Indiana University
CSCI-C212: Introduction to Software Systems (F25) Lab 05
Joshua Crotts

Arrays & ArrayLists

What To Do:

Follow each step carefully. As you complete the lab, submit the source files (`.java`) problems to the autograder (link is in the Canvas portal). After finishing, please submit your work to the autograder and let one of the TAs know.

For this lab, please place each method inside its own class file labeled as `ProblemX`, where `X` is the problem number. The accompanying test files should be named `ProblemXTest`.

Problem 1

Design the `int[] countEvenOdds(int[] vals)` method that returns a tuple (an array of two values) where index 0 stores the amount of even values and index 1 stores the amount of odd values.

```
countEvenOdds({11, 9, 2, 3, 7, 10, 12, 114}) => {4, 4}
countEvenOdds({11, 13, 15, 17})              => {0, 4}
```

Problem 2

Warning: *A lot of students find this problem unintuitive. When we say, "take the following test cases as motivation," we mean to actually sit down, work through them, and understand what's going on. Don't just rush through it!*

Design the `List<String> assignGrades(List<Double> G, List<String> L, double[] C)` method that, when given a `List<Double>` of exam grades G, a `List<String>` of letter grades L, and a cut-off score array C, assigns a letter grade to the given exam. That is, the method should return a list where the i^{th} element corresponds to the i^{th} element of G.

You may assume that L`.length` = C`.length` + 1, and that $L[i]$ corresponds to $C[i]$. Importantly, $C[i]$ is the lowest score at which the letter $L[i]$ is awarded. The fact that L`.length` is one greater than C`.length` implies that anything lower than the lowest cutoff is awarded the "lowest" grade. You may also assume that C is sorted in descending order. Finally, you may assume that all grades in G can be assigned to exactly one of the letters. Take the following test cases as motivation. (Note: if you use **only** these test cases, you will not earn a good score.)

```
// Test 1:
List<Double> G1 = List.of(95.0);
List<String> L1 = List.of("A", "B", "C", "D", "F");
double[] C1 = new double[]{90, 80, 70, 60};
assignGrades(G1, L1, C1) => ["A"]

// Test 2:
List<Double> G2 = List.of(80.0);
List<String> L2 = List.of("A", "B", "C", "D", "F");
double[] C2 = new double[]{90, 80, 70, 60};
assignGrades(G2, L2, C2) => ["B"]

// Test 3:
List<Double> G3 = List.of(79.0, 85.5, 89.95, 90.14, 0.0, 50.0, 60.01);
List<String> L3 = List.of("A", "B", "C", "D", "F");
double[] C3 = new double[]{90, 80, 70, 60};
assignGrades(G3, L3, C3) => ["C", "B", "B", "A", "F", "F", "D"]

// Test 4:
List<Double> G4 = List.of(79.0, 85.5, 89.95, 90.14, 0.0, 50.0, 60.01);
List<String> L4 = List.of("A", "A-", "B+", "B", "B-", "C+", "C",
                          "C-", "D+", "D", "D-", "F");
double[] C4 = new double[]{93, 90, 87, 83, 80, 77, 73, 70, 67, 63, 60};
assignGrades(G4, L4, C4) => ["C+", "B", "B+", "A-", "F", "F", "D-"]
```