NAME : GRACE GIKANDI
REG NO : SCT222 - 0163/2023.

a) Convert the following expression into an expression tree using a stack and show the process:

a + b / c * d - e.

~~Incoming Token~~ stack

~~Operator~~ ~~demand~~

a

~~started~~

~~content~~

Step 1; Paranthesize the expression based on operator priority:

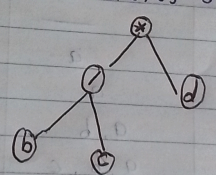
- Division (/) and multiplication (*) have higher priority than addition (+) and subtraction (-) and are evaluated from left to right.

Paranthesize the expression:

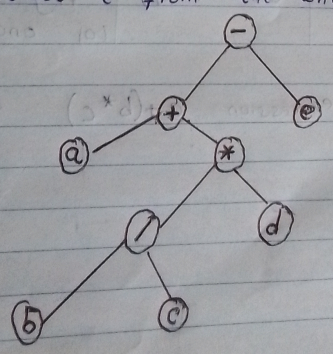
((a + ((b/c) * d)) - e)

Step 2; Build the tree from innermost expression outward.

First construct b/c then subexpression (b/c) * d.



Now add a+((b/c)+d) and finally subtract e from the whole expression.



b) Convert the expression into a postfix expression.

a + b / c * d - e

Incoming tokeo	stack	postfix
a	—	a
+	+	a
b	+	ab
/	+/	ab
c	+/	abc
*	+* (pop /)	abc/
d	+*	abc/d
-	- (pop *)	abc/d*+
e	-	abc/d*+e

pop remaing - abc/d*+e-

Final postfix Expression abc/d*+e-

c Convert into infix using stack.

+a*bc.

Prefix expression	stack	Action.
+a*bc	—	Read operator(+)
a*bc	a	Push operand (a) onto stack
*bc	a	Read operator (*)
bc	a, b	Push operand (b)
c	a, b, c	Push operand (c)
—	a, (b*c)	Apply (*) to be and c
—	(a+(b*c))	Apply (+) operator to (a) and (b*c)

Final infix expression a+(b*c)