Monsoon-2023

Infix to Postfix

CS2002D Program Design

Operator Precedence in C

Category	Operator	Associativity
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - -	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR	|	Left to right
Logical AND	&&	Left to right
Logical OR	||	Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=>>= <<= &= ^= |=	Right to left
Comma	,	Left to right

Algorithm

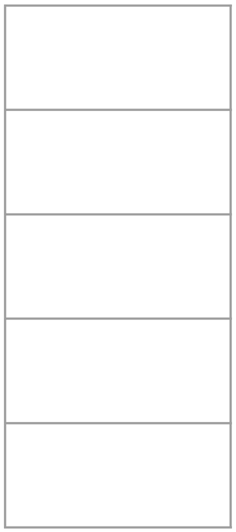
Scan the infix characters from left to right. Initialize an empty stack to hold operators and parentheses. Initialize the postfix expression to ε (empty string).

- If the character is an operand, then immediately append it to the postfix expression.
- If the character is left parenthesis ‘(‘, then Push it onto the Stack.
- If the character is right parenthesis ‘)’, then Pop all the contents of the stack until the respective left parenthesis is popped and append each popped symbol to the postfix expression.
- If the character is an operator (+, –, *, /), then Push it onto the Stack. However, first, pop the operators which are already on the stack that have **higher** or **equal** precedence than the current operator and append them to the postfix expression. If an open parenthesis is there on top of the stack then push the operator into the stack.
- If the input is over, pop all the remaining symbols from the stack and append them to the

