Exercises

1. Discuss the `stringMatch` problem on the next page. What three `String` methods will you need to solve it? (If you have time during the activity, complete the method.)

`String.length` for the loop, and `String.substring` and `String.equals` for the comparison.

2. Discuss the `stringYak` problem on the next page. What two `String` methods will you need to solve it? (If you have time during the activity, complete the method.)

`String.length` for the loop, and `String.charAt` to identify `'y'` and `'k'`.

[CodingBat] Given two strings, return the number of positions where they contain the same substring of length two. So `"xxcaazz"` and `"xxbaaz"` yields 3, since the `"xx"`, `"aa"`, and `"az"` substrings appear in the same place in both strings.

```java
public static int stringMatch(String a, String b) {
    // figure which string is shorter
    int len = Math.min(a.length(), b.length());
    int count = 0;

    // look at both substrings starting at i
    for (int i = 0; i < len - 1; i++) {
        String aSub = a.substring(i, i + 2);
        String bSub = b.substring(i, i + 2);
        if (aSub.equals(bSub)) {
            count++;
        }
    }

    return count;

}
```

[CodingBat] Suppose the string `"yak"` is unlucky. Given a string, return a version where all the `"yak"` are removed, but the `'a'` can be any character. The `"yak"` strings will not overlap.

```
stringYak("yakpak") → "pak"
stringYak("pikyik") → "pik"
stringYak("yak123ya") → "123ya"
```

```java
public static String stringYak(String str) {
    String result = "";

    for (int i = 0; i < str.length(); i++) {
        if (i + 2 < str.length() && str.charAt(i) == 'y'
                                 && str.charAt(i + 2) == 'k') {
            i = i + 2;
        } else {
            result += str.charAt(i);
        }
    }

    return result;

}
```