

* Input

Infix expression

* Outcome

- 1) equivalent postfix expression
- 2) Result of evaluation of an expression

* Theory

* Infix to postfix conversion

Infix notation is the common way to writing expression (eg - A+B) postfix notation (or Reverse) parenthesis an infix expression to postfix, we can use the shunting yard algorithm developed by edsger dijkstra

step for conversion

- 1) Initialize an empty stack and an empty output list
- 2) Read the infix expression from left to right
- 3) for each token
 - If it's an operand (like A,B,C) add it to the output
 - If it's right parenthesis pop from the stack to the output until a left parenthesis is top the stack Remove the left parenthesis.

After reading all tokens pop any remaining operators from the stack to the output

e.g)

Infix expression : $A + B * C$

- 1) Read A output $\rightarrow A$
- 2) Read + stack $\rightarrow +$
- 3) Read B output $\rightarrow AB$
- 4) Read * stack $\rightarrow +*$ (because * has higher parenthesis than +)
- 5) Read C output $\rightarrow ABC$
- 6) end expression " pop remaining operator
output $\rightarrow ABC*+$

postfix expression : $ABC*+$

\therefore The result shows that you should multiply B and C first then add A

* Algorithm

step1) start

step2) start reading the infix expression from left to right

step3) Repeat step ④ to ⑦ for each element until stack is empty

step4) If we scan operand we output it printit

step5) Else

If the scanned operator is greater is precedence than the operator in the stack or if the stack is empty or the stack contain a '(' push it else

pop all the operators having greater or equal precedence than of the scanned operator

Step 6) If a 'c' is encountered push it onto stack

Step 7) If a 'd' is encountered repeatedly pop from stack and output it until a 'c' is encountered

Step 8) The output is printed in post fix

Step 9) Stop

* flowchart

(start)

