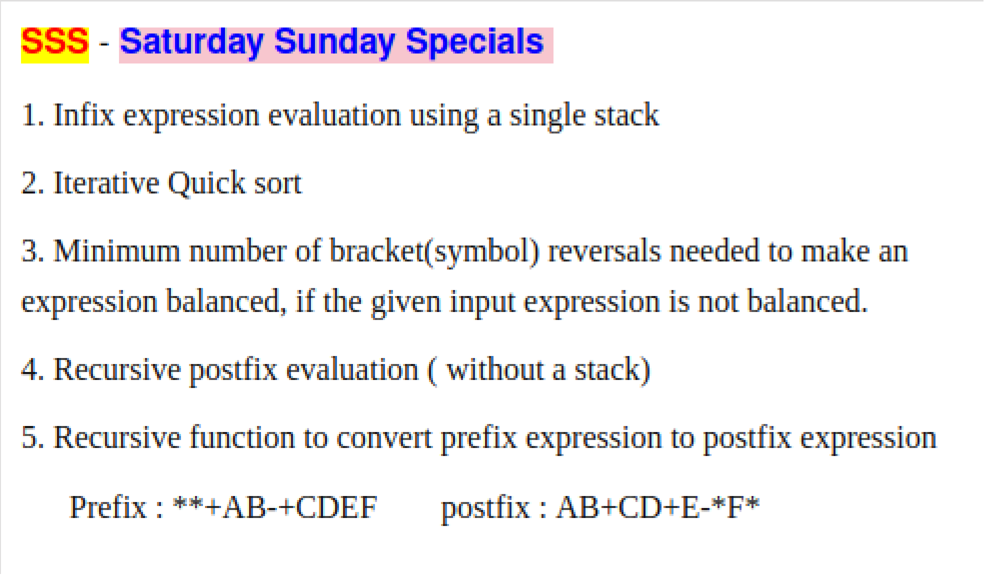


SSS - Saturday Sunday Specials

1. Infix expression evaluation using a single stack

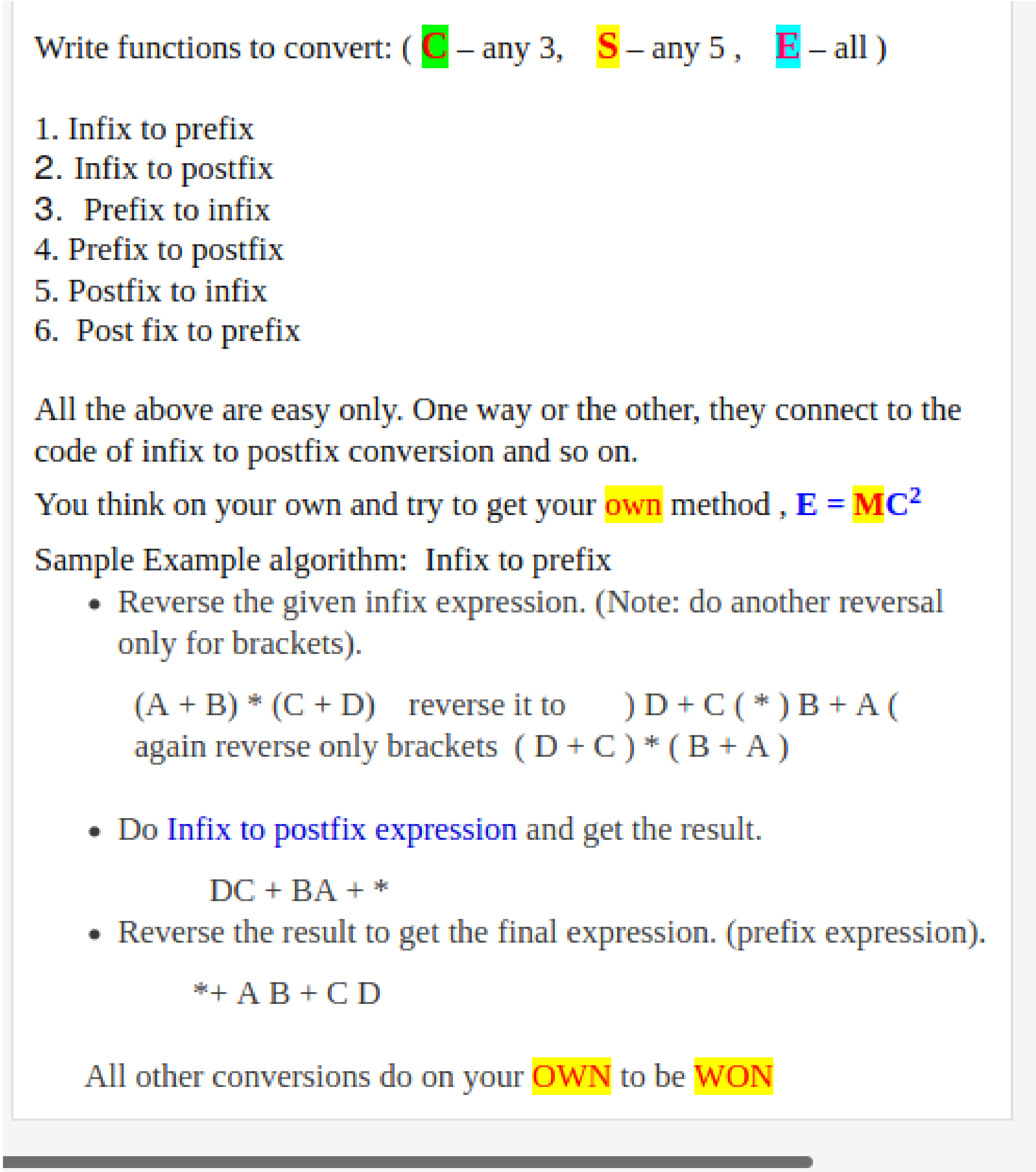
2. Iterative Quick sort

3. Minimum number of bracket(symbol) reversals needed to make an expression balanced, if the given input expression is not balanced.

4. Recursive postfix evaluation (without a stack)

5. Recursive function to convert prefix expression to postfix expression

Prefix : **+AB-+CDEF   postfix : AB+CD+E-*F*



Write functions to convert: (C – any 3, S – any 5 , E – all)

1. Infix to prefix
2. Infix to postfix
3. Prefix to infix
4. Prefix to postfix
5. Postfix to infix
6. Post fix to prefix

All the above are easy only. One way or the other, they connect to the code of infix to postfix conversion and so on.

You think on your own and try to get your own method , **E = MC^2**

Sample Example algorithm: Infix to prefix

- Reverse the given infix expression. (Note: do another reversal only for brackets).

  (A + B) * (C + D)   reverse it to   ) D + C (*) B + A (
  again reverse only brackets (D + C) * (B + A)

- Do Infix to postfix expression and get the result.

  DC + BA + *

- Reverse the result to get the final expression. (prefix expression).

  *+ A B + C D

All other conversions do on your OWN to be WON