Conversion from Infix to Postfix Using a Stack

Although the algorithm to evaluate postfix expressions is straightforward, it is often more natural to write an expression in infix notation

For example, the infix expression:

A **+** B ***** C **+ (** D ***** E **+** F **)** ***** G

is equivalent to the following postfix expression:

A B C ***** **+** D E ***** F **+** G ***** **+**

Infix expressions can be converted to postfix using a stack

For each symbol in the infix expression, if the symbol is an operand, it is written as the next symbol in the postfix expression

Otherwise if the symbol is an operator:

First, the top of the stack is checked for any operator of higher or equal precedence, if so this operator is popped from the stack and written to the postfix expression

Converting an Infix Expression to Postfix

The top of the stack is checked again for any operator of higher or equal precedence removing these from the stack and placing in the postfix expression until a lower precedence operator (or the bottom of the stack) is reached

After this the current operator read from the infix expression is pushed onto the stack

Precedence when reading symbols from the infix expression:
(: highest precedence
) : special case: pop operators from stack
    until reaching '('
*,/,%
+,-: lowest precedence

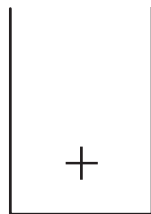Precedence when popping from the stack:
*,/,% : highest precedence
+,-
(: lowest precedence, to keep '(' on the stack
    until reaching ')'

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

```
|     |
|     |
|     |
|  +  |
|_____|
```

Stack

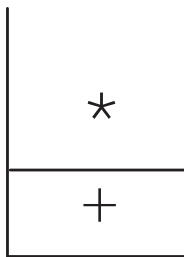| a b |
|------|

Output

Converting an Infix Expression to Postfix

Example: converting the following infix expression to postfix
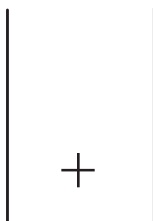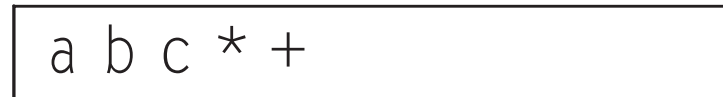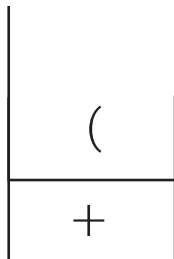
$$A + B * C + (D * E + F) * G$$

Stack:
```
*
+
```
Stack

Output: a b c

Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

+

Stack

a b c * +

Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

(
+

Stack

a b c * + d

Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

```
*
(
+
```
Stack

```
a b c * + d e
```
Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

Stack:

```
+
(
+
```

Stack

Output:

```
a b c * + d e * f
```

Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

+

Stack

a b c * + d e * f +

Output

Example: converting the following infix
expression to postfix

$$A + B * C + (D * E + F) * G$$

*
+

Stack

a b c * + d e * f + g

Output

Converting an Infix Expression to Postfix

Example: converting the following infix expression to postfix
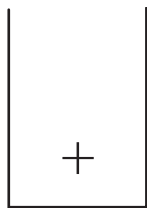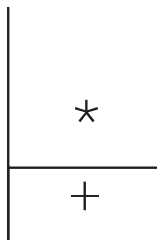
$$A + B * C + (D * E + F) * G$$

Stack

a b c * + d e * f + g * +

Output