CS-4100/5100
PJ05

Infix to Postfix Converter

For the operators specified in HW05 Problem #1, write a program that converts expressions in "normal" infix notation into postfix (i.e., RPN, notation).

The input to your program will come from the file infix.txt with one expression per line. Your output should be contained in two files. The first is postfix.txt with one RPN expression per line and the other is results.txt which contains the result of evaluating the RPN expression. A given line number in each of the three files should all refer to the same expression. If your program cannot handle a particular line, you need to output something for that line. You can either output whatever results, or you can output "ERROR" to the RPN file and "NaN" to the results file.

For simplicity, the expression operands will be either function calls or non-negative 16-bit singed integers (i.e., values between 0 and +32767, inclusive). Your answers should be 16-bit signed integers (i.e., values between -32768 and +32767, inclusive). You evaluations should reflect the use of 16-bit arithmetic, meaning that overflow in either direction should result in "wrapping around"; thus 32767 + 1 should result in -32768.

The only functions that you need to support in your evaluations are sq(), abs(), min(), and max(). The sq(x) function returns the square of x, the abs(x) function returns the absolute value of x, the min(x,y) function returns the minimum value of x and y; the max(x,y) function returns the larger of the two values.

As usual, the relational and logical operators return 0 for false and -1 for true.

You should submit a zip file using the expected naming convention with a folder named 'code' that has your source code. At the top level should be both output files. The input file does not need to be included, but it is fine if it is.

To aid in grading, the format off your postfix.txt file should be comma delimited with no spaces. For, for example, the input

2 * sq(123 – 234) << 36 % 17

should yield

2,123,234, –,sq,*,36,17,%,<<

Note that the result of this is -32504. Be sure you understand why this is the case.