Infix to prefix algorithm

27 April 2023 11:58

1) start scanning from right to left
2) If(current element is an operand)
   Append it to the prefix expression
3) Else if (current element is closing bracket ')')
   Push it onto the stack
4) Else if (current element is opening bracket '(')
   - Pop elements from the stack and append them to prefix exp till its corresponding closing brackets does not occur
   - Pop closing brackets from the stack and discard both the brackets

   Else
   // If(current element is an operator)
   While(stack is not empty && priority of topmost element > priority of current element)
   {
   Pop element from the stack and append it to prefix expression
   }
   Push current element onto the stack.
5) Repeat the above steps till end of infix expression
6) Pop all the remaining elements from the stack one by one and append them to prefix expression.
7) Reverse the prefix expression.

Infix expression :
5+9-4*(8-6/2)+1$(7-3)

Stack :

Current element :

Prefix Expression: 37-1$26/8-4*95+-+

Reverse: +-+59*4-8/62$1-73

$ → 10
* % / → 9
+ − → 8
default → 0

5+9-4*(8-6/2)+1$(7-3)

6 7 5 2 1 8 4 3

5 + 9 − 4 * (8 − 6 / 2) + 1 $ (7 − 3)
5 + 9 − 4 * (8 − /62) + 1 $ (7 − 3)
5 + 9 − 4 * −8/62 + 1 $ −73
5 + 9 − 4 * −8/62 + $1−73
5 + 9 − *4−8/62 + $1−73
+59 − *4−8/62 + $1−73
−+59*4−8/62 + $1−73
+−+59*4−8/62$1−73 → prefix

(→ ^ → $
* / %
+ −

+-+59*4-8/62$1-73

+-+59*4-8/62$1-73