Read infix expression

scan from left to right one by one

Is

stack empty

No

Is

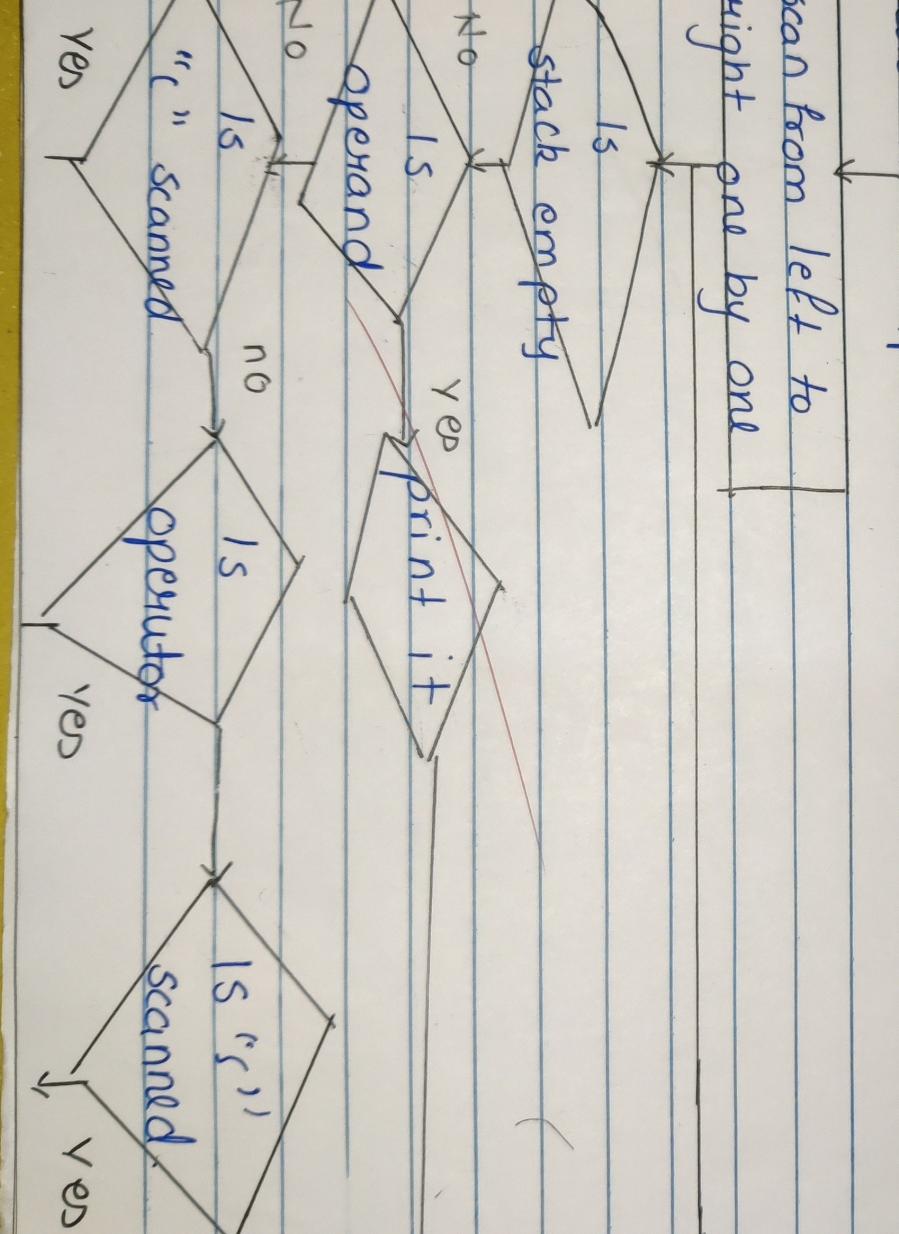
operator

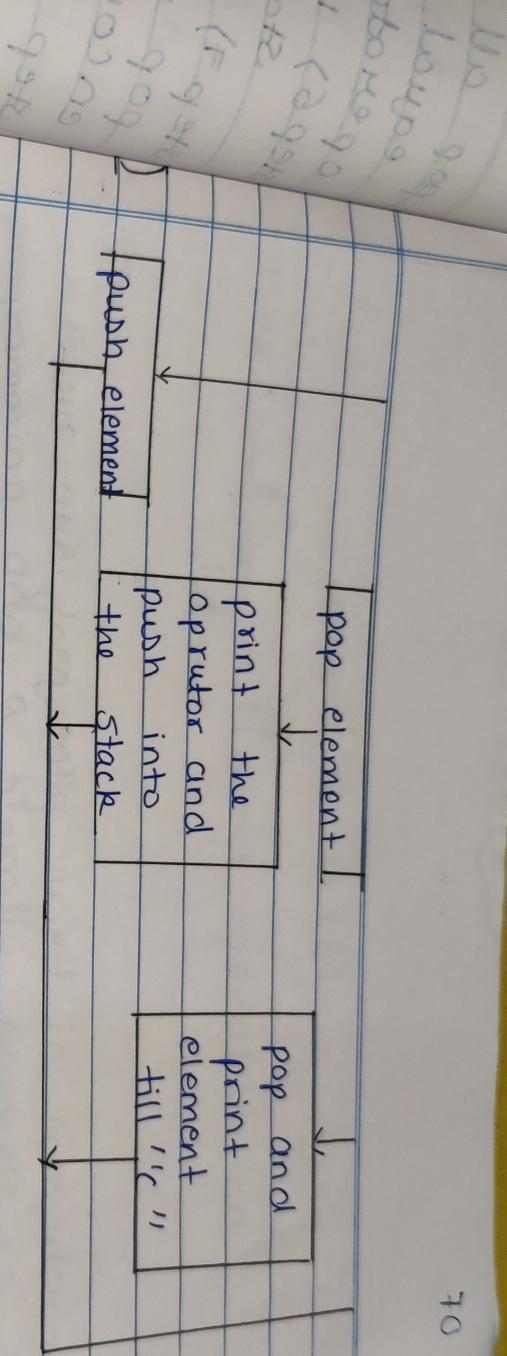
No

Is

"(" scanned

Yes





* conclusion

By this way we can perform expression as infix to postfix and it's evaluations using stack