

Stack Application:

Balancing Symbol

1. Make an empty stack
2. Reading Characters
3. If the char is opening symbol -> push
4. If the char is closing symbol
  - a. Check stack
    - i. If empty -> error
    - ii. Else -> pop
      1. If it is not the corresponding opening, error
5. At the end if the stack is not empty -> report error

$a + \{b + c * (d - e)\}$


Infix to Postfix

$a + b * c$

ab+

In postfix expression, we do not have to consider the precedence

## Infix to Postfix

- Read operand - > output
- Read operator
  - if the stack is empty or has parenthesis on top -> Push
  - else check precedence with Existing Operator (E.O.)
    - if  $C.O > E.O$  -> Push C.O. (Current Operator)
    - if  $E.O \geq C.O$  -> Pop E.O. and Push C.O.
- Read Parenthesis
  - If Opening Parenthesis (O.P) -> Push
  - If Closing Parenthesis (C. P.) -> Pop all operands till O.P.
- Finish reading input
  - Pop the rest of the elements (operators)