Infix to Postfix

1. Push " (" onto stack, and add ") " to the end of X.
2. Scan x from left to right and repeat step 3 to 6 for each element of x until the stack is empty.
3. If an operand is encountered, add it to Y.
4. If a left parenthesis is encountered push it onto stack.
5. If an operator is encountered then:
   1. Repeatedly pop from stack and add to Y each operator (on top of stack) which has same precedence as or higher precedence than operator.
   2. Add operator to stack.
   [End of if]
6. If a right parenthesis is encountered then:
   1. Repeatedly pop from stack and add to Y each operator (on top of stack) until a ~~left~~ left parenthesis is encountered.
   2. Remove left parenthesis.
   [End of if]
   [End of if]
7. END

Pseudo Code

```
Infix to Postfix (exp)
{
  create a stack s
  for i <- 0 to length(exp)-1
  {
    if exp(i) is operand
```

Scanned with CamScanner

```
res <- res + exp(i)
else if exp(i) is operator
    while ( !S.empty() && HasHigherPre( S.top(), exp(i)))
    {
        res <- res + S.top()
        S.pop()
    }
    S.push(exp(i))
else if Is opening Parenthesis (exp(i))
        S.push(exp(i))
else if Is closing parenthesis (exp(i))
        {
            while( !S.empty() && !Is opening parenthesis(S.top()))
                {
                    res <- res + S.top()
                    S.pop()
                }
                S.pop()
            }
    }
    while ( !S.empty())
    {
        res <- res + S.top()
            S.pop
    }
    }
    return res
}
```

Scanned with CamScanner