

UNIT-2
Stack and Queue
Contd.....

TOPIC: Expression

Expression

- An expression consists of constants, variables, and symbols.
- Symbols can be operators or parenthesis.
- Examples of expressions are:

5+6,

A+B,

A-B,

A*B+C,

A/B*C,

(A+B)-C,

A+(B-C)*D, etc..

- An **Operand** is an data or a value on which the operation is to be performed.
- In the above expressions, **5, 6, A, B, C, D** are the operands while '+', '-', '*' and '/' are the **Operators**.

Types of expressions:

1. Infix expression
2. Prefix expression
3. Postfix expression

1. What is Infix expression?

- When the operator is written in between the operands, then it is known as **infix expression**.
 - For example: $a+b$, $10-5$, etc..
-

Note:

- Operand does not have to be always a constant or a variable it can also be an expression itself. For example, **$(p + q) * (r + s)$**
- In the above expression, both the expressions of the multiplication operator are the operands, i.e. **$(p + q)$** , and **$(r + s)$** are the operands.

Note:

- **While performing the operations on the expression, we need to follow some set of rules to evaluate the result.**
- If there is only one operator in the expression, we do not require applying any rule. For example, $10 + 2$; in this expression, addition operation can be performed between the two operands (10 and 2), and the result of the operation would be 12.
- If there are multiple operators in the expression, then some rule needs to be followed to evaluate the expression.
- If the expression is: **$4 + 6 * 2$** (If the plus operator is evaluated first, then the expression would look like: **$10 * 2 = 20$**)
- If the multiplication operator is evaluated first, then the expression would look like: **$4 + 12 = 16$**

- The above problem can be resolved by following the **operator precedence rules**.
- In the algebraic expression, the **order of the operator precedence** is given in the below table:

Operators	Symbol	Precedence	Associativity
Parenthesis	(), { }, []	5	Left to Right
Exponents	\wedge	4	Right to Left
Multiplication and Division	$*$, $/$	3	Left to Right
Addition and Subtraction	$+$, $-$	2	Left to Right

- The **first preference** is given to the **Parenthesis**.
- The **next preference** is given to the **Exponents**. In the case of multiple exponent operators, then the operation will be applied from **right to left**.
- After exponent, **multiplication, and division operators** are evaluated. If both the operators are present in the expression, then the operation will be applied from **left to right**.
- The next preference is given to **addition and subtraction**. If both the operators are available in the expression, then we go from **left to right**.

- If the operators have the same precedence, then we should follow the **associativity** rules.
- The operators that have the same precedence termed as **operator associativity**.
- If we go from left to right, then it is known as **left-associative**.
- If we go from right to left, then it is known as **right-associative**.

2. What is Prefix Expression?

- Prefix expression consists of operators followed by operands.
- For example:
 - The prefix expression of infix notation **(2+3)** can be written as **+23**
 - The prefix expression of infix notation **(2+3*4)** can be written as **+2*34**

Evaluation of prefix expression using stack.

Example 1:

Infix= $((4+3)*2)-5=9$

Prefix= $-*+4325$

Example: $-*+4325$

Symbol	opnd1	opnd2	value	opndstack
5				5
2				5, 2
3				5, 2, 3
4				5, 2, 3, 4
+	4	3	7	5, 2
				5, 2, 7
*	7	2	14	5
				5, 14
-	14	5	9	
				9

result

Example 2:

Infix= $(3*4) + (2*5)=22$

Prefix= $+*34*25$

Symbol	Operand 1	Operand 2	Value	Operand Stack
5				5
2				5,2
*	2	5	10	
				10
4				10,4
3				10,4,3
*	3	4	12	10
				10,12
+	12	10	22	
				22



Result

3. What is Postfix Expression?

- The postfix expression is an expression in which the operator is written after the operands.
- For example:
 - The postfix expression of infix notation **(2+3)** can be written as **23+**
 - The postfix expression of infix notation **(2+3*4)** can be written as **234*+**

Evaluation of postfix expression using stack.

- Scan the expression from left to right.
- If we encounter any operand in the expression, then we push the operand in the stack.
- When we encounter any operator in the expression, then we pop the corresponding operands from the stack.
- When we finish with the scanning of the expression, the final value remains in the stack.

- Let's understand the **evaluation of postfix expression** using stack.
- Example 1:**
 - Infix** = $(2+3*4)=14$
 - Postfix** = $2\ 3\ 4\ *\ +$

Input	Stack	
2 3 4 * +	empty	Push 2
3 4 * +	2	Push 3
4 * +	3 2	Push 4
* +	4 3 2	Pop 4 and 3, and perform $4*3 = 12$. Push 12 into the stack.
+	12 2	Pop 12 and 2 from the stack, and perform $12+2 = 14$. Push 14 into the stack.

The result of the above expression is 14.

