

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, as a ZIP, to Canvas, and let one of the AIs 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:

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:

- (a) Design the standard recursive `int chickenCounter(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 "abcchicken", which itself contains another instance of "chicken".
- (b) Then, design the `int chickenCounterTR(String s)` method that uses tail recursion and accumulators to solve the problem. Hint: you will need to design a private static helper method to solve this problem.
- (c) Finally, design the `int chickenCounterLoop(String s)` method that solves the problem using a loop.

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.**