

Assignment 1.

- 1) Convert the following expressions into an expression tree using a stack
 $a + b/c * d - e$

And show the process.

- 2) Convert the same expression into a postfix expression.

- Given the following prefix expressions $+ a * b/c$ convert it into an infix using a stack show your work.

Incoming Token.

(postfix element)

stack

stacked

content

a

a

b

ab

c

abc

/

/ - pop b/c

ba/c

d

ad

*

*

la*d

+

a+t*

e

te

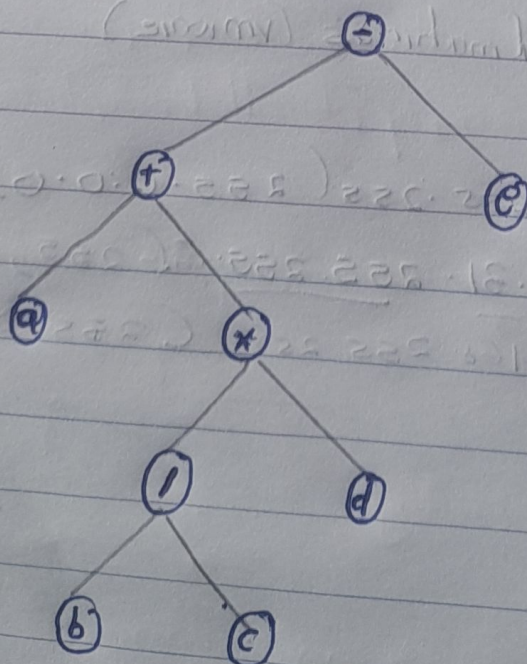
-

+(-)e

ii. Final binary tree

The root function is (-)

Expression root tree. $a + b/c * d - e$



2. Given the following prefix expressions $+a*bc$ convert it into an infix using a stack. show your work.

token	stack	stacked content
c	-	c
b	-	cb
*	* - pop bc	b*c
a	*	ab*c
+	+ * pop(ab*c)	a+b*c

Final expression = $a+b*c$

Explanation; scan the expression from right to left $\Rightarrow cb*a+$

- Push the operands onto the stack.
- When encountering an operator, pop 2 operands from the stack, combine them with the operator in infix forms and push the result back to the stack.