

## Loops

### 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.

*Remember:* you must submit your files to the autograder **EXACTLY** as we specify or it will be auto-rejected. Code that does not compile receives a 0 regardless of its style and tests. *Your* unit tests also must compile or we will not accept them.

**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:**

Recall the `isNestedParentheses` problem from the previous lab. Design the boolean `isNestedParenthesesLoop(String s)` method that uses a loop to solve the problem. Remember the translation pipeline!

## Problem 2:

Recall the `isFactorion` problem from the previous lab. Design the boolean `isFactorion-Loop(int n)` method that uses a loop to solve the problem. Remember the translation pipeline!

**Problem 3:**

Design the `int chickenCounterLoop(String s)` method, which receives a string `s` and returns the number of times the substring "chicken" appears in `s`. You must account for overlapping instances of "chicken". For example, calling the method with "abcchickechickenn" returns 2 because, after removing the substring "chicken" from the original string, we are left with "abc-chicken", which itself contains another instance of "chicken". **Your solution should use a loop and not be recursive.**

**(Extra Credit, 25 points) Problem 4:**

Design the boolean `isNumberPalindrome(int n)` method that, when given an integer  $n$ , returns whether or not that number is a palindrome. You cannot convert the number to a string. **Your solution should use a loop and not be recursive.**