

CSC 210 - Software Development Programming Assignment 3

Due: Monday, September 13th at 11:59pm

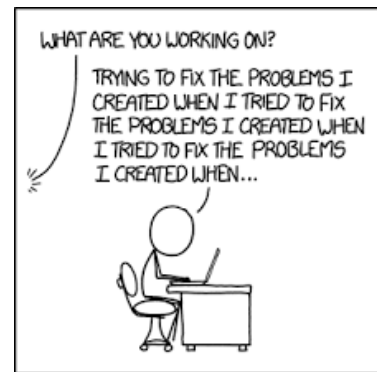
Submit: Recursion.java and RecursionTestClass.java to Gradescope

Recursion!

In computer science recursion is the process of defining a problem, or the solution to a problem, in terms of a simpler or smaller version of itself.

For a dictionary definition:

Recursion. See: *Recursion*



Requirements

Six recursive methods are needed:

- 1) Given two strings, s1 and s2, return the index in s1 at which s2 begins, or -1 if s2 is not in s1.
- 2) Given a stack, remove the first k even numbers and return the number of evens removed.
- 3) Given an integer, return a new number containing only the even digits in the same order, or 0.
- 4) Given a queue of characters, evaluate it as a mathematical expression and return the result.
- 5) Given a stack of integers, replace each integer with two copies of it and don't return anything.
- 6) Given a queue of integers, change each integer to twice its value and don't return anything.

Design

When designing each algorithm try to draw a diagram that shows how you want the algorithm to work. Think about how the problem can be made smaller and how the smallest or base case is defined. A recursive algorithm works by calling itself on a smaller version of the input and terminates when the base case is reached at which point the recursive calls begin returning. Sometimes the solution requires careful attention to what is returned by each recursive call.

Implementation

Two files are provided to you: Recursion.java and RecursionTestClass.java.

The Recursion.java file contains more detailed descriptions and prototypes for each function.

You may need to write some very short 'helper functions' but for the most part you will just need to implement the given function, and the code for a recursive function is typically very brief.

Testing

The RecursionTestClass.java contains code that is used to test the individual functions.

It uses JUnit, a Java library for unit testing. Unit testing is the process of examining a small "unit" of software, such as a single class or function, to verify that it meets its expectations.

JUnit isn't part of the standard Java class libraries but it does come with Eclipse.

You will need to write tests for each of your functions.

Remember you need to test the normal case, and the edge cases. What are all the strange (but valid) inputs that you might receive? What sorts of inputs might cause issues with your code?

Try to take a 'test-driven development' approach. Write your tests first, before your code. This requires an understanding of the problem, the valid inputs, and the expected outputs.