Example

Infix Expression

a + b * c



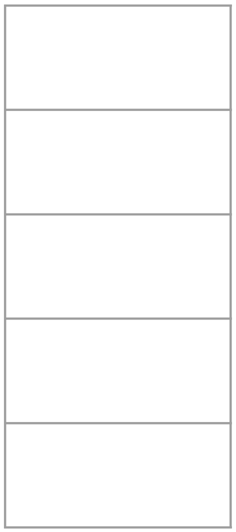
Stack

Postfix Expression

Example

Infix Expression

a + b * c



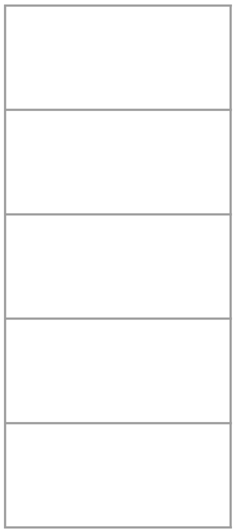
Stack

Postfix Expression

Example

Infix Expression

a + b * c



Stack

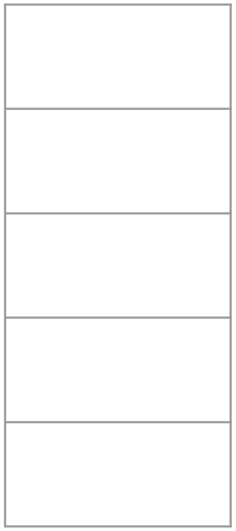
Postfix Expression

a

Example

Infix Expression

a **+** b * c



Stack

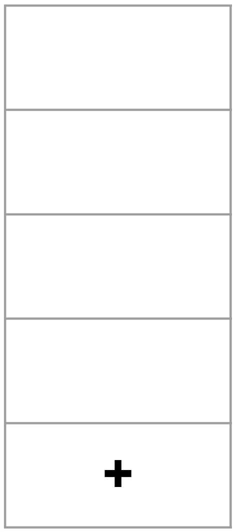
Postfix Expression

a

Example

Infix Expression

a + b * c



+

Stack

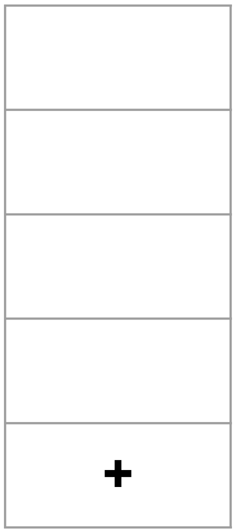
Postfix Expression

a

Example

Infix Expression

a + b * c



Stack

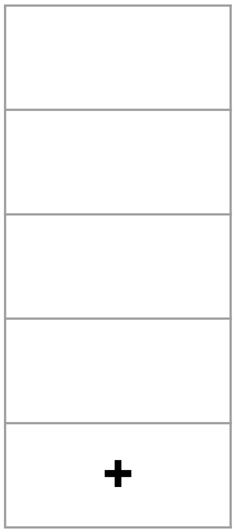
Postfix Expression

a

Example

Infix Expression

a + b * c



Stack

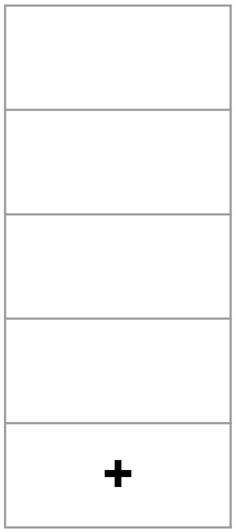
Postfix Expression

ab

Example

Infix Expression

a + b * c



Stack

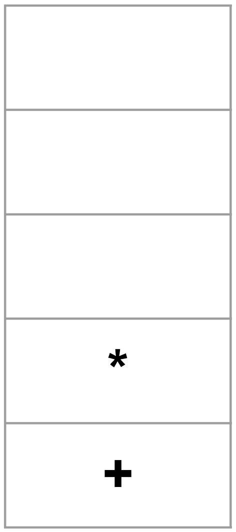
Postfix Expression

ab

Example

Infix Expression

a + b * c



*

+

Stack

Postfix Expression

ab

Example

Infix Expression

a + b * c

*
+

Stack

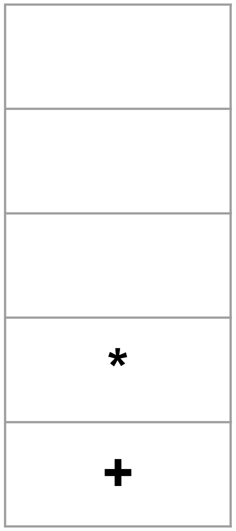
Postfix Expression

ab

Example

