**Infix to Postfix Algorithm**

Example: A+B/C\*(D+E\*F-G\*H)/I

1. Scan the infix expression from left to right.

2. If the scanned character is an operand, append to output

3. Else if the scanned character is an operator

a. If the precedence of the scanned operator is greater than the precedence of the operator in the top of the stack(or the stack is empty or the top is at ‘(‘ ), push it.

b. Else, Pop all the operators from the stack which are greater than or equal to in precedence than that of the scanned operator and append them to output. After doing that Push the scanned operator to the stack. (If you encounter parenthesis while popping then stop there and push the scanned operator in the stack.)

4. If the scanned character is an ‘(‘, push it to the stack.

5. If the scanned character is an ‘)’, pop the stack and append it to output until a ‘(‘ is encountered, and discard both the parenthesis.

6. Repeat steps 2-6 until infix expression is scanned.

7. Pop all operators from stack and append them to output

8. Print the output

Example: A+B/C\*(D+E\*F-G\*H)/I  
ch=A

|  |  |
| --- | --- |
| 7 |  |
| 6 |  |
| 5 |  |
| 4 |  |
| 3 |  |
| 2 |  |
| 1 |  |
| 0 |  |

Top=-1

Output=

Ex1: A+B\*C-(D-F)

Ex2: A^B\*C/(D\*E-F)