GATE

DS & AI
Data Structure through Python
Stack

DPP: 1

Q1 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/-

Q2 What is the outcome of the prefix expression +, – , * , 3, 2, /, 8, 4, 1 ?

(A) 12 (B) 5

(C) 11 (D) 4

Q3 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

Q4 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?

Q5 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

(B) Stacks have large capacity

(C) Information in a stack cannot be altered by other instructions

(D) Stacks permit easy nesting of subroutine

Q6 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

Stack:

push (54)

top
54

push (52)

top
52
54

pop ( )

top
54

push (55

top
55
54

push (62)

GATE

top
62

55
54

S = pop( ) = 62

GATE

Answer Key

Q1	(D)	Q4	10
Q2	(B)	Q5	(D)
Q3	(D)	Q6	82

GATE

Hints & Solutions

Q1 Text Solution:

() has highest precedence

* and I has same precedence while + and - has same precedence

(* and /) and higher precedence than (+, -)

Associativity is left to right:

Explanation:

(A + B) * C – D * E / F

A B + *C – D* E / F

A B + C * – D * E / F

A B + C * – D E* / F

A B + C * – D E* F/

A B + C * D E * F/–

Q2 Text Solution:

Concept:

	Infix	Prefix	Postfix	Converse infix	Converse prefix	Converse postfix
Method Flow	1. Operand 1 2. Operator 3. Operand 2	1. Operator 2. Operand 1 3. Operand 2	1. Operand 1 2. Operand 2 3. Operator	1. Operand 2 2. Operator 3. Operand 1	1. Operator 2. Operand 2 3. Operand 1	1. Operand 2 2. Operand 1 3. Operator

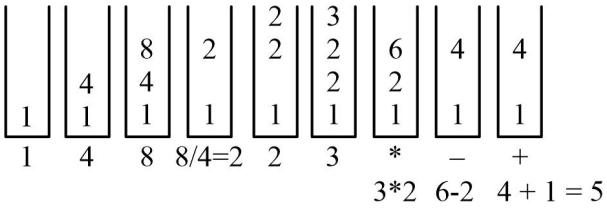
Prefix:

A Prefix expression prints the operator operand 1, and operand 2 respectively.

Explanation:

The given prefix is +, *, 3, 2,1, 8, 4, 1.

The given prefix is +, - , * , 3 , 2, / , 8, 4, 1.



Hence the correct answer is 5.

Q3 Text Solution:

Stack:

Stack is a linear data structure that allows to insert or delete elements from one end i.e. from the top of the stack. It follows a particular order in which elements are inserted or deleted i.e. LIFO (Last in first out).

Queue:

Queue is a linear data structure in which elements are inserted from one end and deleted from the other ends. It follows FIFO property i.e. first in first out.

Explanation:

Given five items P, Q, R, S, and T.

These are pushed into the stack as :

T (TOP)
S
R
Q
P

Now stack is popped four times, and each item popped is inserted into the queue.

Now stack is popped four times, and each item popped is inserted into the queue.

So, items that are popped from the stack are: T, S, R, Q

After insertion into queue, queue looks like this:

Q	R	S	T(FRONT)

Now, two items are deleted from the queue which is: T, S

These are inserted back into the stack.

Stack will be :

S (TOP)

GATE

T
P

Queue will be :

Q	R

Now, one item is popped from the stack which is S.

Q4 Text Solution:

In the given 'stack' implementation, 'size' is defined as 11, which means the array 'self.arr' can hold up to 11 elements. The variable ' self, top' represents the index of the top element of the stack.

Given that 'self.arr' is initialized with zeros and 'self.top' is initialized to '-1', the maximum index ' self .top' can reach without causing a stack overflow (i.e., without attempting to push more elements than the array can hold) is'size - 1'.

So, in this case:

Maximum value of self .top = SIZE – 1 = 11 – 1 = 10

Q5 Text Solution:

- A set of instructions which are used repeatedly in a program can be referred to as Subroutine; Only one copy of this Instruction is stored in the memory
- When a Subroutine is required it can be called many times during the execution of a particular program; A call Subroutine Instruction calls the Subroutine
- Subroutine nesting is a common Programming practice in which one Subroutine call another Subroutine
- A stack is a basic data structure which can be implemented anywhere in the memory; It stores the address of the first subroutine to bottom of stack and stores and last subroutine to top of the stack and returns the address from the top of stack first; Hence stack permits easy nesting of a subroutine

Q6 Text Solution:

enqueue (21)

Front	21			Rear

enqueue (24)

Front	21	24		Rear

dequeue ( )

Front	24			Rear

enqueue (28)

Front	21	28		Rear

enqueue (32)

Front	21	28	32	Rear

q = dequeue() = 24

s + q = 62 + 24 = 86

The value of s + q is 86