CS302: Homework 2

An exercise in stacks.

Exercise 1.
The first assignment was to create a “postfix calculator” using stacks. The program, when run, will output the following:

```
Exercise 1: Five postfix expressions.
(5+6)*(4/3) = 56+43/* = 14.6667
((5-1)/(6/3))^3 = 51-63//3^ = 8
6+9-3/5*4 = 69+35/4*- = 12.6
(7*7*5)-(8+9-5+0/3) = 77*5*89+5-03/+- = 233
4^2*3/6 = 42^3*6/ = 8
```

Part of the difficulty came from correctly translating from infix to postfix.

Exercise 2.
This program was meant to utilize the functionality of stacks to determine if delimiters were appropriately matched.

```
This program checks for properly matched delimiters.
1. {}[]()<> is valid
2. (() is invalid
3. {{[>]}} is invalid
4. {Used with non-brace characters} is valid
5. <{[]}(<>))> is invalid
```

A significant issue I faced was with the switch statement, wherein all opening delimiters would be pushed but only one closing delimiter would be popped before the program would say it is invalid, whether it was or not. This happened because the `break;` command was within the if-statement to check if the appropriate opening delimiter was found at the top of the stack. So, the switch statement would run down the list until it inevitably found a closing delimiter that did not have a matching opening delimiter, always returning 0.