12. Convert infly expression to postfix expression

Infix notation when the operator is written in between the operands, then its known as infix notation. Eq: (P+q)x(Y+5)

post-fix notation—The post-fix expression is an expression in which the operator is written after the operands for example, the post-fix expression of inflx notation (2+3) can be written as 23+.

Infix to postfix Transformation

procedure

step 1: Scan the infin expression from left to right

Step 21 a) It the scanned symbol is left parenthesis, push it onto the stack.

b) If the scanned symbol is operand, then place directly in the postfix expression (output)

e) It the symbol scanned is a right paranthesis, then go on papping all the items from the stack and place them in the post-flx expression till we get the matching left parenthesis.

d) if the scanned symbol is an operator, then go on removing all the operators from the stack and place them in postfix expression, if and only if the precedence if the scanned operator which is on top of the stack is greater than cov equal) to the precedence of the scanned operator and push the sanged operator onto the stack otherwise, push the scanned operator onto the stack

Example:		
,	Ax (Bx c + Dx e) f	
convent To	oken stack	postfix string
		A superior and the state of the
2 +	·X	
s C	-X-C	A
4 B	· X C	AB
- *	·X.C-X	AB
6	-X C-X	ABC
1 +	·X (1	ABCX
8 D	***************************************	APLXID
9 *	XC+X	ABCXD
io E	XC 7X	ARCKDE
	A design of the second of the	ABCXDEX
2 +		ABCXDEXX
13 F	A TO A DESCRIPTION OF THE PROPERTY OF THE PROP	ABCX DETX F
4		ABOX DETX FT
IP	pression of Axla	5*c+D*E)+F
post-11x ex	PY(SSIOT) of ABCX	DC+XF+
	men en anti-chaptan proprieta de la continua de proprieta de la continua del la continua de la continua del la continua de la continua del la continua de la continua del la continua del la continua del la continua de	
The second section of the second section and the second section of the second section is the second section of the sec		

Code: def infly to post-fix (expression): Stack = [] output = 1) Operations 2 set ([41, -1, 1x1, 12, 12, 12, 12])

Priority = 2(4): 1, (-1: 1, 1x1: 2, 12: a, 12: a) for character in expression: it character not in operators: Output + = character elit character = (c): stack.append('c') elit character = = 12% While stack and stack [-1] 1= (c): output=stack.pop() Stack pop() else: while stack and stack [-1] = 'c' and priority (charater L= priority output += stack popc) Stack append (character) While stack: Output = Stack. POPC) veturn output expression=input ('Enter infix expression') print (inter notation: ', expression) print (postfix notation: infinto postfix (expression))

Output -

Enter infix expression: (A-B)*(DIE)
infix notation: (A-B)*(DIE)
postfix notation: AB-DEIX