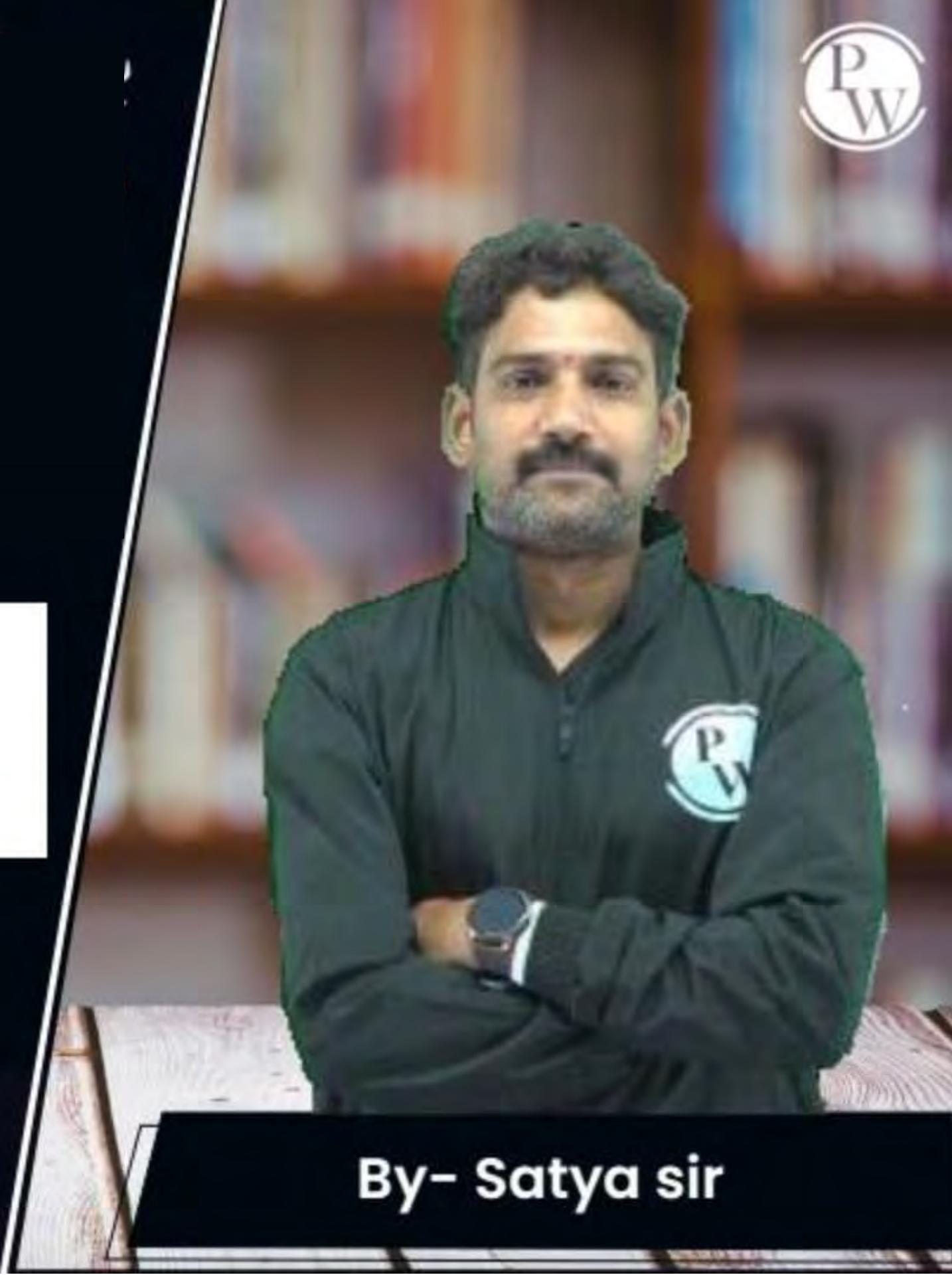


Data Science & AI

DATA STRUCTURES Through Python

STACK

DPP 01 Discussion Notes

A portrait of a man with a beard and mustache, wearing a dark green polo shirt with a 'PW' logo on the chest. He is seated at a desk with his arms crossed.

