

Two Marks Each ::

1. Infix to post conversion

- i. $(A-(b+c)*D)^{(D+E)}$
 a. $Abc+-D* D^E+$ b. $Abc+D* -DE+^$
 c. $Abc+*D-DE+^$ d. $Abc+D*DE+-^$

- ii. $b * c + d / e$
 iii. $A / B \wedge C - D$
 iv. $A + B * (C + D) / F + D * E$
 v. $(A-2 * (B+C)-D*E)*F$
 vi. $(A+B) *(C-D) *F+C$

2. Infix to Prefix conversion

- i. $(A-(b+c)*D)^{(D+E)}$
 a. $Abc+-D* D^E+$ b. $Abc+D* -DE+^$
 c. $Abc+*D-DE+^$ d. $Abc+D*DE+-^$

- ii. $b * c + d / e$
 iii. $A / B \wedge C - D$
 iv. $A + B * (C + D) / F + D * E$

3. Evaluate forllowing postfix expression

- i. $ABCD*-+ (A=2, B=3, C=4, D=5)$
 ii. $DC*BA+- ((A=2, B=3, C=4, D=5)$
 iii. $AB *C-D+$
 iv. $ABC+*D-$
 v. $25 7* 14 -6 +$
 vi. $1 24 3 +* 41 -$
 vii. $2 37 4 + * 15 -$

4. Evaluate forllowing prefix expression

- i. $+*ABCD (A=2, B=3, C=4, D=5)$
 ii. $-*A+BCD (A=2, B=3, C=4, D=5)$

5. Using Pop(S,I),Push(S,I),Read(I),and Print(I), the variable S(stack),an dI and given input file A,B,C,D,E,F<EOF> which stacks will give the correct sequence::

- a) 5 A b) 5 c) 5 d) 5
 4 B 4 4 4
 3 C 3 D 3 F 3 C
 2 D 2 A 2 D 2 E
 1 E 1 F 1 B 1 B

6. stack A has entries a,b,c (with a on the top). Stack B is empty. An entry popped from stack A can be printed immediately or pushed to stack B. An entery popped out of the stack B can only be printed. In this arrangement which of the following permutation of a,b,c are not possible.

- a) b,a,c b) b,c,a c) c,a,b d) a,b,c

7. following sequence of operation is performed on a stack push(1), push(2),pop push(1),push(2) , pop,pop,pop push(2),pop. The sequence of popeed out values are ::

- a) 2,2,1,1,2 b) 2,2,1,2,2 c) 2,1,2,2,1 d) 2,1,2,2,2

8. In evaluating the arithmetic expression $2*3-(4+5)$, using stacks to evalutate its equivalent postfix form, which of the following stack configuration is not possible ?

- a) - b) - c) - d) -
 - 4 - 9
 4 5 9 3
 6 6 6 2

9. the follwoing postfix expression with single digit operands is evaluated using stack

$$8 2 3 \wedge / 2 3 * + 5 1 * -$$

The top two elements of the stack after the first * is evaluated are :
a) 6,1 b) 5,7 c) 3,2 d) 1,5

10. What is the postfix representation of this expression?

$(12 - a) * (b + 9) / (d * 4)$

A. $4b * d9 + a12$ B. $/12a - b9 + d$ C. $12 - a * b + 9 /$ D. $12a - b9 + * d$

E. None of the above

11. What will be the postfix form of the above expression -

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

A. $AB + CDE * - FG / * *$ B. $AB + CD * E - FG / * *$
C. $AB + CD * E - FG \square / \square$ D. None of these

12. if the sequence of operations -

push(1) push(2) pop() push(1)push(2) pop()pop()pop()
push(2) pop()

13. Expression

$1 * 2 ^ 3 * 4 ^ 5 * 6$ is evaluated as -

A. 173458 B. 162^{30} C. 32^{30} D. 49152

For questions 15- 17, consider the following operations on a Queue data structure that stores int values.

```
Queue q = new Queue();  
q.enqueue(3); q.enqueue(5); q.enqueue(9);  
System.out.println(q.dequeue()); // d1  
q.enqueue(2); q.enqueue(4);  
System.out.println(q.dequeue()); // d2  
System.out.println(q.dequeue()); // d3  
q.enqueue(1); q.enqueue(8);
```

14. After the code above executes, how many elements would remain in q?

a) 0 b) 4 c) 5 d) 6 e) 7

15. What value is returned by the last dequeue operation (denoted above with a d3 in comments)?

a) 3 b) 5 c) 9 d) 2 e) 4

16. If we replace the System.out.println statements (denoted in comments as d1, d2 and d3) with the statement q.enqueue(q.dequeue()); q would contain which order of int values after all instructions have executed?

