

Week 7 Homework Exercises

Mandatory Problems:

1. Write a method called **filterRange()** that
 - a. Accepts an **ArrayList** of integers and two **integer** values, *min* and *max*, as parameters
 - b. **Removes** all elements whose values are in the range *min* through *max* (inclusive) from the list. For example, if a variable called *list* stores the values:
[4, 7, 9, 2, 7, 7, 5, 3, 5, 1, 7, 8, 6, 7]
The call of `filterRange(list, 5, 7)`; should remove all values between 5 and 7, therefore it should change the list to store [4, 9, 2, 3, 1, 8].
 - c. If no elements in range *min-max* are found in the list, the list's contents are unchanged. If an empty list is passed, the list remains empty. You may assume that the list is not null.
 - d. Note: this method should not return anything.

Optional Problems:

1. In this assignment, you will model the game of Bulgarian Solitaire.
 - a. The game starts with N cards (they need not be playing cards; unmarked index cards work just as well.)
 - b. Randomly divide them into some number of piles of random size. For example, if the game starts with 45 cards, you might start with piles of size 20, 5, 1, 9, and 10.
 - c. In each round, you take one card from each pile, forming a new pile with these cards. For example, the sample starting configuration would be transformed into piles of sizes 19, 4, 8, 9, and 5.
 - d. The solitaire is over when the piles have size 1, 2, 3, 4, 5, 6, 7, 8, and 9, in some order. (It can be shown that you always end up with such a configuration if N is a [triangular number](#), that is $N = 1 + 2 + \dots + k$ for some k .)
 - e. In your program, produce a random starting configuration and print it. Then keep applying the solitaire step and print the result. Stop when the solitaire final configuration is reached.