605.202 Data Structures LAB 1

Use of Stacks

Assume a Machine has a single register and six instructions (Problem 2.3.10 from the LAT text.)

LD	***A***	places the operand A in the register
ST	***A***	places the contents of the register into the variable A
AD	***A***	adds the contents of the variable A to the register
SB	***A***	subtracts the contents of the variable A from the register
ML	***A***	multiples the contents of the register by the variable A
DV	***A***	divides the contents of the register by the variable A

Write a program that accepts a postfix expression containing single letter operands and the operators +, -, *, and / and prints a sequence of instructions to evaluate the expression and leaves the result in the register. Use variables of the form *TEMPn* as temporary variables. For example, using the postfix expression ABC*+DE-/ should print the following:

```
LD   B
ML   C
ST   TEMP1
LD   A
AD   TEMP1
ST   TEMP2
LD   D
SB   E
ST   TEMP3
LD   TEMP2
DV   TEMP3
ST   TEMP4
```

Use the following postfix expressions for input

AB+C-
ABC+-
AB-C+DEF-+$
ABCDE-+$*EF*-
ABC+*CBA-+*
ABC+/CBA*+
AB-*CBA+-*
ABC-/BA-+/
ABC+$CBA-+*
AB0+/CBA+-/

Keep in mind that you are NOT evaluating the postfix expression, you are generating the machine language instructions that would perform the evaluation if actually executed.

In your analysis, be sure to discuss the implementation you choose and why, why a stack makes sense. Consider a recursive solution (you do not need to 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, including the formatting requirements for the analysis. You may not use library functions. You must write your own code, in particular you must write the stack code. Be sure to include the stack source code in your suibmission. You must read and write from named files.