a) 3, 5, 9, 2, 4, 1, 8 b) 3, 5, 9, 1, 8, 2, 4
b) 5, 9, 2, 4, 1, 8, 3 c) 3, 2, 4, 5, 9, 1, 8
c) 2, 4, 1, 8, 3, 5, 9

17. Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

```
void fun(Queue *Q)  
{  
    Stack S; // Say it creates an empty stack S  
    // Run while Q is not empty  
    while (!isEmpty(Q))  
    {  
        // dequeue an item from Q and push the dequeued item to S  
        push(&S, dequeue(Q));  
    }  
    // Run while Stack S is not empty  
    while (!isEmpty(&S))
```


(2)

```
{
// Pop an item from S and enqueue the popped item to Q
enqueue(Q, pop(&S));
}
```

What does the above function do in general?

- A. Removes the last from Q B. Keeps the Q same as it was before the call
C. Makes an empty Q D. Reverse the Q

18. An implementation of a queue Q, using two stacks S1 and S2, is given below:

```
void insert(Q, x) {
push (S1, x);
}
void delete(Q) {
if (stack-empty(S2)) then
if (stack-empty(S1)) then {
print("Q is empty");
return;
}
else while (! (stack-empty(S1))) {
x=pop(S1);
push(S2, x);
}
x=pop(S2);
}
```

Let n insert and m ($\leq n$) delete operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n ?

- A. $n+m \leq x < 2n$ and $2m \leq y \leq n+m$ B. $n+m \leq x < 2n$ and $2m \leq y \leq 2n$
C. $2m \leq x < 2n$ and $2m \leq y \leq n+m$ D. $2m \leq x < 2n$ and $2m \leq y \leq 2n$

19. Consider the following operation along with Enqueue and Dequeue operations on queues, where k is a global parameter.

```
MultiDequeue(Q) {
m = k
while (Q is not empty and m > 0) {
Dequeue(Q)
m = m - 1
}
}
```

What is the worst case time complexity of a sequence of n MultiDequeue() operations on an initially empty queue? (GATE CS 2013)

- (A) $\Theta(n)$ (B) $\Theta(n+k)$ (C) $\Theta(nk)$ (D) $\Theta(n^2)$

20. Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
{
IntQueue q = new IntQueue();
q.enqueue(0); q.enqueue(1);
for (int i = 0; i < n; i++)
{
int a = q.dequeue(); int b = q.dequeue(); q.enqueue(b);
q.enqueue(a + b); printf(a); }
}
```

- A. Prints numbers from 0 to $n-1$ B. Prints numbers from $n-1$ to 0
C. Prints first n Fibonacci numbers D. Prints first n Fibonacci numbers in reverse order

18. Explanation:
The order in which insert and delete operations are performed matters here. The best case: Insert and delete operations are performed alternatively. In every delete operation, 2 pop and 1 push operations are performed. So, total $m + n$ push (n push for insert() and m push for delete()) operations and $2m$ pop operations are performed. The worst case: First n elements are inserted and then m elements are deleted. In first delete operation, $n + 1$ pop operations and n push operation are performed. Other than first, in all delete operations, 1 pop operation is performed. So, total $m + n$ pop operations and $2n$ push operations are performed (n push for insert() and n push for delete())

19 Explanation : since queue is empty initially, the condition of while loop never becomes true. So the time complexity is $\theta(n)$.

20 Explanation:
The function prints first n Fibonacci Numbers. Note that 0 and 1 are initially there in q. In every iteration of loop sum of the two queue items is enqueued and the front item is dequeued.

One Marks Each ::

Q1. The process of accessing data stored in a tape is similar to manipulating data on a
a) stack ☒ b) queue c) list d) heap

Q2. which of the following is a collection of item into which items can be inserted arbitrarily and from which only smallest item can be removed.

☒ a) Ascending order Priority Queue b) Descending order Priority Queue
c) Fifo Queue d) Lifo Queue.

Q3. Postfix expression is just reverse of the prefix expression

a. True b. False. c. Cant say. d None of these.
b.

Q4. Stack cant be used for

a) postfix expression evaluation c) expression conversion
b) In recursion ☒ d) Resource allocation by Operating system

Q5. A linear List in which elements can be added or removed at either end but not in the middle is called as
a) queue b) stack ☒ c) dequeue d) tree

Q6. Consider a linked implementation of queue with two pointers : front and rear. The time needed to insert element in a queue of length n is

☒ a) $O(1)$ b) $O(\log^2 n)$ c) $(\log n)$ d) $(n \log n)$
b)

Q7. Let Q be the queue of the integer defined as follows ::

