Exercise

Please read this page before proceeding!
The following questions (a) and (b) help you program code that has the following structure:

```java
public class Exercise {
    public static void main(String args[]) {
        // body of main
    } // method main

    public static int[] char_counts(String S) {
        // body of char_counts
    } // method char_counts
} // class Exercise
```

You need to complete the code for the bodies of methods `char_counts` and `main`. If you get stuck on `char_counts`, you may assume `char_counts` has been correctly programmed so can you finish code for `main`.

Hint: When completed, the code must generate the following output:

```
abc
===
131
110
220
112
310
102
032
400
```
Part a)

Fill in the code for method `char_counts`, below. You may assume that method `main` calls `char_counts` from within the body of `main`. The method must count how many times each character `a`, `b`, and `c` appears in `String S`. If the method detects an illegal character, the method must exit the program with an error message, `Wrong input!`, printed to the user. Method `char_counts` returns a 1-D array of integers. This array stores the number of times `a`, `b`, and `c` appear in `S` as the first, second, and third elements of the returned array, respectively. Hint: You might wish to use the methods `charAt(int index)` and `length()` from the `String` class somewhere in `char_counts`.

```java
public static int[] char_counts(String S) {
    // Method body goes here
}
```

} // Method char_counts
Part b)

Fill in the code for method `main`, started below. The initializer list, referenced by `A`, stores arrays of strings. The following code should call `char_counts` to find the character count inside each string element of the initializer list. You must use a 3-D array of integers called `key` to store the returned 1-D array from `char_counts`. Instantiate sizes for `key` only as large as necessary in each dimension. Method `main` must also output each array returned by `char_counts`, as shown on page 1. You must include blank lines as depicted in the output displayed on page 1.

```java
public static void main(String[] args) {
    String A[][] = {
        {"abcbb", "ba"},
        {"abba", "accb"},
        {"aaba"},
        {"cca", "cbcbb", "aaaa"}
    };
    System.out.println("abc");
    System.out.println("===");

    // method main
```