Infix Expression

a + b * c



Stack

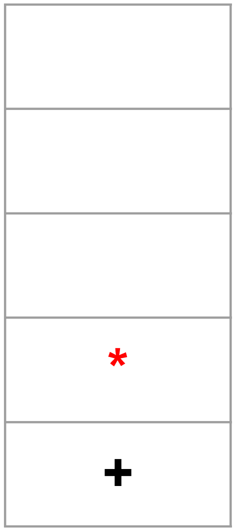
Postfix Expression

abc

Example

Infix Expression

a + b * c



Stack

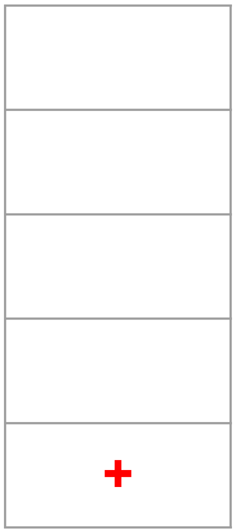
Postfix Expression

abc*

Example

Infix Expression

a + b * c



Stack

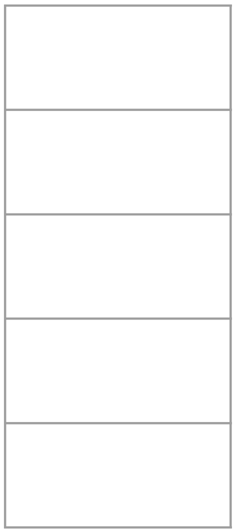
Postfix Expression

abc*+

Example

Infix Expression

a + b * c



Stack

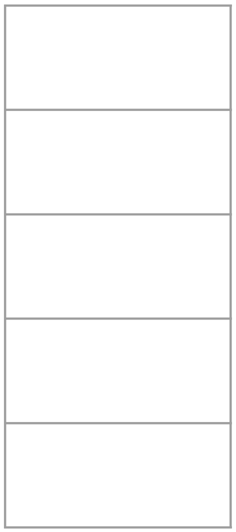
Postfix Expression

abc*+

Example 2

Infix Expression

a * b + c



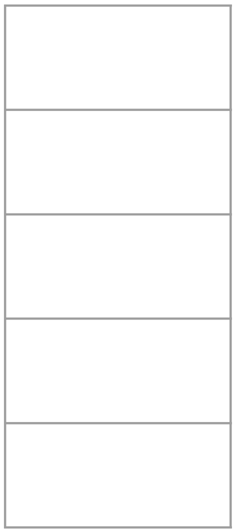
Stack

Postfix Expression

Example 2

Infix Expression

a * b + c



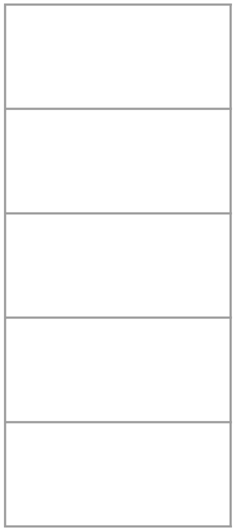
Stack

Postfix Expression

Example 2

Infix Expression

a * b + c



Stack

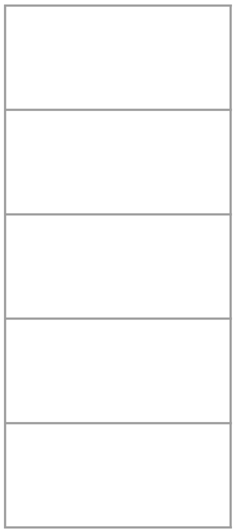
Postfix Expression

a

Example 2

Infix Expression

a * b + c



Stack

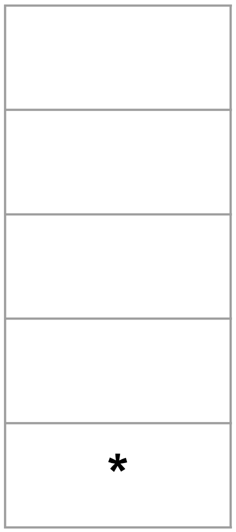
Postfix Expression

a

Example 2

Infix Expression

a * b + c



Stack

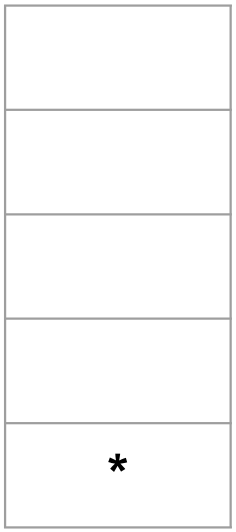
Postfix Expression

a

Example 2

Infix Expression

a * b + c



Stack

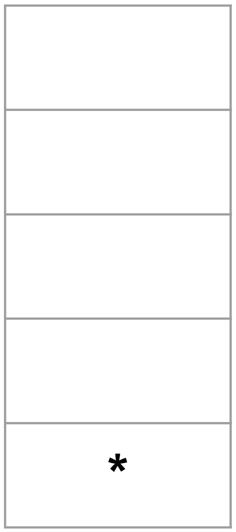
