

━━━ The Key class ━━━

Create a new Java class named `Key`. This class should represent a single key on the keyboard. Looking at the pictures above, what state (i.e. instance variables) and what methods should a key have? It's extremely important that you stop and spend some time thinking about this question.

When I thought about this question, I came up with the following: a key can be represented using a symbol, a row index, and a column index. These then should be the instance variables in your `Key` class.

In terms of methods, it seems like a key needs only basic methods such as a constructor, accessors, and a `toString` method. Note that it doesn't make sense for a key to have mutator methods – once the instance variables are initialized in the constructor they shouldn't change.[1]

It would also be useful to have a method that takes in another key and computes the distance between *this* key and the *other* key. As such, you should add the following method to your `Key` class:

$$\text{public int distance(Key other)}$$

The distance between two keys on a keyboard is simply the difference between their rows plus the difference between their columns. For example, suppose we had a key at position (3,2) – i.e. row 3, column 2 – and another key at (5,4) – i.e. row 5, column 4. Then the distance between these two keys is:

$$|3 - 5| + |2 - 4| = 2 + 2 = 4$$

You can use the `abs` method in the Java `Math` class to take the absolute value.

---

[1]You might have thought the `Key` class should have a `push` method. Depending upon the application, this could definitely be useful.

The second Java class is a `Keyboard` class that represents the keyboard itself using a 2-dimensional array. This class should have the following methods:

1. `public Keyboard(String chars, int numColumns)` – A constructor that takes a string containing the sequence of characters on the keyboard, and the number of columns in each row. Look at the `Controller` class to see an example of how the constructor is called.

   The constructor should create and initialize the 2-dimensional array. Given the number of columns and the sequence of characters, you will need to compute the number of rows. You may need to use the `ceil` function in the `Math` class to help you round up.

2. `public int pressesRequired(String text)` – A method that takes a piece of text (e.g. "Harry Potter") and computes the number of key presses required to navigate to and select each character in the string. The cursor should always start at row 0, column 0 on the keyboard. You can ignore any characters in the text that are not in the keyboard.

3. `public String toString()` - A method that returns a string representation of the keyboard.

Feel free to add private methods to help you keep your code organized and your methods short. Below is what my solution outputs when I run the `Controller` class:

```
It took 58 key presses to type
"harry potter" using this key arrangement:
a b c d e f g h i j
k l m n o p q r s t
u v w x y z   0 1 2
3 4 5 6 7 8 9

It took 97 key presses to type
"harry potter" using this key arrangement:
a b c d e f g h i j k l m n o p q r s t u v w x y z   0 1 2 3 4 5 6 7 8 9

It took 45 key presses to type
"harry potter" using this key arrangement:
q w e r t y u i o p
a s d f g h j k l z
x c v b n m   0 1 2
3 4 5 6 7 8 9

It took 87 key presses to type
"harry potter" using this key arrangement:
q w e r t y u i o p a s d f g h j k l z x c v b n m   0 1 2 3 4 5 6 7 8 9


It took 309 key presses to type
"dr strangelove or how i learned to stop worrying and love the bomb" using this key arrangement:
a b c d e f g h i j
k l m n o p q r s t
u v w x y z   0 1 2
3 4 5 6 7 8 9

It took 727 key presses to type
```

"dr strangelove or how i learned to stop worrying and love the bomb" using this key arrangement:
a b c d e f g h i j k l m n o p q r s t u v w x y z   0 1 2 3 4 5 6 7 8 9

It took 331 key presses to type
"dr strangelove or how i learned to stop worrying and love the bomb" using this key arrangement:
q w e r t y u i o p
a s d f g h j k l z
x c v b n m   0 1 2
3 4 5 6 7 8 9

It took 907 key presses to type
"dr strangelove or how i learned to stop worrying and love the bomb" using this key arrangement:
q w e r t y u i o p a s d f g h j k l z x c v b n m   0 1 2 3 4 5 6 7 8 9

It took 224 key presses to type
"the quick brown fox jumped over the lazy dog." using this key arrangement:
a b c d e f g h i j
k l m n o p q r s t
u v w x y z   0 1 2
3 4 5 6 7 8 9

It took 544 key presses to type
"the quick brown fox jumped over the lazy dog." using this key arrangement:
a b c d e f g h i j k l m n o p q r s t u v w x y z   0 1 2 3 4 5 6 7 8 9

It took 247 key presses to type
"the quick brown fox jumped over the lazy dog." using this key arrangement:
q w e r t y u i o p
a s d f g h j k l z
x c v b n m   0 1 2
3 4 5 6 7 8 9

It took 568 key presses to type
"the quick brown fox jumped over the lazy dog." using this key arrangement:
q w e r t y u i o p a s d f g h j k l z x c v b n m   0 1 2 3 4 5 6 7 8 9

The Controller class

Come up with your own keyboard arrangement that takes fewer key presses than `KEYS1` and `KEYS2` on the text `LONG_TEXT`. This variable contains the opening lines of the book "Moby Dick".

You can experiment with the ordering of the characters as well as the number of columns. You should add your keyboard as a new `static` variable called `KEYS3`.

Style Guide

Before you submit your assignment, go through the checklist below and make sure your code conforms to the style guide.

Checklist
- ☐ All unused variables are deleted
- ☐ All instance variables are used in more than one method (if not, make them local)
- ☐ Javadoc comment for all classes
- ☐ All methods have Javadoc comments (except for the `main` method)
- ☐ All numbers have been replaced with constants (i.e. no magic numbers)
- ☐ Proper capitalization of variables, methods, and classes
- ☐ Use white space to separate different sections of your code

Read the "Style Guide" (under "Resources" on the course website) for more information.

Submitting your homework assignment

You should submit your `hw10` folder with the `Key`, `Keyboard`, and `Controller` class inside. Please rename your folder before you zip it and upload to Moodle.