**Exercise 1:**

For both exercises, I implemented a Linked List Stack, I felt with the linked list, it would be much easier for both, since there is no reason for either of these exercises to pull an element in the middle of the stack. For the postfix, I basically did it as explained within the homework pdf, I push numbers onto the stack until an operand is found, when it is, the program will pop the top 2 elements, and use those two elements with the operand(+, - , \*, /), and the last result in the stack would be popped and returned to the main. I could have made a separate function to determine the type of operand in terms of more organization, but it did not feel completely necessary to me, although it would have definitely made my code more clean and readable due to the repeated if statements.

**Output for postfix.cpp:**

1st: infix: (3 + 4) \* (2 / 5), postfix: 34+52/\*, is equal to: 2.8

2nd: infix: (8 + 3) \* (4 - 2), postfix: 34+52/\*, is equal to: 22

3rd: infix: (8 \* 3) + 7 - (2 / 7), postfix: 83\*7+27/-, is equal to: -27.5

4th: infix: (6 / 3) \* (2 - 5), postfix: 63/25-\*, is equal to: 1.5

5th: infix: (4 \* 3) + 6, postfix: 43\*6+, is equal to: 18

6th: infix: (5 - 2) \* (6 - 3), postfix: 52-63-\*, is equal to: 9

**Exercise 2:**

Delimeters, once again, uses a Linked List Stack implementation, for the same reason as mentioned above. And the process is generally the same, pushes any opening brackets onto the stack, and if it finds a closing bracket, it will pop the top element and check if that closing bracket has a matching opening bracket, and it will continue to do so until the stack is empty or the compiler finds that it does not have a matching opening bracket, which will result in the function returning as false. If the stack is found to be empty, then it has successfully found a match for every bracket, therefore the input was valid.

**Output for delimeters.cpp:**

This program checks for properly matched delimiters.

Test 1: (([<>])) is: Valid

Test 2: (This should be invalid is: Invalid

Test 3: {{)is: Invalid

Test 4: <>(){}[]is: Valid

Test 5: [This should be valid]is: Valid