Conversion from infix to postfix expression:

Algorithm

POSTFIX (Q, P)

Suppose Q is an arithmetic expression written in infix notation. This algorithm finds the equivalent postfix expression P.

Step1: Push “(” onto STACK and add ’’)’’ to the end of Q.

Step2: Scan Q from left to right and repeat step 3 to 6 for each element of Q, until the STACK is empty.

Step3: If an operand is encountered, push it to STACK.

Step4: If a left parenthesis is encountered, push it to STACK.

Step5: If an operator X is encountered, then
a) Repeatedly pop from STACK and add to P each operator (Top of Stack)
which has same precedence as or higher precedence than X.
b) Add X to STACK.

Step6: If a right parentheses is encountered, then;

a) Repeatedly pop from STACK and add to P each operator (top of stack)until a left parentheses is encountered.

b) Remove the left parentheses from stack [Do not add it to P]

```
[End of if]
[End of step 2 loop].

Step7: Exit.
```

ALGORITHM TO EVALUATE POSTFIX EXPRESSION

```
P → postfix expression
1. Add a right parenthesis ")" at the end of P
2. Scan P from left to right and repeat steps 3 & 4 until sentinel ")" is
   encountered
3. If an operand is encountered, put it on stack
4. If an operator    is encountered , then:
   a) Remove the top two elements of stack , where A is the top element
      & B is the next to top element
   b) Evaluate B   A
   c) Place the result of (b) back on stack
5. Set value equal to the top element on stack
6. Exit
```