Postfix Expression

a

Example 2

Infix Expression

a * b + c



Stack

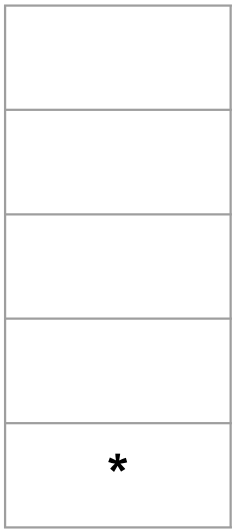
Postfix Expression

ab

Example 2

Infix Expression

a * b + c



Stack

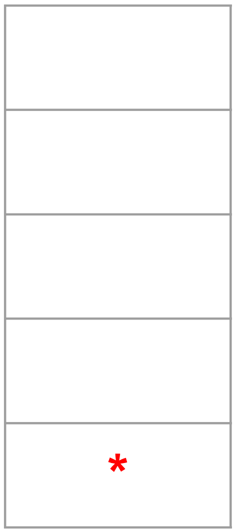
Postfix Expression

ab

Example 2

Infix Expression

a * b + c



Stack

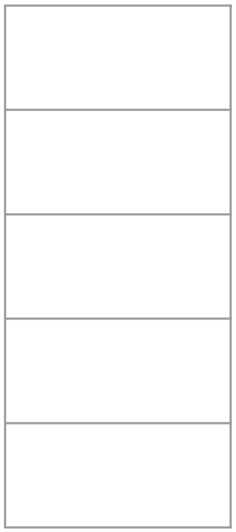
Postfix Expression

ab

Example 2

Infix Expression

a * b + c



Stack

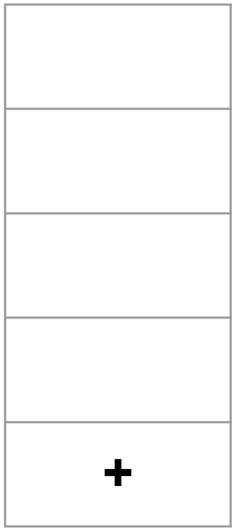
Postfix Expression

ab*

Example 2

Infix Expression

a * b + c



Stack

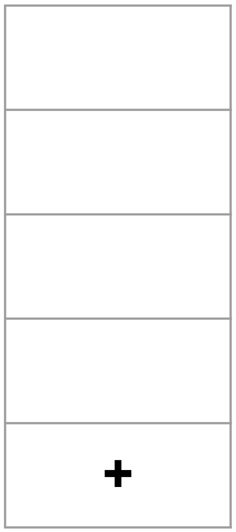
Postfix Expression

ab*

Example 2

Infix Expression

a * b + c



+

Stack

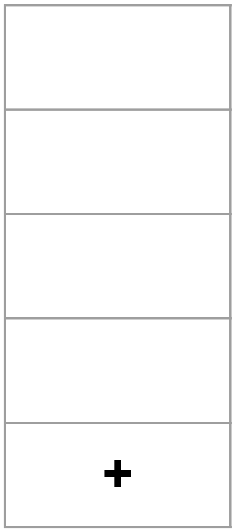
Postfix Expression

ab*

Example 2

Infix Expression

a * b + c



Stack

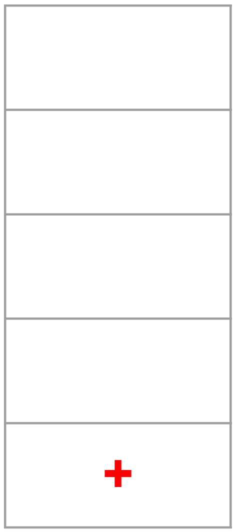
Postfix Expression

ab*c

Example 2

Infix Expression

a * b + c



Stack

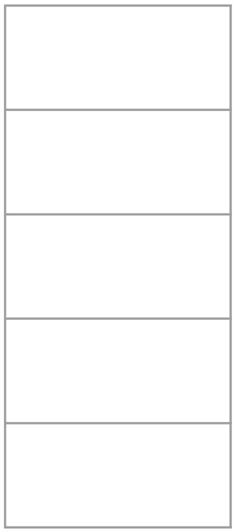
Postfix Expression

ab*c+

Example 2

Infix Expression

a * b + c



Stack

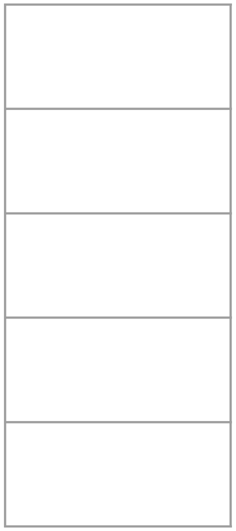
Postfix Expression

ab*c+

Example 3

Infix Expression

a * (b + c)