By- Satya sir

#Q What is the postfix representation of the following infix expression?

$$(A + B) * C - D * E / F$$

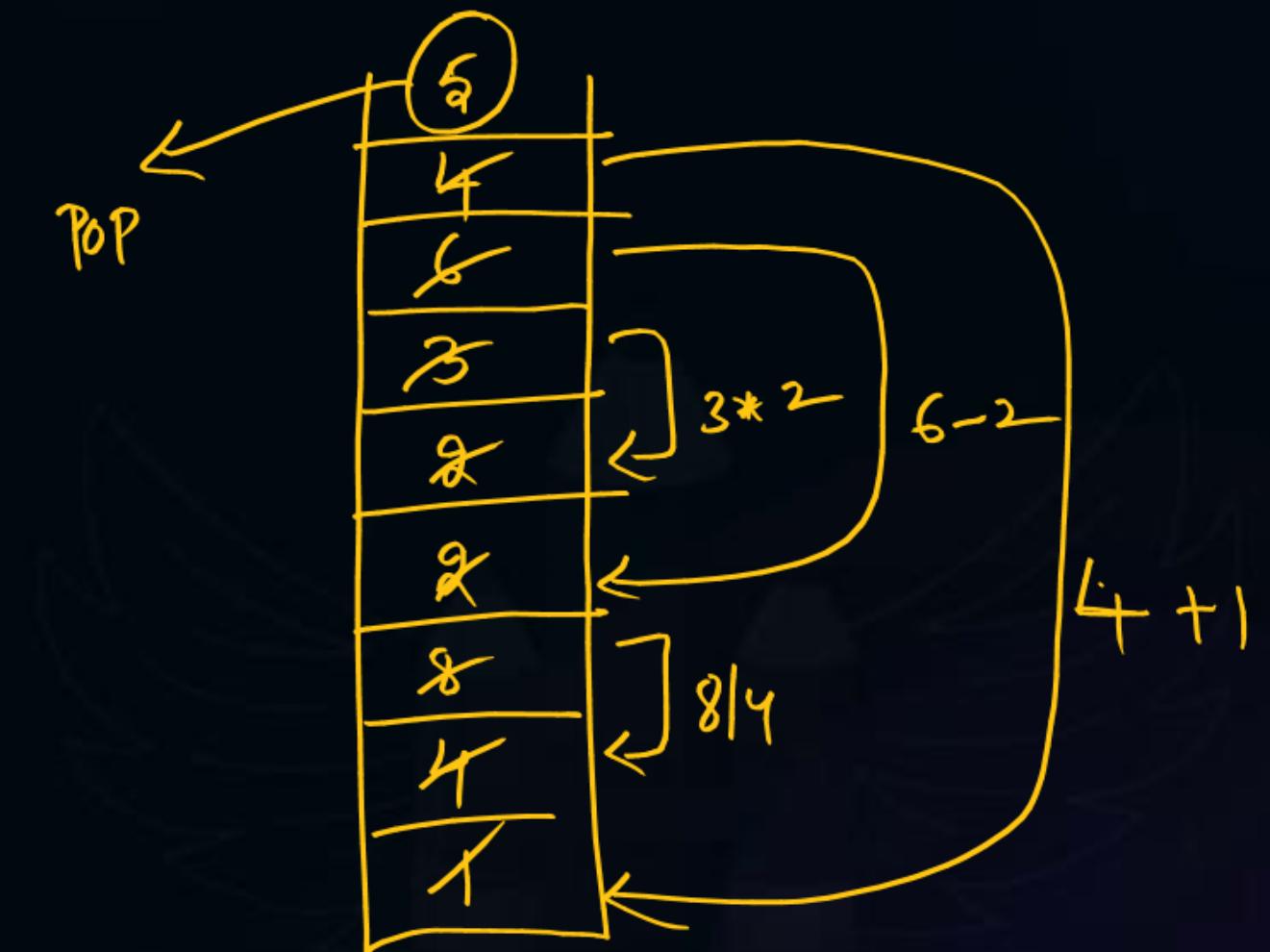
- A AB + C * DE * F - /
- B A B * C + D E * F / -
- C AB + C - DE * F /'
- D AB + C * DE * F / -

