Infix to Postfix Conversion Using Stack.

Infix Expression:-

An infix expression can be represented as: if a, b as operand => a <operator> b

e.g a + b

postfix operation :- can be represented as: a b <operator>

e.g a b +

Priority order :- (highest to low)

(1). '^'

(2). '*', '/'

(3). '+', '-'

Rules :- / Algorithm (can be)

-> start by inserting '(' in the stack.

-> if character == operand,
put it in postfix expression

-> if character == operator,
put it in stack.

→ if scanned operator priority (e.g '*') > top operator priority (e.g '+"), push in stack

→ if scanned operator priority eg '+' < top operator priority e.g "*", pop till priority [scanned] becomes > priority [top]

→ if ')' , encountered, pop till '('

Example:- (A+B/C*(D+E)-F)

Symbol	Stack	Postin Expression.
(	(	
A	C	A
+	C+	A
B	C+	AB
-	C+	AB
/	C+/	AB
C	C+/	ABC
*	C+*	ABC/
(	C+*(	ABC/
D	C+*(	ABC/D
+	C+*C+	ABC/D
E	E+*C+	ABC/DE
)	(+*(+) (here it is removed)	ABC/DE+
	'('	ABC/DE+*+ and '+' moved here
-	(-	ABC/DE+*+
F	(-	ABC/DE+*+F
)	(-) (closed)	ABC/DE+*+F

⇒ ABC/DE+*+F