Infix to Postfix Conversion

Thursday, February 16, 2016 8:56 AM

We need: 1) Dictionary of precedences (eg. [("/",1), ("*",1), ("+",0)...])
("/" ← "operator", 1 ← "operand")
2) Input queue (infix expression)
3) Output queue (final result)
4) Stack of operators (initially empty)

Algorithm: While there are tokens to read:
- Read token
- if the token is an operand (number):
  → add it to the output queue
- If the token is an operator:
  → Keep removing operators from the stack (if any) until the precedence of the operator on top of the stack has **strictly less** precedence than the token's precedence (the ones removed go to the output queue.) **OR** there's a left parenthesis "(".
- If the token is a left parenthesis:
  → push it onto the stack
- If the token is a right parenthesis:
  → Until the token/operator at the top of the stack is a left parenthesis:
    → Keep popping operators from the stack and onto the output queue.
  → pop the "(" from the stack

Pop rest of the operators on the stack, if any, to the output queue.

Infix to Postfix Example

Thursday, February 18, 2016 9:43 AM

9 * 4 + 14 - (7 * 10) / 2

Dictionary: {("-", 1), ("*", 2), ("/", 3)}

Input q	Output queue:	Stack
9 * 4 + 14 - (7 * 10) / 2		
9 * 4 + 14 - (7 * 10) / 2	9	
9 * 4 + 14 - (7 * 10) / 2	9	*
9 * 4 + 14 - (7 * 10) / 2	9 4	*
9 * 4 + 14 - (7 * 10) / 2	9 4 *	~~*~~ +
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14	+
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 +	~~+~~ -
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 +	- (
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7	- (
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7	- (*
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7 10	- (*
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7 10 *	- ~~(*)~~
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7 10 *	- /
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7 10 * 2	- /
9 * 4 + 14 - (7 * 10) / 2	9 4 * 14 + 7 10 * 2 / -	~~+ /~~

Output queue: 9 4 * 14 + 7 10 * 2 / -

↳ ~~9~~ ~~4~~ ~~*~~ 14 + 7 10 * 2 / -

~~36~~ ~~14~~ ~~+~~ 7 10 * 2 / -

5 ~~0~~ ~~7~~ ~~10~~ * 2 / -

50 ~~70~~ ~~2~~ ~~/~~

~~50~~ ~~35~~ (15)