

Explanations for the Expression Evaluator

Postfix notation is a way of writing algebraic expressions without the use of parentheses or rules of operator precedence. The expression above would be written as AB+CD-/ in postfix notation. (Don't panic! We'll explain this in a moment.) Postfix notation had its beginnings in the work of Jan Łukasiewicz* (1878-1956), a Polish logician, mathematician, and philosopher. Łukasiewicz developed a parenthesis-free prefix notation that came to be called Polish notation and a postfix notation now called Reverse Polish Notation or RPN. From these ideas, Charles Hamblin developed a postfix notation for use in computers. Łukasiewicz's work dates from about 1920. Hamblin's work on postfix notation was in the mid-1950's. Calculators, notably those from Hewlett-Packard, used various postfix formats beginning in the 1960s.

One algorithm for postfix expression evaluation can be found at:

<http://www.maths.abdn.ac.uk/~igc/tch/mx4002/notes/node74.html>

The method used in our sample uses the following principle: any time we can get the shortest valid expression (i.e. two operands followed by an operator), we replace that expression in the original text with its value, and repeat the process on the resulting text till we are left with only one operand. That will be the result of the evaluation. Our implementation takes care of skipping white space which separates elements.

The expression is assumed to be written on a single line, which is read into character buffer named bufferA.

Buffer bufferB is used for constructing the new expression, and at the end of each loop, it replaces the contents of bufferA.

Example. For the infix floating point expression

((0.2 + 1.3) * 2 - (24 / 3)) / 2,

the postfix equivalent is

0.2 1.3 + 2 * 24 3 / - 2 /

The successive transformations are:

1.5 2 * 24 3 / - 2 /

3 24 3 / - 2 /

3 8 - 2 /

-5 2 /

-2.5

Here, with bold red characters we emphasize the expression which will be evaluated next.