

Data Structure Questions and Answers – Stack Operations – 2

[« Prev](#)[Next »](#)

This set of Data Structure Interview Questions and Answers focuses on “Stack Operations – 2”.

1. The postfix form of the expression $(A + B) * (C * D - E) * F / G$ is?

- a) $AB + CD * E - FG / **$
- b) $AB + CD * E - F ** G /$
- c) $AB + CD * E - * F * G /$
- d) $AB + CDE * - * F * G /$

[View Answer](#)

Answer: c

Explanation: $((A + B) * (C * D - E) * F) / G$ is converted to postfix expression as

$(AB + (* (C * D - E) * F) / G)$

$(AB + CD * E - * F) / G$

$(AB + CD * E - * F * G /)$. Thus Postfix expression is $AB + CD * E - * F * G /$

advertisement

2. The data structure required to check whether an expression contains balanced parenthesis is?

- a) Stack
- b) Queue
- c) Array
- d) Tree

[View Answer](#)

Answer: a

Explanation: The stack is a simple data structure in which elements are added and removed based on the LIFO principle. Open parenthesis is pushed into the stack and a closed parenthesis pops out elements till the top element of the stack is its corresponding open parenthesis. If the stack is empty, parenthesis is balanced otherwise it is unbalanced.

3. What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm?

- a) Linked List
- b) Stack
- c) Queue



d) Tree

[View Answer](#)

Answer: b

Explanation: In recursive algorithms, the order in which the recursive process comes back is the reverse of the order in which it goes forward during execution. The compiler uses the stack data structure to implement recursion. In the forwarding phase, the values of local variables, parameters and the return address are pushed into the stack at each recursion level. In the backing-out phase, the stacked address is popped and used to execute the rest of the code.

4. The process of accessing data stored in a serial access memory is similar to manipulating data on a _____

- a) Heap
- b) Binary Tree
- c) Array
- d) Stack

[View Answer](#)

Answer: d

Explanation: In serial access memory data records are stored one after the other in which they are created and are accessed sequentially. In stack data structure, elements are accessed sequentially. Stack data structure resembles the serial access memory.

5. The postfix form of $A*B+C/D$ is?

- a) $*AB/CD+$
- b) $AB*CD/+$
- c) $A*BC+/D$
- d) $ABCD+/*$

[View Answer](#)

Answer: b

Explanation: Infix expression is $(A*B)+(C/D)$

$AB*+(C/D)$

$AB*CD/+$. Thus postfix expression is $AB*CD/+$.

advertisement

6. Which data structure is needed to convert infix notation to postfix notation?

- a) Branch
- b) Tree
- c) Queue
- d) Stack

[View Answer](#)

Answer: d

Explanation: The Stack data structure is used to convert infix expression to postfix expression. The purpose of stack is to reverse the order of the operators in the expression. It also serves as a storage structure, as no operator can be printed until both of its operands have appeared.

7. The prefix form of $A-B / (C * D ^ E)$ is?

- a) $-/*^ACBDE$
- b) $-ABCD*^DE$
- c) $-A/B*C^DE$
- d) $-A/BC*^DE$

[View Answer](#)

Answer: c

Explanation: Infix Expression is $(A-B)/(C*D^E)$

$(-A/B)(C*D^E)$

$-A/B*C^DE$. Thus prefix expression is $-A/B*C^DE$

8. What is the result of the following operation?

Top (Push (S, X))

- a) X
- b) $X+S$
- c) S
- d) XS

[View Answer](#)

Answer: a

Explanation: The function Push(S,X) pushes the value X in the stack S. Top() function gives the value which entered last. X entered into stack S at last.

9. The prefix form of an infix expression $(p + q) - (r * t)$ is?

- a) $+pq - *rt$
- b) $- +pqr * t$
- c) $- +pq * rt$
- d) $- + * pqrt$

[View Answer](#)

Answer: c

Explanation: Given Infix Expression is $((p+q)-(r*t))$

$(+pq)-(r*t)$

$(-+pq)(r*t)$

$-+pq*rt$. Thus prefix expression is $-+pq*rt$.

10. Which data structure is used for implementing recursion?

- a) Queue
- b) Stack
- c) Array



d) List

[View Answer](#)

Answer: b

Explanation: Stacks are used for the implementation of Recursion.

advertisement

Sanfoundry Global Education & Learning Series – Data Structure.

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

« [Prev - Data Structure Questions and Answers – Stack Operations – 1](#)

» [Next - Data Structure Questions and Answers – Stack Operations – 3](#)

advertisement

Recommended Posts:

1. [C Programming Examples](#)
2. [C Programming Examples using Recursion](#)
3. [C Programming Examples without using Recursion](#)
4. [Data Science Questions and Answers](#)
5. [C++ Algorithms, Problems & Programming Examples](#)
6. [Java Algorithms, Problems & Programming Examples](#)
7. [C Programming Examples on Linked List](#)
8. [Python Programming Examples on Linked Lists](#)
9. [Data Structures & Algorithms II – Questions and Answers](#)
10. [C# Programming Examples on Conversions](#)
11. [C++ Programming Examples on Graph Problems & Algorithms](#)
12. [Java Programming Examples on Graph Problems & Algorithms](#)
13. [C Programming Examples on Graph Problems & Algorithms](#)
14. [Python Programming Examples on Stacks & Queues](#)



15. [C Programming Examples on Stacks & Queues](#)
16. [C++ Programming Examples on Data-Structures](#)
17. [C Programming Examples on Data-Structures](#)
18. [Java Programming Examples on Data-Structures](#)
19. [C# Programming Examples on Data Structures](#)
20. [Data Structure Questions and Answers – Non-recursive Depth First Search](#)



[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.

