

# Expression Evaluation & Recursion in Data Structure

Course on C-Programming & Data Structures: GATE - 2024 & 2025

# Data Structure Stack: Expressions & Recursion

By: Vishvadeep Gothi

# Applications of Stack

- ◆ Expression Evaluation
- ◆ Recursion

# Expressions

- ◆ **Prefix Notation:** When operator is placed before both of operands
- ◆ **Infix Notation:** When operator is placed between both of operands
- ◆ **Postfix Notation:** When operator is placed after both of operands

# Precidence and Associativity

## ◆ **Precedence:** High to Low

- |                              |   |            |
|------------------------------|---|------------|
| 1. Parenthesis               | : | ( )        |
| 2. Exponent                  | : | $\uparrow$ |
| 3. Multiplication & Division | : | * & /      |
| 4. Addition & Subtraction    | : | + & -      |

# Infix to Prefix & Postfix

2 + 3 \* 5

# Infix to Prefix & Postfix

2 + 5 \* 3 ↑ 2 + 9

# Infix to Prefix & Postfix

2 - 5 \* 1 / 6 ↑ 3 ↑ 2 + 9

# Infix to Prefix & Postfix

$-b$

$\log x$

$x!$

$\log x!$

Postfix    to    Infix

$a b + c * d /$

$(a+b) c * d /$

$((a+b) * c) d /$

$(a+b) * c / d$

$(a+b) * c / d$

# Postfix to Infix

2 5 3 2<sup>1</sup>\* + 9 +

2 5 (3<sup>1</sup>) \* + 9 +

2 (5 \* 3<sup>1</sup>) + 9 +

(2 + 5 \* 3<sup>1</sup>) 9 +

2 + 5 \* 3<sup>1</sup> 9 +

Ques)

2 3 5 ↑ 9 ↑ + 6 \*

2 (3↑5) 9 ↑ + 6 \*

2 ((3↑5)↑9) + 6 \*

(  
2 + ((3↑5)↑9))  
6 \*

( 2 + (3↑5)↑9 ) \* 6 ↘

Ques) Postfix  $\rightarrow$  prefix

$a b * c +$

$\frac{* a b}{+} \quad c +$

$+ * a b c$

~~Infix~~

$a * b + c$

$+ * a b c$

Ques)

