

Data Structure Questions and Answers – Stack Operations – 3

[« Prev](#)[Next »](#)

This set of Data Structure Questions and Answers for Freshers focuses on “Stack Operations – 3”.

1. The result of evaluating the postfix expression 5, 4, 6, +, *, 4, 9, 3, /, +, * is?

- a) 600
- b) 350
- c) 650
- d) 588

[View Answer](#)

Answer: b

Explanation: The postfix expression is evaluated using stack. We will get the infix expression as $(5*(4+6))*(4+9/3)$. On solving the Infix Expression, we get

$$\begin{aligned} &(5*(10))*(4+3) \\ &= 50*7 \\ &= 350. \end{aligned}$$

advertisement

2. Convert the following infix expressions into its equivalent postfix expressions

$(A + B \wedge D)/(E - F) + G$

- a) $(A B D \wedge + E F - / G +)$
- b) $(A B D + \wedge E F - / G +)$
- c) $(A B D \wedge + E F / - G +)$
- d) $(A B D E F + \wedge / - G +)$

[View Answer](#)

Answer: a

Explanation: The given infix expression is $(A + B \wedge D)/(E - F) + G$.

$(A B D \wedge +) / (E - F) + G$

$(A B D \wedge + E F -) + G$. '/' is present in stack.

$A B D \wedge + E F - / G +$. Thus Postfix Expression is $A B D \wedge + E F - / G +$.

3. Convert the following Infix expression to Postfix form using a stack

$x + y * z + (p * q + r) * s$, Follow usual precedence rule and assume that the expression is legal.

- a) $xyz*+pq*r+s*+$

- b) $xyz^*+pq^*r+s^*+$
- c) $xyz+^*pq^*r+s^*+$
- d) $xyzp+^{**}qr+s^*+$

[View Answer](#)

Answer: a

Explanation: The Infix Expression is $x + y * z + (p * q + r) * s$.

$(x y z) + (p * q + r) * s$. '+' , '*' are present in stack.

$(x y z * + p q * r) * s$. '+' is present in stack.

$x y z * + p q * r + s * +$. Thus Postfix Expression is $x y z * + p q * r + s * +$.

4. Which of the following statement(s) about stack data structure is/are NOT correct?

- a) Linked List are used for implementing Stacks
- b) Top of the Stack always contain the new node
- c) Stack is the FIFO data structure
- d) Null link is present in the last node at the bottom of the stack

[View Answer](#)

Answer: c

Explanation: Stack follows LIFO.

advertisement

5. Consider the following operation performed on a stack of size 5.

Push(1);

Pop();

Push(2);

Push(3);

Pop();

Push(4);

Pop();

Pop();

Push(5);

After the completion of all operation, the number of elements present in stack are

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: a

Explanation: Number of elements present in stack is equal to the difference between number of push operations and number of pop operations. Number of elements is $5-4=1$.

6. Which of the following is not an inherent application of stack?

- a) Reversing a string
- b) Evaluation of postfix expression
- c) Implementation of recursion
- d) Job scheduling

[View Answer](#)

Answer: d

Explanation: Job Scheduling is not performed using stacks.

7. The type of expression in which operator succeeds its operands is?

- a) Infix Expression
- b) Prefix Expression
- c) Postfix Expression
- d) Both Prefix and Postfix Expressions

[View Answer](#)

Answer: c

Explanation: The expression in which operator succeeds its operands is called postfix expression. The expression in which operator precedes the operands is called prefix expression. If an operator is present between two operands, then it is called infix expressions.

advertisement

8. Assume that the operators +, -, X are left associative and ^ is right associative.

The order of precedence (from highest to lowest) is ^, X, +, -. The postfix expression for the infix expression $a + b \times c - d \wedge e \wedge f$ is

- a) $abc \times + def \wedge \wedge -$
- b) $abc \times + de \wedge f \wedge -$
- c) $ab+c \times d - e \wedge f \wedge$
- d) $-+a \times bc \wedge \wedge def$

[View Answer](#)

Answer: b

Explanation: Given Infix Expression is $a + b \times c - d \wedge e \wedge f$.

$(a \ b \ c \ \times \ +) \ (d \ \wedge \ e \ \wedge \ f)$. '-' is present in stack.

$(a \ b \ c \ \times \ + \ d \ e \ \wedge \ f \ \wedge \ -)$. Thus the final expression is $(a \ b \ c \ \times \ + \ d \ e \ \wedge \ f \ \wedge \ -)$.

9. If the elements "A", "B", "C" and "D" are placed in a stack and are deleted one at a time, what is the order of removal?

- a) ABCD
- b) DCBA
- c) DCAB
- d) ABDC

[View Answer](#)

Answer: b

Explanation: Stack follows LIFO(Last In First Out). So the removal order of elements are DCBA.

Sanfoundry Global Education & Learning Series – Data Structure.

To practice all areas of Data Structure for Freshers, [here is complete set of 1000+ Multiple Choice Questions and Answers](#).

« [Prev - Data Structure Questions and Answers – Stack Operations – 2](#)

» [Next - Data Structure Questions and Answers – Queue Operations](#)

advertisement

Recommended Posts:

1. [Automata Theory Questions and Answers](#)
2. [Java Programming Examples on Collections](#)
3. [Data Structures & Algorithms II – Questions and Answers](#)
4. [C Tutorials](#)
5. [C Programming Examples](#)
6. [Java Programming Examples on Graph Problems & Algorithms](#)
7. [C++ Programming Examples on Graph Problems & Algorithms](#)
8. [C Programming Examples on Linked List](#)
9. [C Programming Examples on Bitwise Operations](#)
10. [Data Science Questions and Answers](#)
11. [Python Programming Examples on Linked Lists](#)
12. [C Programming Examples on Graph Problems & Algorithms](#)
13. [C# Programming Examples on Conversions](#)
14. [C++ Programming Examples on Data-Structures](#)
15. [Java Programming Examples on Data-Structures](#)
16. [C Programming Examples on Data-Structures](#)
17. [Python Programming Examples on Stacks & Queues](#)
18. [C Programming Examples on Stacks & Queues](#)



19. [C# Programming Examples on Data Structures](#)

20. [Python Program to Implement Stack using One Queue](#)



[Manish Bhojasia](#), a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ [LinkedIn](#)

Subscribe Sanfoundry Newsletter and Posts

Subscribe

[About](#) | [Certifications](#) | [Internships](#) | [Jobs](#) | [Privacy Policy](#) | [Terms](#) | [Copyright](#) | [Contact](#)



© 2011-2020 Sanfoundry. All Rights Reserved.