Infix= $(3*4) + (2*5)=22$

Postfix= $34*25*+$

Example 2: Postfix expression: 3 4 * 2 5 * +

Input	Stack	
3 4 * 2 5 * +	empty	Push 3
4 * 2 5 * +	3	Push 4
* 2 5 * +	4 3	Pop 3 and 4 from the stack and perform $3*4 = 12$. Push 12 into the stack.
2 5 * +	12	Push 2
5 * +	2 12	Push 5
* +	5 2 12	Pop 5 and 2 from the stack and perform $5*2 = 10$. Push 10 into the stack.
+	10 12	Pop 10 and 12 from the stack and perform $10+12 = 22$. Push 22 into the stack.

The result of the above expression is 22.

#Conversion of Infix Expressions to Prefix and Postfix Expression:

1. Convert (a+b) into prefix and postfix expression.

Ans:

Given Infix= (a+b)

Infix= (a+b)

Prefix= +ab

Infix= (a+b)

Postfix= ab+

2. Convert $(2+3*4)$ into prefix and postfix expression.

Ans:

Given Infix= $(2+3*4)$

Infix= $(2+\underline{3}\underline{*}\underline{4})$

Prefix = $\underline{2} + \underline{(*34)}$
 = $\underline{+2 *34}$

Infix= $(2+\underline{3}\underline{*}\underline{4})$

Postfix = $\underline{2} + \underline{34*}$
 = $\underline{234*+}$

3. Convert $(3*4) + (2*5)$ into prefix and postfix expression.

Ans:

Given Infix= $(3*4) + (2*5)$

Infix= $(\underline{3}*\underline{4}) + (2*5)$

Prefix = $(\underline{*34}) + (\underline{2*5})$
 = $(\underline{*34}) + (\underline{*25})$
 = $\underline{+*34*25}$

Infix= $(\underline{3}*\underline{4}) + (2*5)$

Postfix = $(\underline{34*}) + (\underline{2*5})$
 = $(\underline{34*}) + (\underline{25*})$
 = $\underline{34*25*+}$

4. Convert $A * B + C / D$ into prefix and postfix expression.

Ans:

Given Infix= $A * B + C / D$

Infix= A * B + C / D

Prefix = *AB + C / D
 = *AB + /CD
 = +*AB/CD

Infix= A * B + C / D

Postfix = AB* + C / D
 = AB* + CD/
 = AB*CD/+

5. Convert $A+B*C/(E-F)$ into prefix and postfix expression.

Ans:

$$\text{Infix} = A + B * C / (E - F)$$

$$\text{Prefix} = A + \underline{B} * \underline{C} / \underline{-EF}$$

$$= A + \underline{*BC} / \underline{-EF}$$

$$= \underline{A} + \underline{/ * BC - EF}$$

$$= \underline{+ A / * BC - EF}$$

$$\text{Infix} = A + B * C / (E - F)$$

$$\text{Postfix} = A + \underline{B} * \underline{C} / \underline{EF -}$$

$$= A + \underline{BC *} / \underline{EF -}$$

$$= \underline{A} + \underline{BC * EF - /}$$

$$= \underline{ABC * EF - / +}$$

6. Convert $(A - B/C) * (A/K - L)$ into prefix and postfix expression.

Ans:

$$\text{Infix} = (A - \underline{B/C}) * (A/K - L)$$

$$\text{Prefix} = (\underline{A} - \underline{/BC}) * (A/K - L)$$

$$= \underline{-A/BC} * (\underline{A/K} - L)$$

$$= \underline{-A/BC} * (\underline{/AK} - \underline{L})$$

$$= \underline{-A/BC} * \underline{-/AKL}$$

$$= \underline{* -A/BC - /AKL}$$

$$\text{Infix} = (A - \underline{B/C}) * (A/K - L)$$

$$\text{Postfix} = (\underline{A} - \underline{BC/}) * (A/K - L)$$

$$= \underline{ABC/-} * (\underline{A/K} - L)$$

$$= \underline{ABC/-} * (\underline{AK/} - L)$$

$$= \underline{ABC/-} * \underline{AK/L-}$$

$$= \underline{ABC/-AK/L-*}$$

7. Convert $((A*B)+(C/D))$ into prefix and postfix expression.

Ans:

$$\text{Infix} = ((\underline{A} * \underline{B}) + (C/D))$$

$$\text{Prefix} = (\underline{*AB} + (C/D))$$

$$= (\underline{*AB} + \underline{/CD})$$

$$= \underline{+*AB/CD}$$

$$\text{Infix} = ((\underline{A} * \underline{B}) + (C/D))$$

$$\text{Postfix} = (\underline{AB*} + (C/D))$$

$$= (\underline{AB*} + \underline{CD/})$$

$$= \underline{AB*CD/+}$$

8. Convert $((A*(B+C))/D)$ into prefix and postfix expression.

Ans:

