GATE

CRASH COURSE

Data Science & AI

Subject

Data Structure & Algorithms
Applications Of Stack
Lec No. - 01

By – Satya Sir

Topics to be covered

1. Data Structure?, Classification
2. Stack Operations
3. Applications Of Stack
4. Examples

Data Structure?

1 Data Structure :

- The way of organizing data
  - → Management / Supervision : It defines how operations are Performed on data
    - Create
    - Add / Insert
    - Delete / Modify
    - Access / Traverse
    - Search
    - Sort

(OR)

- The Representation of data

Classification Of DS

Data Structures

- Linear Data Structures (Single-level)
  - Arrays (Lists), Stack, Queue, Hash Tables, Linked Lists
- Non-Linear Data Structures (Multiple-levels)
  - Trees, Graphs, Sets, Maps

Stack

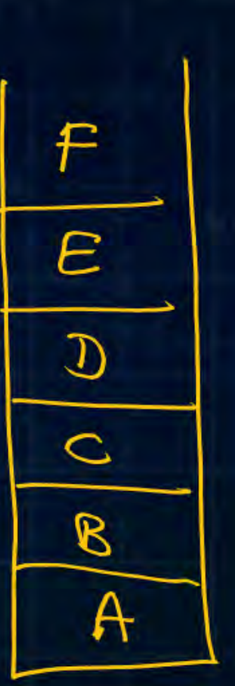
- A Linear Data Structure, which Permits, Insertion, Deletion, Access operations from only one End of the list, called Top of stack.

- In Stack, Insertion operation == Push operation
  Deletion operation == Pop operation
  Access operation == Peek operation

- Stack organization (or) DS is said to be Last-In-First-Out (LIFO) organization / DS.

Push sequence :
A → B → C → D → E → F

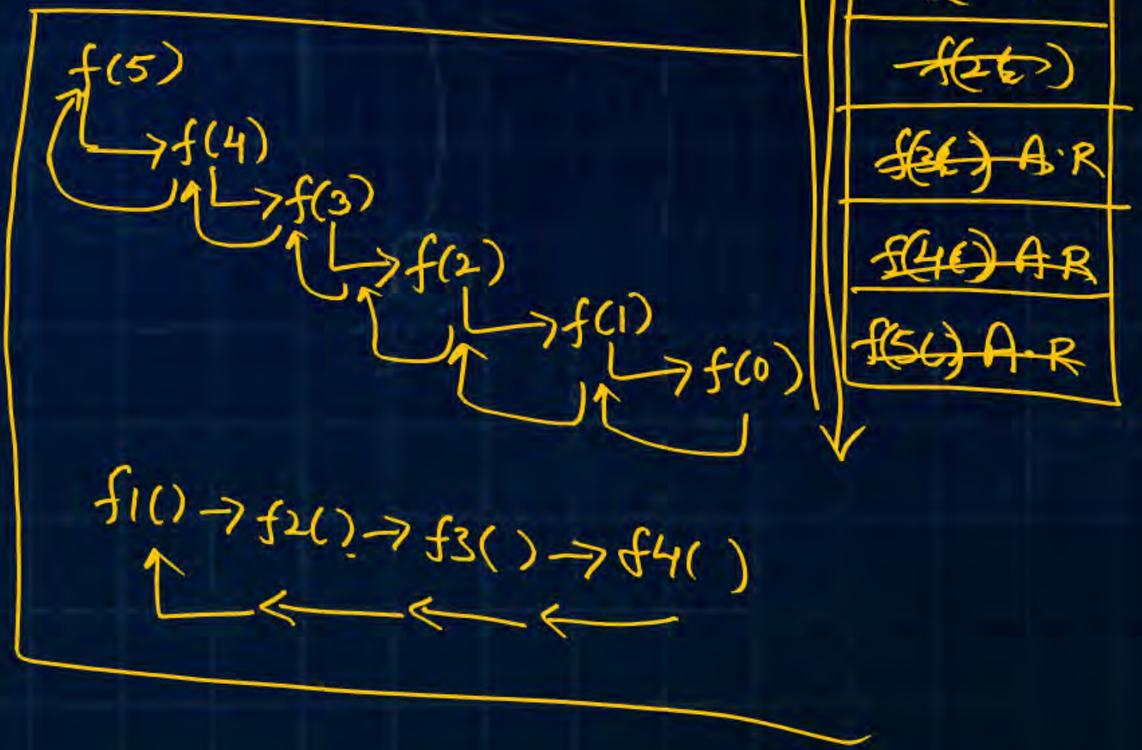
Pop sequence :
F → E → D → C → B → A



Stack

Applications of Stack :

