Name: David Kamau Wambui 25/10/2024.
Reg no: SCT222-0284/2022. Unit code: ICS 2105
Unit name: Data structures and Algorithms

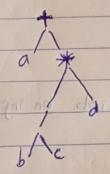
1a. Convert the following expression into an expression tree using a stack and show the process: a+b/c*d-e

Step 1: To parenthesize the expression based on the priority of the operators.
In this case division and multiplication have higher precedence than addition and subtraction hence evaluate from left to right.

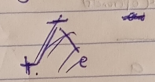
[(a+((b/c)*))-e)

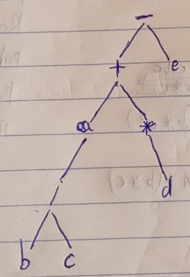
Construct the subexpression. b/c:

Then (b/c) * d and add a+((b/c)*d):



Finally substract e from the whole expression.





b. Convert the expression above into a postfix expression.
a+b/c*d-e into a post-fix expression. Using stack.

Incoming Token	stack	postfix
a	-	a
+	+	a
b	+	ab
/	+/	ab
c	+/	abc
*	+*	abc/
d	+*	abc/d
-	-	abc/d*+
e	-	abc/d*+e-

final expression ⇒ abc/d*+e-

2. Given the following Prefix expression Convert it into an infix using a stack, show workings. +a*bc.

Prefix expression	Stack(operands)	action.
+a*bc	——	Read operator + wait for operands
a*bc	a	Push operand a onto the stack
*bc	a	Read operator * waiting for operand
bc	a,b	push operand b onto the stack
c	~~(a,b*c)+~~ a,b*c	push operand c ont the stack
Empty	(a+(b*c))	Apply

Infix expression ⇒ a+(b*c)