Using a stack to evaluate an arithmetic expression in RPN format

Task

The task is to write a program that uses a stack to evaluate simple postfix (RPN) expressions such as “34*9-”

For this exercise the rules for defining RPN expressions are:

- an expression consists only of operators and operands
- each operand is a single decimal digit (0..9)
- the permitted operators are * / + and -

The expression will be entered by the user as a string or defined within the program as a string.

Algorithm

```
READ expression
FOR each character in the expression
   IF not an operator THEN {its an operand}
         push to stack
   ELSE {it must be an operator}
        pop operand B
        pop operand A
        apply the operator to operands A and B
        push the result
   ENDIF
ENDFOR
```

Implementation

Define a stack class and then use it to create a stack object. Use the stack object to evaluate a postfix (RPN) expression.

Testing

Test your program by evaluating the following expression: 34*9-38-+

Extension

1. Extend your program to process multiple character operands.

2. Extend your program to include exponentiation – what problem does this pose?