1) Undo operation
2) Browsing history organization
3) Recursion Implementation
4) Function Implementation
5) Interrupt service (Non-Maskable)
6) Backtracking design Implementation
7) Graph Traversals, } Depth-First Traversal
8) Tree Traversals }
9) Expression Conversion
10) Expression Evaluation
11) To check balance of Paranthesis
⋮



Stack

Expression : Any Executable Statement

- → Computational
- → I/P
- → O/P
- → Initialization
- → definition
- → Assignment
- ⋮

→ Arithmetic Expression : A Statement that uses Arithmetic operators (+, −, *, /, ^) on operands.

^ → Exponentiation

→ It Can be Expressed in either of 3 forms :

1) Infix Notation : Operand1 operator Operand 2 Ex: a + b
2) Prefix Notation : operator operand1 Operand 2 Ex: + a b
3) Postfix Notation : Operand1 Operand2 operator Ex: a b +

Stack

Infix to Postfix Conversion

(iii) If $OP_t >= OP_s$, Then Pop OP_t, write in Y, Push OP_s.

3.D) If scanned Element == ')', Keep Popping from Stack and Put it Y until first '(' is encountered; ignore ( ).

4) Repeat step ② and ③ until 'X' is scanned.

5) Once 'X' is completely scanned, make stack Empty. 'Y' is result.

Procedure :

1) Let 'X' be Infix Expression, 'Y' be Equivalent Postfix Expression, 'S' be an Empty stack.

2) Scan 'X' from Left to Right one Element at a time.

3) A) If Scanned Element == Operand, Put it in Y.

B) If Scanned Element == '(', Push it on to Stack, S

C) If Scanned Element == operator (OP_s). Then, Compare it with Top of stack (OP_t)

i) If OP_t == '(', Then Push OP_s

ii) If Stack is Empty, Push OP_s

default Precedence :

① $\wedge$ | ② *, / | ③ +, -

Stack

Example 1: The Postfix Expression for an infix expression $A+B-(C/D*E)\wedge F+G/H$ is _____

X: $A+B-(C/D*E)\wedge F+G/H$ ⟶

Scanned Element: $A, +, B, -_{(OP_S)}, (, C, /, D, *, E,), \wedge, F, +, G, /, H$

S, Stack

(OPt) Pop Pop Pop Pop Pop Pop

Y: $AB+CD/E*F\wedge-GH/+$

Stack

Example 2 : Convert Infix Expression 7+3*(5/2 ^ 9-3)+(4/3) to Postfix Expression :

X: 7+3*(5/2 ^ 9-3)+(4/3)

Scanned Element : 7, +, 3, *, (, 5, /, 2, ^, 9, −, 3,), +, (, 4, /, 3,)

Stack :

Y: 735 2 9 ^ / 3 − * + 4 3 / +

~~Question~~

Postfix Expression Evaluation

1) Let 'x' be given Postfix Expression. 'S' be Empty stack.

2) Scan 'x' from Left to Right one Element at a time.

3) A) If Scanned Element == Operand, Push it on to Stack, S.

B) If Scanned Element == Operator, i) Pop top 2 operands from stack

ii) Perform operation 

A op B

iii) Push back result.