```
# define Max 300
struct queue
```


Q 8. to insert an element in the queue, we may write operation

- a) $++Q.item[Q.rear]=X$; b) $Q.item[Q.rear]++=X$; c) $Q.item[++Q.rear]=X$; d) none of these

Q9. which of the following data structure may give overflow error, even though the current number of element in it are less than its size?

- a) simple Q b) circular Q c) stack d) none of these

Q10. Entries in a stack are "ordered". What is the meaning of this statement?

- a. A collection of stacks can be sorted. b. Stack entries may be compared with the ' $<$ ' operation.
c. The entries must be stored in a linked list. d. There is a first entry, a second entry, and so on.

Q11. Which of the following stack operations could result in stack underflow?

- a. is_empty b. pop c. push d. Two or more of the above answers

Q12. Which of the following applications may use a stack?

- a. A parentheses balancing program. b. Keeping track of local variables at run time.
c. Syntax analyzer for a compiler. d. All of the above.

Q13. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. What is the maximum number of parentheses that will appear on the stack AT ANY ONE TIME when the algorithm analyzes: $(()()())$?

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

Q14. Suppose we have an array implementation of the stack class, with ten items in the stack stored at $data[0]$ through $data[9]$. The CAPACITY is 42. Where does the push member function place the new entry in the array?

- a. $data[0]$ b. $data[1]$ c. $data[9]$ d. $data[10]$

Q15. Consider the implementation of the stack using a partially-filled array. What goes wrong if we try to store the top of the stack at location $[0]$ and the bottom of the stack at the last used position of the array?

- A. Both peek and pop would require linear time. B. Both push and pop would require linear time.
C. The stack could not be used to check balanced parentheses.
D. The stack could not be used to evaluate postfix expressions.

Q.16 In the linked list implementation of the stack class, where does the push member function place the new entry on the linked list?

- A. At the head B. At the tail C. After all other entries that are greater than the new entry.
D. After all other entries that are smaller than the new entry.

Q.17 In the array version of the stack class (with a fixed-sized array), which operations require linear time for their worst-case behavior?

- A. is_empty B. peek C. pop D. push
E. None of these operations require linear time.

Q.18 In the linked-list version of the stack class, which operations require linear time for their worst-case behavior?

- A. is_empty B. peek C. pop D. push
E. None of these operations require linear time.

Q.19. What is the value of the postfix expression $6\ 3\ 2\ 4\ +\ -\ *$;

- A. Something between -15 and -100 B. Something between -5 and -15
C. Something between 5 and -5 D. Something between 5 and 15

ans --> A

Q.20. 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. What is 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

Q.21. Which of the following name does not relate to stacks?

- ☒ a. FIFO lists b. LIFO list c. Piles d. Push-down lists

Q.22. Identify the data structure which allows deletions at both ends of the list but insertion at only one end.

- ☒ a. Input-restricted deque b. Output-restricted deque c. Priority queues d. None of above

Q.23. Which of the following data structure is non-linear type?

- a. Strings b. Lists c. Stacks ☒ d. None of above

.24 Entries in a stack are "ordered". What is the meaning of this statement?

- A. A collection of stacks can be sorted. B. Stack entries may be compared with the '<'
C. The entries must be stored in a linked list. ☒ D. There is a first entry, a second entry, and so on.

Q.25 Consider the following pseudocode:

declare a stack of characters

```
while ( there are more characters in the word to read )
{
    read a character
    push the character on the stack
}
while ( the stack is not empty )
{
    write the stack's top character to the screen
    pop a character off the stack
}
```

What is written to the screen for the input "carpets"?

- A. serec B. carpets ☒ C. steprac D. ccaarrppectss

Q.26 Here is an INCORRECT pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while ( more input is available )
{
    read a character
    if ( the character is a '(' )
        push it on the stack
    else if ( the character is a ')' and the stack is not empty )
        pop a character off the stack
    else
        print "unbalanced" and exit
}
print "balanced"
```

Which of these unbalanced sequences does the above code think is balanced?

- A. ((()) B. ()) (() C. (() ()) D. (()) ()

Q. 27 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). What is the maximum number of parentheses that will ever appear on the stack AT ONE TIME during the computation?

- A. 1 B. 2 C. 3 D. 4 E. 5 or more

Q. 28 in a array version of the stack class (with a fixed-sized array), which operations require linear time for their worst-case behavior

- A. is_empty B. peek C. pop D. Push E. None of these operations require linear time.

Q. 29 What is the value of the postfix expression 6 3 2 4 + - *;

- A. Something between -15 and -100 B. Something between -5 and -15 C. Something between 5 and -5
D. Something between 5 and 15 E. Something between 15 and 100