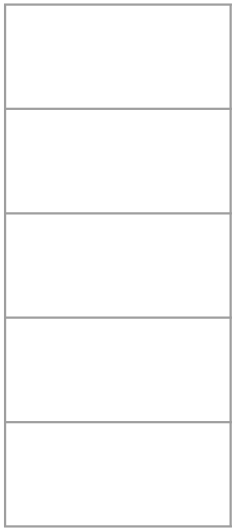
Stack

Postfix Expression

Example 3

Infix Expression

a * (b + c)



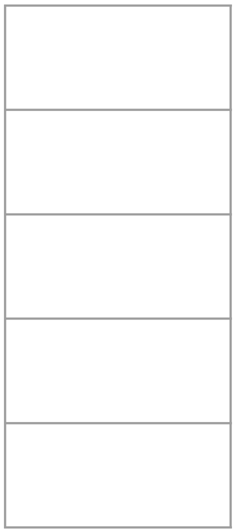
Stack

Postfix Expression

Example 3

Infix Expression

a * (b + c)



Stack

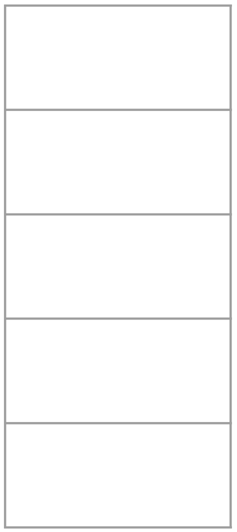
Postfix Expression

a

Example 3

Infix Expression

a * (b + c)



Stack

Postfix Expression

a

Example 3

Infix Expression

a * (b + c)

*

Stack

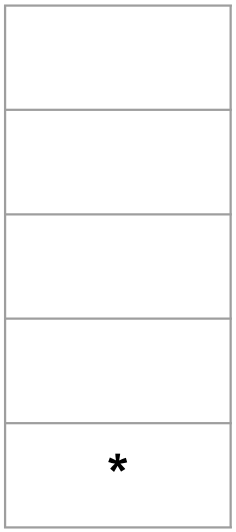
Postfix Expression

a

Example 3

Infix Expression

a * **(** b + c)



Stack

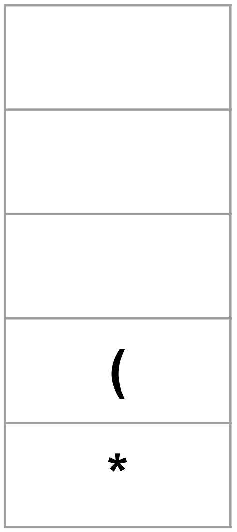
Postfix Expression

a

Example 3

Infix Expression

a * (b + c)



Stack

Postfix Expression

a

Example 3

Infix Expression

a * (b + c)

(
*

Stack

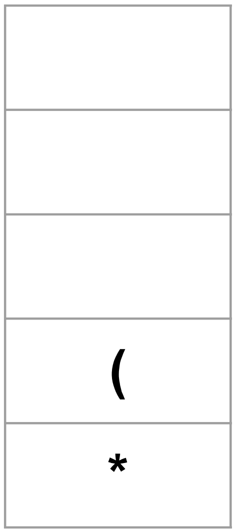
Postfix Expression

a

Example 3

Infix Expression

a * (b + c)