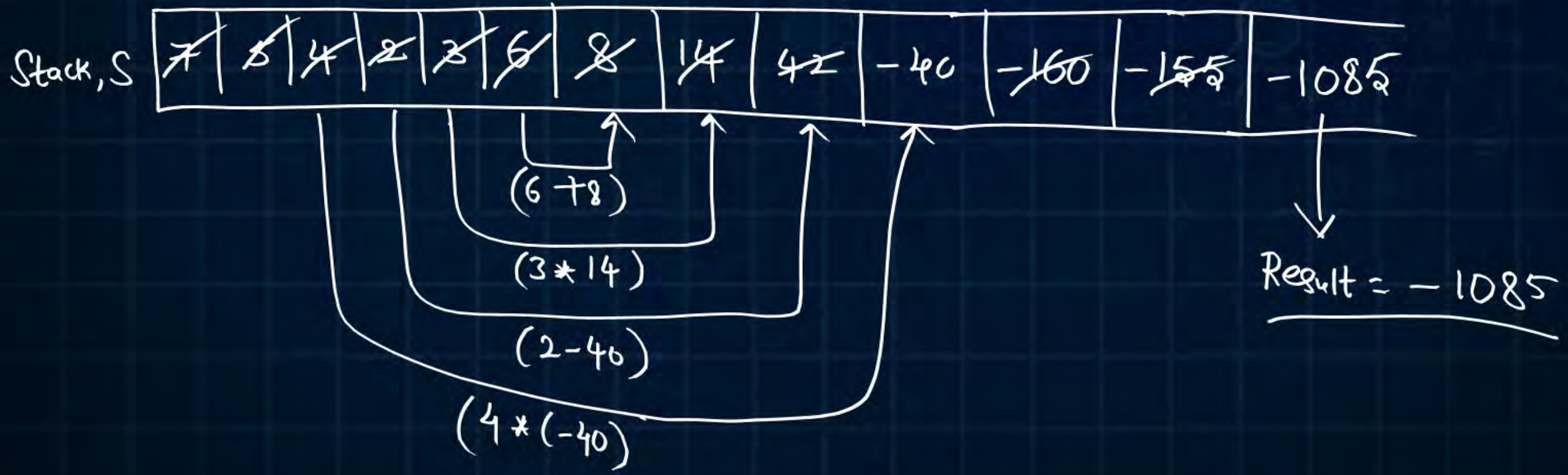
4) Repeat steps (2) and (3) until 'x' is Over.

5) Result on top, Pop it.

Question

Example 1 :

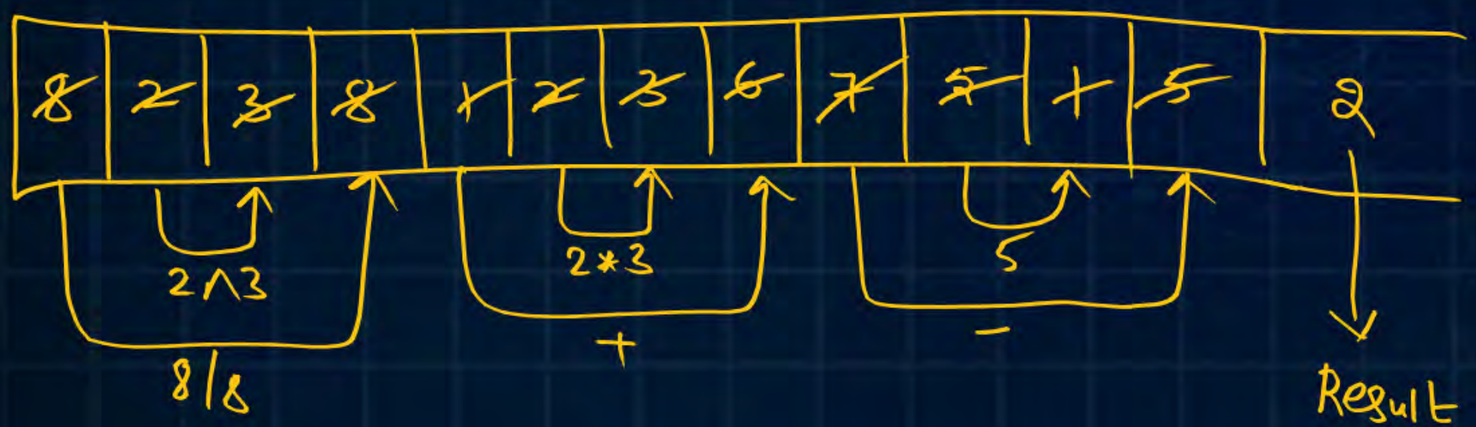
The Result after Evaluating the Expression "7 5 4 2 3 6 8 + * - * + *" is_



Question

GATE 2007

Evaluate 8 2 3 ^ / 2 3 * + 5 1 * −



Question

Home work - 1

Evaluate Postfix Expression :

12 5 7 9 3 - * + * 4 / 3 -

Home work - 2

Convert infix Expression : $\left(\left((P+Q) - \left(R \wedge (S/T*U)/V \right) - W \right) + X \right)$

into Postfix Expression.

Post Your Homework Answers / Queries / Doubts @

t.me/ satyasirpw

Summary

- Data Structures
- Stack
  - Operations
  - Applications
    - Expression Conversion
    - Expression Evaluation.


Thank
THANK
Keep Hustling!