Stack exercises:

Exercise 1. (*Trivial problem, but keep going …*) Write an algorithm that will move the top value on one stack to the top of another stack.

Exercise 2. *(A little more interesting …)* Write an algorithm that starts with a stack containing n integers and finishes with the same integers in the same stack, but with the value that was on the bottom of the stack moved to the top, and all other values moved down one position. For example if the stack initially looks like this:

4 ← top
17
9
23

then it should finish like this:

23
4
17
9

You may use another temporary stack in your algorithm.

Exercise 3. (*And now, an excellent stack question based on the first two!*) Write an algorithm that takes as input the integers {1, 2, ... , n} in a randomly determined arrangement on two stacks, and a target arrangement of the same integers on the same two stacks. Using ***only*** the methods created in exercises 1and 2, rearrange the integers to match the target arrangement.

For example suppose

n = 3,

start arrangement is $\begin{matrix}3\\1\end{matrix}$ on the first stack and $\underline{2}$ on the second stack,

target arrangement is $\begin{matrix}3\\2\\1\end{matrix}$ on the first stack and nothing on the second stack

One solution is

- move the top of Stack 1 to Stack 2 (as in Exercise 1)
- move the bottom of Stack 2 to the top of Stack 2 (as in Exercise 2)
- move the top of Stack 2 to the top of Stack 1
- move the top of Stack 2 to the top of Stack 1

It's not hard to create a generic algorithm that will transform any initial arrangement to any target arrangement ... but creating an algorithm that performs the transformation in the smallest number of steps is much more challenging.

Exercise 4: Improve the "move the smallest value to the top of the stack" algorithm from the notes so that it works properly when there are duplicates of the smallest value in the stack.

Exercise 5: Write a stack-based algorithm that will check a sequence of tokens to see if it is a well-formed postfix expression. For example, $1 \quad 3 \quad * \quad +$ and $+ \; 4 \quad 10$ are not well-formed. (Hint: adapt the postfix evaluation algorithm.)

Exercise 6: Given a stack S containing integers, write an algorithm that will reverse the order of the values in the stack.

Exercise 7: Given a stack S containing integers, write an algorithm that will re-order S so that all the even values are above all the odd values.

Exercise 8: Given a stack S containing integers and an integer k, write an algorithm that will re-order S so that the top value is swapped with the value that is k from the top. All other values in the stack should remain where they are. For example, if k = 3, the top value should be swapped with the value that is third from the top. If k is greater than the number of values in the stack, nothing should happen.