There are two folders on the repository. One is for the queue program and the other is for the stack program. The queue program adds integers to a queue and performs functions with some output statements. There is no user input for the program.

The other folder is for the stack, which tests the validity of infix expressions and converts them to postfix.

The files in a folder should be added to the same directory and run by running the main.cpp file.

Some things to note about the stack file:

There is no implicit multiplication. If the user enters an expression such as 6\*5(4-3) then this expression is valid because the parenthesis match. When converting this expression to postfix form, the algorithm I used will not identify that there should be an additional multiplication operator between the product of 6\*5 and the difference of (4-3). It will output “6 5 4 3 - \*”.

I didn’t include implicit multiplication because that wasn’t in the assignment requirements.

I also wouldn’t want to have my program say that an expression like this is invalid because the assignment says that it should check for balanced parenthesis and ignore other characters. If you need more clarity on this just let me know and I would be glad to explain.

For this assignment, I used ***Dijkstra's Shunting-Yard Algorithm*** to convert the infix expression to the postfix expression. I believe it is the same algorithm used on the slides. I wanted to make sure to cite that I used this algorithm.

There is probably some input you could type for your input expression that may break the program, but most likely it will say valid expression unless parentheses aren’t balanced.

There are assumptions that the user types in “reasonable” mathematical expressions that may or may not be valid.

*From assignment handout:*

* *{, }, (, ), [, ] are the only symbols considered for the check. All other characters can be ignored*

Seth

Screenshots below:

Queue program:

A computer screen with white text

Description automatically generated

Stack program:

A screen shot of a computer

Description automatically generated

More complicated expression:

A computer screen with white text

Description automatically generated

Something like a string is valid because the assignment says to only check for balanced parentheses and also a string can be a variable (one operand):

A computer screen with white text

Description automatically generated