$$\text{Infix} = ((A * (B + C)) / D)$$

$$\text{Prefix} = ((A * +BC) / D)$$

$$= (\underline{*A+BC} / \underline{D})$$

$$= \underline{/ * A + B C D}$$

$$\text{Infix} = ((A * (B + C)) / D)$$

$$\text{Postfix} = ((A * BC +) / D)$$

$$= (\underline{ABC + *} / \underline{D})$$

$$= \underline{ABC + * D /}$$

9. Convert $(A*(B+(C/D)))$ into prefix and postfix expression.

Ans:

$$\text{Infix} = (A * (B + (C/D)))$$

$$\text{Prefix} = (A * (B + \underline{C/D}))$$

$$= (\underline{A} * \underline{+B/C/D})$$

$$= \underline{*A+B/C/D}$$

$$\text{Infix} = (A * (B + (C/D)))$$

$$\text{Postfix} = (A * (B + \underline{C/D}))$$

$$= (\underline{A} * \underline{BCD/+})$$

$$= \underline{ABCD/+*}$$

10. Convert $A+(B*C-(D/E-F)*G)*H$ into prefix and postfix expression.

$$\text{Infix} = A + (B * C - (D / E - F) * G) * H$$

$$\begin{aligned} \text{Prefix} &= A + (B * C - (/ DE - F) * G) * H \\ &= A + (\underline{B} * \underline{C} - \underline{- / DEF} * G) * H \\ &= A + (\underline{* BC} - \underline{- / DEF} * G) * H \\ &= A + (\underline{* BC} - \underline{* - / DEF G}) * H \\ &= A + \underline{- * BC * - / DEF G} * H \\ &= \underline{A + * - * BC * - / DEF G H} \\ &= \underline{+ A * - * BC * - / DEF G H} \end{aligned}$$

$$\text{Infix} = A + (B * C - (D / E - F) * G) * H$$

$$\begin{aligned} &= A + (B * C - (\underline{DE /} - F) * G) * H \\ &= A + (\underline{B} * \underline{C} - \underline{DE / F -} * G) * H \\ &= A + (\underline{BC * } - \underline{DE / F -} * G) * H \\ &= A + (\underline{BC * } - \underline{DE / F - G * }) * H \\ &= A + \underline{BC * DE / F - G * -} * H \\ &= \underline{A + BC * DE / F - G * - H * } \\ &= \underline{ABC * DE / F - G * - H * +} \end{aligned}$$

11. Convert $(A+B^{\wedge}C)*D+E^{\wedge}5$ into prefix and postfix expression.

Ans:

$$\text{Infix} = (A + \underline{B}^{\wedge}\underline{C}) * D + E^{\wedge}5$$

$$\text{Prefix} = (\underline{A} + \underline{ABC}) * D + E^{\wedge}5$$

$$= \underline{+A^{\wedge}BC} * D + \underline{E^{\wedge}5}$$

$$= \underline{+A^{\wedge}BC} * \underline{D} + \underline{E^{\wedge}5}$$

$$= \underline{*+A^{\wedge}BCD} + \underline{E^{\wedge}5}$$

$$= \underline{+*+A^{\wedge}BCD^{\wedge}E5}$$

$$\text{Infix} = (A + \underline{B}^{\wedge}\underline{C}) * D + E^{\wedge}5$$

$$\text{Postfix} = (\underline{A} + \underline{BC^{\wedge}}) * D + E^{\wedge}5$$

$$= \underline{ABC^{\wedge}} + * D + \underline{E^{\wedge}5}$$

$$= \underline{ABC^{\wedge}} + * D + \underline{E5^{\wedge}}$$

$$= \underline{ABC^{\wedge} + D * } + \underline{E5^{\wedge}}$$

$$= \underline{ABC^{\wedge} + D * E5^{\wedge} + }$$

ASSIGNMENT

Q. NO. 12

Q. NO. 13

Q. NO. 14

12. Convert $(a+b-c)*(d-e)/(f-g+h)$ into prefix and postfix expression.

Ans:

Prefix= $/*-+abc-de+-fgh$

Postfix= $ab+c-de-*fg-h+ /$

13. Convert $(A+B) * C/D + E ^ F/G$ into prefix and postfix expression.

Ans:

Prefix= $+/*+ABCD/ ^ EFG$

Postfix= $AB* C*D / E F^G / +$

14. Convert $K + L - M * N + (O^P) * W/U/V * T + Q$ into prefix and postfix expression.

Ans:

Prefix= $++-+KL*MN*//*^OPWUVTQ$

Postfix= $K L + M N *- OP^ W*U/V / T *+ Q +$

#Conversion of Infix Expressions to Prefix and Postfix Expression using stack

1. Convert $K + L - M * N + (O^P) * W / U / V * T + Q$ into prefix and postfix expression using stack.

See this video For **infix to prefix** conversion

<https://www.youtube.com/watch?v=8QxlrRws9OI>

See this video For **infix to postfix** conversion

<https://www.youtube.com/watch?v=PAceaOSnxQs>