$$a b c * d / + e -$$

$$a \quad \underline{* b c} d / + e -$$

$$a \quad \underline{/ * b c d} + e -$$

$$\underline{+ a / * b c d} \quad e -$$

$$- + a / * b c d e$$

Prefix to Infix :-

$$+ * a b c$$

$$+ (a * b) c$$

$$\{a * b\} + c$$

$$a * b + c$$

$$+ d + * b c a$$

$$+ d + (b * c) a$$

$$+ d \left( \cancel{(b * c)} + a \right)$$

$$\boxed{d + (b * c + a)}$$

Ans

$$d + e * c + a$$

$$a + b + c$$

# Prefix to Infix

$- + 9 / 4 \uparrow 2 3 8$

$- + 9 / \{ (2 \uparrow 3) 8$

$- + 9 (4 / \{ 2 \uparrow 3 \}) 8$

$- + 9 (4 / 2 \uparrow 3) 8$

$- (9 + \{ 4 / 2 \uparrow 3 \}) 8$

$- (9 + 4 / 2 \uparrow 3) 8$

$\{ 9 + 4 / 2 \uparrow 3 \} - 8$

$9 + 4 / 2 \uparrow 3 - 8$

# Question

a b c \* + d e / f \* -

+ - 6 ↑ 2 \* 3 7 9

convert given expressions to infix

①

a + b \* c - d / e \* f

②

6 - 2 ↑ (3 \* 7) + 9

prefix to postfix:-

+ - e \* d / \* a b c x

e d q b \* c / \* - x +

# Evaluation of Postfix Notation Using Stack

~~2 5 3 2 1 \* + 9 / +~~



$$\frac{\text{for } \uparrow : -}{a = 2}$$
$$b = 3$$

$$b \uparrow q = 3 \uparrow 2 = 9$$

~~for \* :-~~

$$a = 9$$
$$b = 5$$

$$5 * 9 = 45$$

~~for + :-~~

$$a = 45$$
$$b = 2$$

$$2 + 45 = 47$$

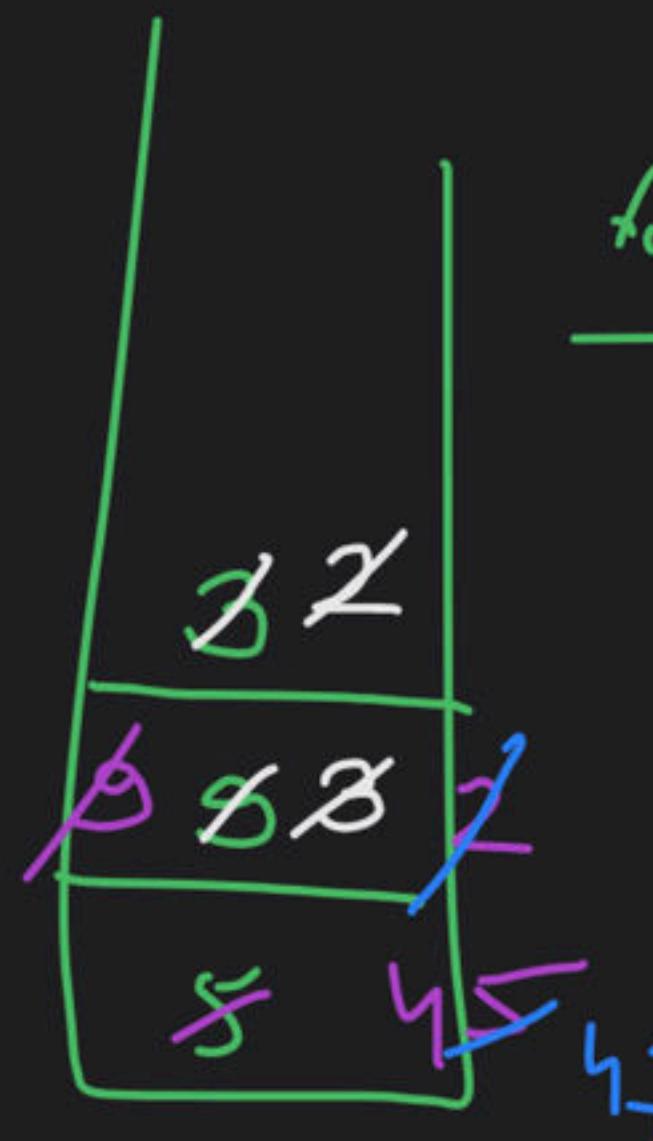
~~for + :-~~

$$a = 9$$
$$b = 42$$

$$47 + 9 = 56$$

ans)

$$5 - \frac{9}{3} / 2 \uparrow * 2 +$$



for / :-

$$\begin{aligned} a &= 3 \\ b &= 9 \end{aligned}$$

$$9/3 = 3$$

for ↑ :-

$$\begin{aligned} a &= 2 \\ b &= 3 \end{aligned}$$

$$3 \uparrow 2 = 9$$

for do :-

$$\begin{aligned} a &= 9 \\ b &= 5 \end{aligned}$$

$$5 * 9 = 45$$

for + :-

$$\begin{aligned} a &= 3 \\ b &= 45 \end{aligned}$$

$$45 + 2 = 47$$

Ans = 47

# Evaluation of Postfix Notation Using Stack

1. Add right parenthesis ) at the end of  $P$
2. Scan  $P$  from left to right until ) is encountered:
  - I. If an Operand is encountered, PUSH it onto stack
  - II. If an operator is encountered:
    - A. POP first two elements from stack , a is top element and b is next to top element
    - B. Evaluate  $b \text{ op } a$  & push the result onto stack
3. Set the result = top of stack

# Evaluation of Prefix Notation Using Stack

1. Add right parenthesis ( at the start of  $P$
2. Scan  $P$  from right to left until ( is encountered:
  - I. If an Operand is encountered, PUSH it onto stack
  - II. If an operator is encountered:
    - A. POP first two elements from stack , a is top element and b is next to top element
    - B. Evaluate  $a \text{ op } b$  & push the result onto stack
3. Set the result = top of stack

# Question GATE-2007

The following postfix expression with single digit operand is evaluated using a stack

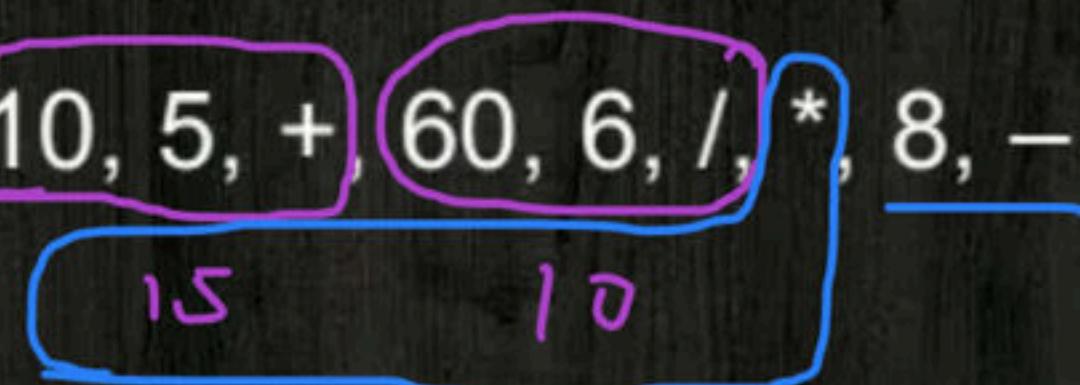
8 2 3 ^ / 2 3 \* + 5 1 \* -

Note that  $\wedge$  is the exponentiation operator. The top elements of the stack after the first  $*$  is evaluated are



