Implement the RPN calculator pseudo-code supplied in the lecture as real code. Your program must:

1. Prompt the user for an infix math problem.
2. Convert the problem to postfix.
3. Output the problem in postfix.
4. Calculate the result.
5. Display the result.
6. Ask the user for another math problem.

If the user enters *quit* for the problem, end the program.

Use the standard stack and queue class/methodology provided by your preferred language's framework, such as the STL stack/queue classes in C++ or the Array class in JavaScript. Put the conversion procedure in its own function. Put the calculate result procedure in its own function as well.

You need to handle multi-digit numbers even though the sample code does not. In your version, numbers (operands) are separated from operators by zero or more spaces, while numbers are separated from other numbers by *one* or more spaces. While negative numbers cannot be input by the user, the result may be negative based on the input math problem.

You must support +, -, \*, /, and % operators, as well as (potentially nested) parenthesis. For +10 extra credit, also support raising a number to a power with the POW operator, which must appear as those three letters in all uppercase.