

# Stack Operation

---

**1. Process of inserting an element in stack is called \*\* \_\_\_\_ \*\***

- a. Create
- b. Push
- c. Evaluation
- d. Pop

**Answer: b**

**Explanation:**

Push operation allows users to insert elements in the stack. If the stack is filled completely and trying to perform push operation stack – overflow can happen.

---

**2. Process of removing an element from stack is called \*\* \_\_\_\_ \*\***

- a. Create
- b. Push
- c. Evaluation
- d. Pop

**Answer: d**

**Explanation:**

Elements in the stack are removed using pop operation. Pop operation removes the top most element in the stack i.e. last entered element.

---

**3. In a stack, if a user tries to remove an element from an empty stack it is called \*\* \_\_\_\_ \*\***

- a. Underflow
- b. Empty collection
- c. Overflow
- d. Garbage Collection

**Answer: a**

**Explanation:**

Underflow occurs when the user performs a pop operation on an empty stack. Overflow occurs when the stack is full and the user performs a push operation. Garbage Collection is used to recover the memory occupied by objects that are no longer used.

---

**4. Pushing an element into stack already having five elements and stack size of 5, then stack becomes \*\* \_\_\_\_ \*\***

- a. Overflow
- b. Crash
- c. Underflow
- d. User flow

**Answer:** a

**Explanation:**

Underflow occurs when the user performs a pop operation on an empty stack. Overflow occurs when the stack is full and the user performs a push operation. Garbage Collection is used to recover the memory occupied by objects that are no longer used.

---

**5. Entries in a stack are “ordered”. What is the meaning of this statement?**

- a. A collection of stacks is sortable
- b. Stack entries may be compared with the '<' operation
- c. The entries are stored in a linked list
- d. There is a Sequential entry that is one by one

**Answer:** d

**Explanation:**

In stack data structure, elements are added one by one using push operation. Stack follows LIFO Principle i.e. Last In First Out(LIFO).

---

**6. Which of the following is not the application of stack?**

- a. A parentheses balancing program
- b. Tracking of local variables at run time
- c. Compiler Syntax Analyzer
- d. Data Transfer between two asynchronous process

**Answer:** d

**Explanation:**

Data transfer between the two asynchronous process uses the queue data structure for synchronisation. The rest are all stack applications.

---

**7. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (0(0)(0))?**

- a. 1
- b. 2
- c. 3
- d. 4 or more

**Answer:** c

**Explanation:**

In the entire parenthesis balancing method when the incoming token is a left parenthesis it is pushed into stack. A right parenthesis makes pop operation to delete the elements in stack till we get left parenthesis as top most element. 3 elements are there in stack before right parentheses comes. Therefore, maximum number of elements in stack at run time is 3.

---

**8. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence**

**that contains 2 left parentheses and 3 right parentheses (in some order). The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation?**

- a. 1
- b. 2
- c. 3
- d. 4 or more

**Answer: b**

**Explanation:**

In the entire parenthesis balancing method when the incoming token is a left parenthesis it is pushed into stack. A right parenthesis makes pop operation to delete the elements in stack till we get left parenthesis as top most element. 2 left parenthesis are pushed whereas one right parenthesis removes one of left parenthesis. 2 elements are there before right parenthesis which is the maximum number of elements in stack at run time.

---

**9. What is the value of the postfix expression 6 3 2 4 + - \*?**

- a. 1
- b. 40
- c. 74
- d. -18

**Answer: d**

**Explanation:**

Postfix Expression is  $(6 * (3 - (2 + 4)))$  which results -18 as output.

---

**10. Here is an infix expression:  $4 + 3 * (6 * 3 - 12)$ . Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?**

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

**Answer: d**

**Explanation:**

When we perform the conversion from infix to postfix expression +, -, (, \_ symbols are placed inside the stack. A maximum of 4 symbols are identified during the entire conversion.

---

**11. 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.  $AB + CDE _ - * F * G /$

**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, the postfix expression is  $AB + CD * E - * F * G /$

---

**12. The data structure required to check whether an expression contains a balanced parenthesis is?**

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

**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, the parenthesis is balanced; otherwise, it is unbalanced.

---

**13. What data structure would you most likely see in non-recursive implementation of a recursive algorithm?**

- a. Linked List
- b. Stack
- c. Queue
- d. Tree

**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.

---

**14. 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

**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.

---

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

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

**Answer:** b

**Explanation:**

Infix expression is  $(AB)+(C/D)$

$AB+(C/D)$

$ABCD/+,$  Thus postfix expression is  $ABCD/+.$

---

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

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

**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.

---

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

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

**Answer:** c

**Explanation:**

Infix Expression is  $(A-B)/(CD^E)$

$(-A/B)(CD^E)$

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

---

**18. What is the result of the following operation? Top (Push (S, X))**

- a. X
- b.  $X+S$
- c. S

d. XS

**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.

---

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

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

**Answer: c**

**Explanation:**

Given Infix Expression is  $((p+q)-(rt))$

$(+pq)-(rt)$

$(-+pq)(rt)$

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

---

**20. Which data structure is used for implementing recursion?**

- a. Queue
- b. Stack
- c. Array
- d. List

**Answer: b**

**Explanation:**

Stacks are used for the implementation of Recursion.

---

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

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

**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  $(5(10))(4+3) = 507 = 350$ .

---

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

**Answer: a**

**Explanation:**

The given infix expression is  $(A + B \wedge D) / (E - F) + G$ .  $(A B D ^ + ) / (E - F) + G$   $(A B D ^ + E F - ) + G$ . '/' is present in stack.  $A B D ^ + E F - / G +$ . Thus Postfix Expression is  $A B D ^ + E F - / G +$ .

---

**23. 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.  $xyz^*+pqr+s+$
- b.  $xyz^*+pqr^+s+$
- c.  $xyz+pqr^+s^*+$
- d.  $xyzp^+*qr^+s+$

**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  $x y z ^ + p q * r + s ^ +$ .

---

**24. 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

**Answer: c**

**Explanation:**

Stack follows LIFO.

---

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

```
Push(1);
Pop();
Push(2);
Push(3);
Pop();
Push(4);
Pop();
Push(5);
```

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

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

**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  $5-4=1$ .

---

**26. 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

**Answer:** d

**Explanation:**

Job Scheduling is not performed using stacks.

---

**27. 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

**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.

---

**28. 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  $a + b \times c - d^e^f$  is?**

- a.  $a \ b \ c \ x \ + \ d \ e \ f \wedge \wedge \ -$
- b.  $a \ b \ c \ x \ + \ d \ e \wedge \ f \wedge \ -$
- c.  $a \ b \ + \ c \ x \ d \ - \ e \wedge \ f \wedge$
- d.  $- \ + \ a \ x \ b \ c \wedge \wedge \ d \ e \ f$

**Answer:** a

**Explanation:**

Given Infix Expression is  $a + b \times c - d^e^f$ . And ^ is right associative. Thus, the final postfix expression is  $a \ b \ c \ x \ + \ d \ e \ f \wedge \wedge \ -$

---



**29. 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

**Answer:** b

**Explanation:**

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

---