# Question GATE-2015

The result evaluating the postfix expression  $10, 5, +, 60, 6, /, *, 8, -$  is



$$\text{Ans} = 142$$

$$\begin{array}{r} 150 \\ 10 \\ \hline 142 \end{array}$$

Consider given postfix expression

2 3 - 6 3 + \*

No. of push and pop operations performed to evaluate this expression using stack?

Sol :- For each operand  $\Rightarrow$  1 push  
4 operands  $\Rightarrow$  4 push

for each operator  $\Rightarrow$  2 pop  
3 operators  $\Rightarrow$  6 pop, 3 push

Total Push = 7  
Pop = 6

Ques) consider a postfix expression with  $n$  binary operators.  
Total no. of PUSH & POP operations to evaluate this  
expression using stack is ?.

---

Sol) if  $n$  binary operators,  $n+1$  operands  
 $n+1$  operands  $\Rightarrow n+1$  PUSH      total,  
 $n$  operators  $\Rightarrow 2n$  POP      PUSH  $\Rightarrow 2^{n+1}$   
   $n$  PUSH      POP  $\Rightarrow 2^n$

# Fibonacci Series

The Fibonacci series is a sequence of numbers where each number is the sum of the previous two.

The first few numbers in the series are:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

...and so on, continuing indefinitely.

The series is named after the Italian mathematician Leonardo Fibonacci, who introduced it to the Western world in his book *Liber Abaci*.

The Fibonacci series has many interesting properties and applications in mathematics, science, and nature.

For example, the ratio of consecutive terms in the series approaches the golden ratio, approximately 1.618, as the terms get larger.

The series also appears in the growth patterns of many plants, such as the arrangement of leaves on a stem or the branching of trees.

In addition, the Fibonacci series is used in various fields of science, such as physics, chemistry, and biology, to model complex systems and phenomena.

The series is also used in finance and economics to predict market trends and price movements.

Overall, the Fibonacci series is a fascinating mathematical concept with far-reaching applications in many fields.

# Fibonacci Series

$$fib(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ fib(n - 1) + fib(n - 2) & n > 1 \end{cases}$$

# Fibonacci Series

$$fib(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ fib(n - 1) + fib(n - 2) & n > 1 \end{cases}$$

Calculate the no of invocations in  $fib(8)$

Calculate the no of additions in  $fib(9)$

# Question

Fill in the blank:

$$pow(x, n) = \begin{cases} 1, & n = 0 \\ 0, & x = 0 \\ x, & n = 1 \\ x * pow(x, n - 1), & n > 0 \\ \frac{1}{x} * pow(x, \underline{\hspace{2cm}}), & n < 0 \end{cases}$$

# Tower of Hanoi

by [John Doe](#)

Published on [2023-01-01](#)

Estimated reading time: 5 minutes

Tags: [algorithm](#), [recursion](#), [problem-solving](#)

Comments: 0 | Likes: 0

Share: [Facebook](#) | [Twitter](#) | [LinkedIn](#)

Report abuse

Get in touch with the author

Leave a comment

View all posts by John Doe

Follow John Doe on [Medium](#)

Follow John Doe on [GitHub](#)

Follow John Doe on [LinkedIn](#)

Follow John Doe on [Twitter](#)

Follow John Doe on [Facebook](#)

Follow John Doe on [Instagram](#)

Follow John Doe on [YouTube](#)

Follow John Doe on [Medium](#)

Follow John Doe on [GitHub](#)

Follow John Doe on [LinkedIn](#)

Follow John Doe on [Twitter](#)

Follow John Doe on [Facebook](#)

Follow John Doe on [Instagram](#)

Follow John Doe on [YouTube](#)

Follow John Doe on [Medium](#)

Follow John Doe on [GitHub](#)

Follow John Doe on [LinkedIn](#)

Follow John Doe on [Twitter](#)

Follow John Doe on [Facebook](#)

Follow John Doe on [Instagram](#)

Follow John Doe on [YouTube](#)

Follow John Doe on [Medium](#)

# Tower of Hanoi

Answer the following questions when there are n=3 disks:

1. How many disk moves?
2. How many total function invocation
3. After how many invocations the first move of the disk is made?
4. After how many invocations the last move of the disk is made?

---

# Happy Learning



---