Stack [| + | > | * | - | * | / |]
 pop pop pop pop pop

Postfix Exp: AB + C * DE * F / -

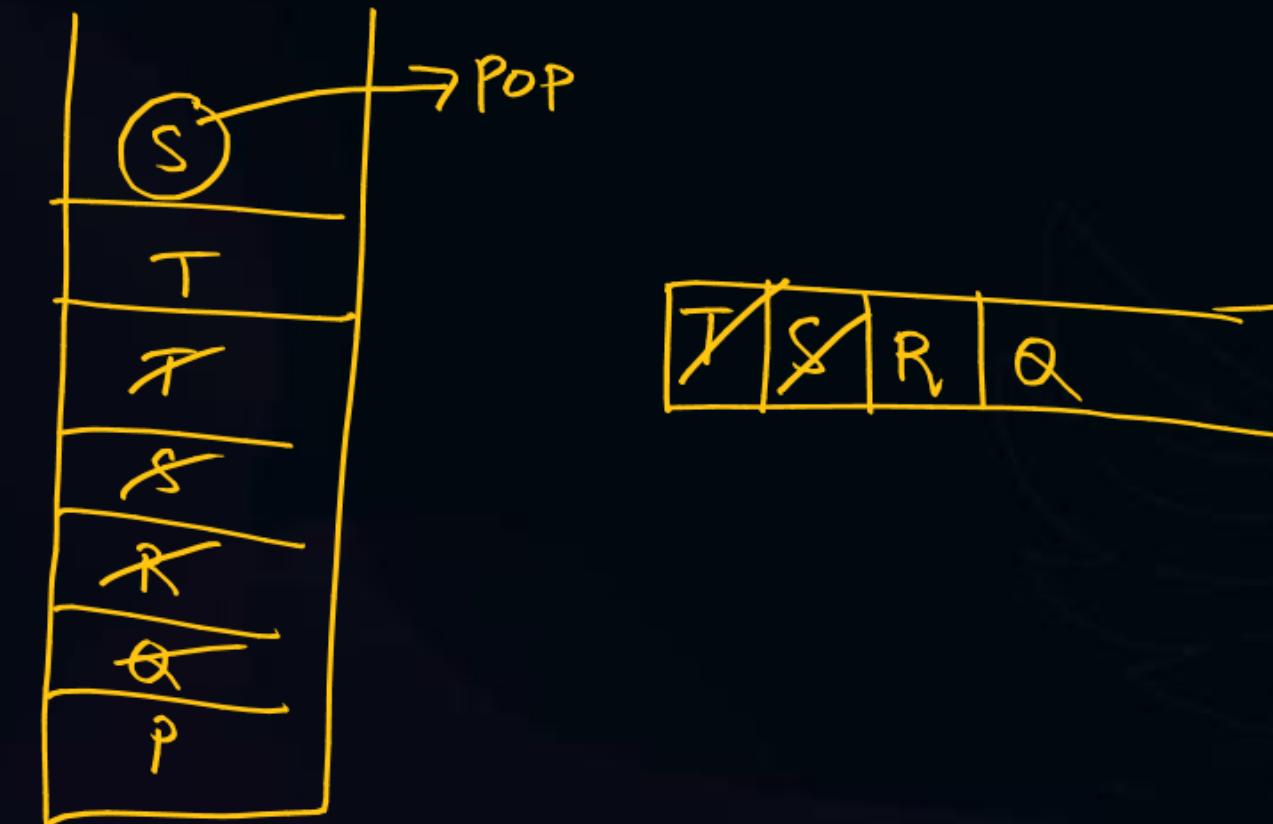
#Q What is the outcome of the prefix expression $+,-,* , 3, 2, /, 8, 4, 1$?

