605.202 Data Structures LAB 1

Use of Stacks

In this assignment you will a stack to convert prefix expressions directly to postfix expressions. You may not use recursion as we will use recursion to revisit this problem in Lab 2.

Write a program that accepts a prefix expression containing single letter operands and the operators +, -, *, /, and $ (representing exponentiation). Output the corresponding postfix epression.

For example, if your input is ***AB** then output should be **AB*.** Output of **BA*** is considered incorrect.

In your analysis, be sure to discuss the implementation you choose and why, why a stack makes sense. Consider a recursive solution (you should not implement recursion) and compare it to your iterative solution. Is one better than the other? Why? Tell us what you learned and what you would do differently. Be sure to review the Programming Assignment Guidelines for specific requirements for the Analysis and before submitting this assignment.

Don't forget to do reasonable error checking and try to make your error messages specific and helpful.

You may not use library functions. You must write your own code and, specifically, you must write the stack code. Be sure to include the stack source code in your submission. You must read and write from named files.

The required input is provided in a separate text file.

In processing the input, you should expect to read line by line and withing a line, character by character, as specified in the programming assignment guidelines. This will allow you to parse the expression as you read it.