

Step 1. Create an empty stack and an empty postfix expression string.

[], ""

Step 2. Scan the infix expression from left to right. $2*(3+4*7)-5$

Step 3. For each element in the infix expression:

Step 3.1. If the element is an operand (a number or a variable), append it to the postfix expression string.

Step 3.2. If the element is an opening parenthesis '(', push it onto the stack.

Step 3.3. If the element is an operator:

i. While the stack is not empty and the top element of the stack is not '(' and the top element of the stack is an operator with higher or equal precedence than the current operator, pop the operator from the stack and append it to the postfix expression string.

ii. Push the current operator onto the stack.

Step 3.4. If the element is a closing parenthesis ')':

i. While the top of the stack is not an opening parenthesis '(', pop the operator from the stack and append it to the postfix expression string.

ii. Pop the opening parenthesis '(' from the stack (and discard it).

Step 4. While the stack is not empty, pop the remaining operators from the stack and append them to the postfix expression string.

The order of precedence for operators is typically: parentheses > exponentiation > multiplication/division > addition/subtraction. If two operators have the same precedence, they are processed from left to right (left-associative).

{addition:1, subtraction:1, multiplication:2, division:2, exponentiation:3}

Sr. no.	Expression	Stack	Postfix
0		(
1	2	(2
2	*	(*	2
3	((* (2
4	3	(* (23
5	+	(* (+	23
6	4	(* (+	234
7	*	(* (+ *	234
8	7	(* (+ *	2347
9)	(*	2347*+
10	-	(-	2347*+*
11	5	(-	2347*+*5
12)		2347*+*5-

POSTFIX Evaluation

Step 1. Create an empty stack. []

Step 2. Note down your postfix Expression. 2347*+5-

Step 3. For each element in the postfix expression(left to right):

Step 3.1. If the element is an operand (a number or a variable), push it onto the stack.

Step 3.2. If the element is an operator:

i. Pop the top two operands from the stack (let's call them operand1 and operand2).

ii. Perform the operation specified by the operator on operand2 and operand1 (in that order) and push the result back onto the stack.

Step 4. After scanning the entire postfix expression, the final result will be the single value left on the stack.

Sr. no.	Expression	Stack
0	2	2
1	3	2 3
2	4	2 3 4
3	7	2 3 4 7
4	*	2 3 (4*7)
5	+	2 (3+(4*7))
6	5	2 (3+(4*7)) 5
7	-	((3+(4*7))-5) = 26