http://www.cinstitute.co.in
https://youtube.com/tarunsir
+917742817777

Infix to Postfix Conversion using Stack

1. **Initialize:**
   - Create an empty stack for operators.
   - Create an empty string for the postfix expression.
2. **Process Each Character:**
   - **Operands:** If the character is an operand (e.g., a variable or number), add it directly to the postfix expression.
   - **Operators:**
     - If the character is an operator (e.g., +, -, *, /):
       1. Pop operators from the stack and append them to the postfix expression **until the stack is empty or the operator at the top of the stack has less precedence** than the current operator.
       2. Push the current operator onto the stack.
   - **Parentheses:**
     - If the character is an opening parenthesis (, push it onto the stack.
     - If the character is a closing parenthesis), pop from the stack and append to the postfix expression **until an opening parenthesis** is encountered. Discard the opening parenthesis.
3. **End of Expression:**
   - After the infix expression has been processed, pop all remaining operators from the stack and append them to the postfix expression.
4. **Result:**
   - The postfix expression is now complete.

Operator Precedence Table

Operator	Precedence
(	1
**	2
*, /, %	3
+, -	4

Example: Infix to Postfix Conversion

Convert: (A + B) * C - D

Step	Stack	Postfix Expression
Read (	(	
Read A	(	A
Read +	(+	A
Read B	(+	A B
Read)		A B +
Read *	*	A B +
Read C	*	A B + C

http://www.cinstitute.co.in
https://youtube.com/tarunsir
+917742817777

Read -	-	A B + C *
Read D	-	A B + C * D
Pop Remaining		A B + C * D -