CS234: DATA TYPES AND STRUCTURES (Spring, 2010)
Assignment 2: Arbitrary precision integer calculator.
Due on June 18th by 5pm

Overview: This is the second of two programming assignments that, when combined, implement a simple reverse polish notation (RPN) integer calculator that can manipulate arbitrarily large integer values. In the first assignment, you implemented an abstract data type (ADT) APnum for manipulating arbitrary precision integers. In this assignment, you use APnum to implement the calculator itself.

Requirements for the Calculator: Use APnum to implement a RPN (postfix notation; cf. pages 84–86 in the textbook) calculator that reads a postfix arithmetic expression terminated by a “;”, evaluates the expression using the APnum ADT, and prints the result(s). The operations supported by the calculator are as follows:

\[(\text{input}) \quad \quad (\text{action})\]

\[
\begin{align*}
nnn & \quad \text{enter a new number} \\
C & \quad \text{replace last entry by its complement} \\
+ & \quad \text{replace last two entries by their sum} \\
= & \quad \text{replace last two entries by 1 if they are equal and by 0 otherwise} \\
< & \quad \text{replace last two entries by 1 if the last is the greater of the two and by 0 otherwise} \\
D & \quad \text{duplicate the last entry} \\
S & \quad \text{swap the last two entries} \\
P & \quad \text{print out all (remaining) entries and empty the stack} \\
Q & \quad \text{terminate execution of the calculator}
\end{align*}
\]

where \(nnnn\) stands for a (potentially very long) sequence of digits. The calculator should report an error if an operation is requested but for which there are not enough values present to complete the operation.

The calculator reads its input from standard input and prints the results to standard output. The output is to appear as in the following example. In particular, the calculator must be able to read consecutive RPN expressions and should terminate only when a \(Q\) command occurs in the input stream. The symbols on the input may be separated by spaces (this is necessary for consecutive numbers) and the number of spaces must not matter. For example the inputs “1 2+C P” and “1 2 +C P” must be treated as the same expression. Also note that “new line” characters in the input stream should be ignored.
A Sample Session: The following sequence of characters in the standard input

```
12 34 56+C78
C++
P
12D D7 + P +Q
```

should produce the following sequence of characters on the standard output

```
Result(s):
-156
Result(s):
19
12
12
Error (not enough operands)
```

Assignment submission: You are to use the `submit` command to submit a full implementation of your calculator in either PYTHON or JAVA by the assignment due date.