- A 12
- B 5
- C 11
- D 4



#Q The five items P,Q,R,S and T are pushed in a stack, one after the other starting from P. The stack is popped four times and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack, now one item is popped from the stack. The popped item is:

- A P
- B R
- C Q
- D S ✓



#Q Consider the following Stack implement using the stack

SIZE = 11

class Stack:

def __init__(self):

 self.arr = [0] * SIZE # Initialize array with zeros

self.top = -1 # Initialize top pointer

Example usage:

stack = Stack()

What would be the maximum value of the top that does not cause the overflow of the stack?

When $\text{top} = \text{SIZE} - 1$ overflow

$\Rightarrow \text{top} = 11 - 1 \Rightarrow \text{top} = 10$ overflow

Max. value = 9

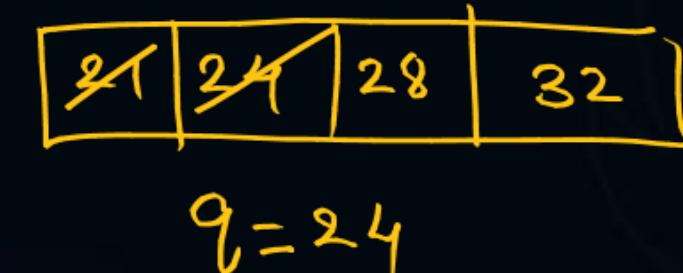
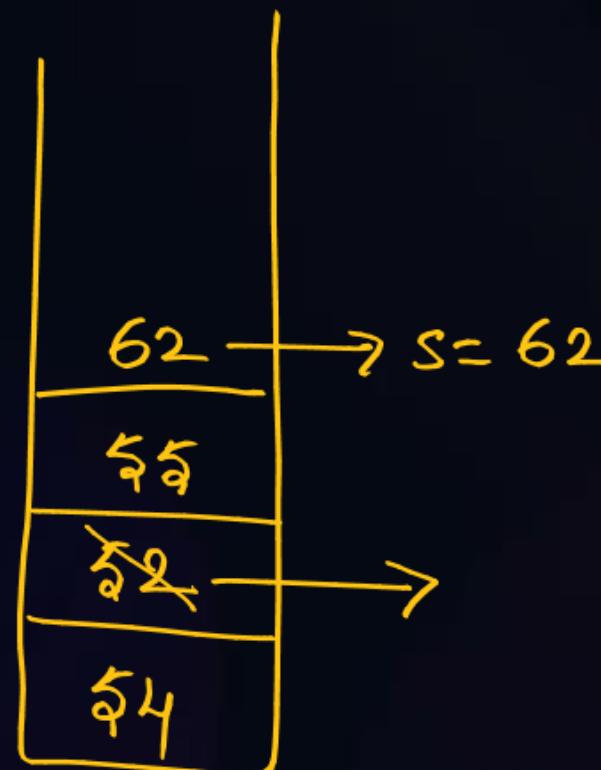
for not to cause overflow

$\text{top} = 0$ ————— [9]

#Q A stack is normally used in digital computers to store the return address at the time of a call because

- A Stacks are non-volatile tile memories X
- B Stacks have large capacity X
- C Information in a stack cannot be altered by other instructions X
- D Stacks permit easy nesting of subroutine (Function Call Sequencing)
Return address can be saved on top of stack.

#Q Consider the following sequence of operations on an empty stack. push(54); push(52); pop(); push(55); push(62); s = pop();
Consider the following sequence of operations on an empty queue. enqueue{21}; enqueue(24); dequeue(); enqueue(28); enqueue(32); q = dequeue();
The value of s + q is 86



$$\begin{aligned}s + q &= 62 + 24 \\&= \underline{\underline{86}}\end{aligned}$$



THANK - YOU