**Homework**

Write a program that accepts a string that is a space-delimited arithmetic expression in postfix notation. Use a vector to compute the value.

Example:

The input string 5 4 + 3 10 \* + is equivalent to the infix expression (5 + 4) + (3 \* 10)

The answer is 39.

The algorithm is:

Take the leftmost token from the string

If it is numeric (which may include a unary operator), push that token onto the vector

If it is a binary operation, pop the vector twice, apply the operation, and push the result

When the input string is empty, the vector will contain one entry: the final answer

Here’s how the given string gets processed:

String Vector

5 4 + 3 10 \* + <empty>

4 + 3 10 \* + 5

+ 3 10 \* + 4 5

3 10 \* + 9

10 \* + 3 9

\* + 10 3 9

+ 30 9

39

The final answer, 39, is the only element in the vector after all tokens in the string have been consumed.

Use atoi to convert string to integer

\*\*notes subtraction, division, and mod are not commutable

\*\* unary operators such as -9 or +9

\*\*if length is one then it then inspect if operator or operand

\*\*if two or more then char then it is a number!

This problem has an extra twist. The string contains characters, so 10 isn’t a ten, but is the char 1 followed by the char 0. You will need to convert string representations of integers (signed and unsigned) into their integer values. You’ll also need to handle unary – and +.

We also have to worry about the non-commuting operators – and / . We will evaluate the postfix string 4 5 – as 4 – 5 and, likewise, will evaluate 4 5 / as 4 / 5 .

The program should continue to accept and process input strings until the user enters a zero as input. There’s no need to check for validity; all input will be well-formed postfix expressions.

As always, provide source code and executable on media, and source code on paper. Valid test strings will contain only integers, the two unary integer operations, and the five binary integer operations. Due at the beginning of class next week.