**TRANSFORM**

**Transform ():**

**Description:** Here I is an arithmetic expression written in infix notation and P is the equivalent postfix expression generated by this algorithm.

1. Push "(" left parenthesis onto stack.
2. Add ")" right parenthesis to the end of expression I.
3. Scan I from left to right and repeat step 4 for each element of I until the stack becomes empty.
4. If the scanned element is:
   (a) an operand then add it to P.
   (b) a left parenthesis then push it onto stack.
   (c) an operator then:
      (i) Pop from stack and add to P each operator which has the same or higher precedence then the scanned operator.
      (ii) Add newly scanned operator to stack.
   (d) a right parenthesis then:
      (i) Pop from stack and add to P each operator until a left parenthesis is encountered.
      (ii) Remove the left parenthesis.

   [End of Step 4 If]
   [End of step 3 For Loop]
5. Exit.