(
*

Stack

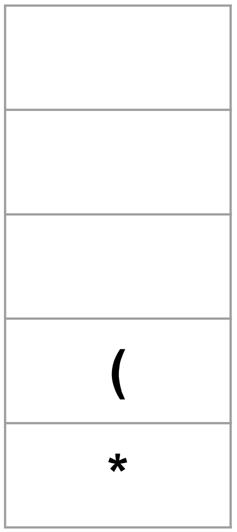
Postfix Expression

ab

Example 3

Infix Expression

a * (b + c)



(

*

Stack

Postfix Expression

ab

Example 3

Infix Expression

a * (b + c)

+
(
*

Stack

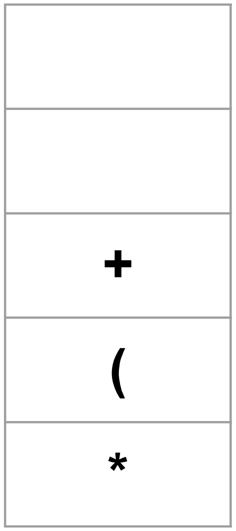
Postfix Expression

ab

Example 3

Infix Expression

a * (b + c)



+
(
*

Stack

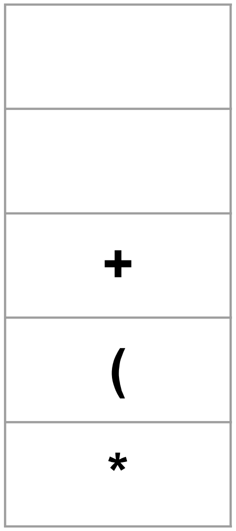
Postfix Expression

ab

Example 3

Infix Expression

a * (b + c)



Stack

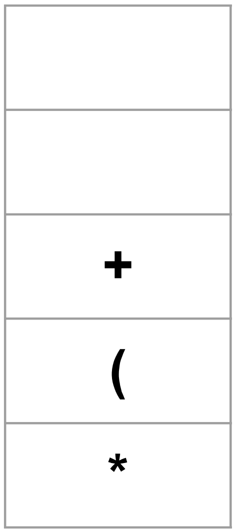
Postfix Expression

abc

Example 3

Infix Expression

a * (b + c)



Stack

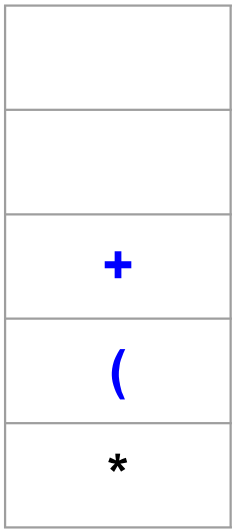
Postfix Expression

abc

Example 3

Infix Expression

a * (b + c)



Stack

Postfix Expression

abc

Example 3

Infix Expression

a * (b + c)

(
*

Stack

Postfix Expression

abc+

Example 3

Infix Expression

a * (b + c)

Postfix Expression

abc+

*

Stack

Example 3

Infix Expression

a * (b + c)

*

Stack

Postfix Expression

abc+

Example 3

Infix Expression

a * (b + c)

*

Stack

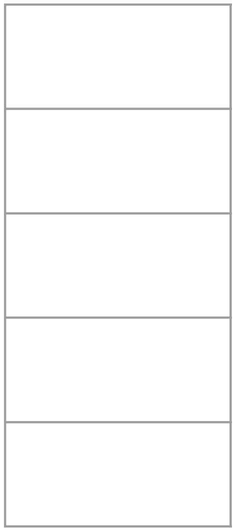
Postfix Expression

abc+*

Example 3

Infix Expression

a * (b + c)



Stack

Postfix Expression

abc+*

Practice Question

Infix Expression

a + ((b * c) + (b / d)) / (a + c - e)

Postfix Expression

?

Practice Question

Infix Expression

a + ((b * c) + (b / d)) / (a + c - e)

Postfix Expression

abc* bd/ + ac+e- / +

Thank You !!!