

## Infix to Postfix using stack:

36

→ Converting an infix expression to its postfix using stack:

- 1) Start moving left to right from the beginning of the expression.
- 2) The moment you receive an operand, concatenate it to postfix expression string
- 3) And the moment you encounter an



operator, move to the stack along with its relative precedence number and see if the topmost operator in the stack has higher or lower precedence. If it's lower, push this operator inside the stack. Else, keep popping operators from the stack and concatenate it to the postfix expression until the topmost operator becomes weaker in precedence relative to the current operator.

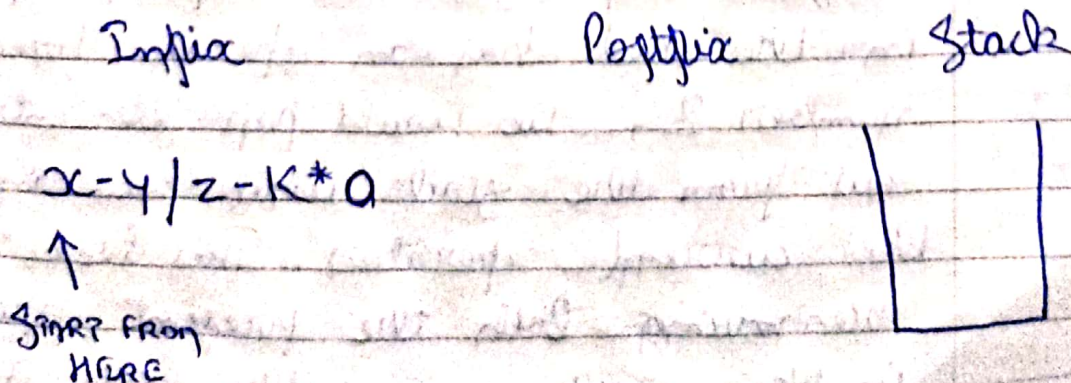
- 4) If you reach the EOF, pop every element from the stack, and concatenate them as well. And the expression you will receive after doing all the steps will be the postfix equivalent of the expression we were given.

Precedence Table

*	/	(2)
+	-	(1)

Ex:  $x - y / z - k * a$

- 1) He will start traversing from left:



- 2) First, we got the letter 'x'. He just pushed it into the postfix string.



The we get the subtracting symbol '-', and we push it into the stack since the stack is empty.

INFIX	POSTFIX	STACK
$x - y / z - k * a$ ↑	X	<div style="border: 1px solid black; padding: 5px; display: inline-block;">-, 1</div>

3) Similarly push '/' as it has greater precedence (2)

INFIX	POSTFIX	STACK
$x - y / z - k * a$ ↑	XY	<div style="border: 1px solid black; padding: 5px; display: inline-block;">/, 2 -, 1</div>

4) The next operator we encounter is again a subtraction. Since the topmost operator in the stack has an operator precedence number 2, we would pop elements out from the stack until we can push the current operator. This leads to removing both the present operators in the stack since they are both greater or equal in precedence. Don't forget to concatenate the



hopped operators to the postfix expression.

INFIX	POSTFIX	STACK
$x-y/z-k*a$ ↑	$xyz/-$	<div style="border: 1px solid black; padding: 5px; display: inline-block;">-, 1</div>

- 5) Next, we have a multiplication operator whose precedence number is 2 relative to the topmost operator in the stack. Hence we simply push it in the stack.

INFIX	POSTFIX	STACK
$x-y/z-k*a$ ↑	$xyz/-k$	<div style="border: 1px solid black; padding: 5px; display: inline-block;">*, 2 -, 1</div>

- 6) And then we get to the EOE and still have two elements inside the stack. So, we just pop them one by one, and concatenate them to the postfix. And this is when we succeed in converting the infix to the postfix expression.

INFIX	POSTFIX	STACK
$x-y/z-k*a$	$xyz/-ka*-$	<div style="border: 1px solid black; padding: 5px; display: inline-block;"></div>