S.No.: 01DLSK_CS_ABCDEFG_130826

MADE EASY
Leading Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

PROGRAMMING & DATA STRUCTURES

COMPUTER SCIENCE & IT

Date of Test : 11/09/2023

ANSWER KEY ➤

1.	(c)	7.	(c)	13.	(a)	19.	(a)	25.	(c)
2.	(c)	8.	(a)	14.	(d)	20.	(c)	26.	(b)
3.	(d)	9.	(a)	15.	(c)	21.	(a)	27.	(d)
4.	(a)	10.	(c)	16.	(b)	22.	(c)	28.	(b)
5.	(d)	11.	(b)	17.	(d)	23.	(d)	29.	(d)
6.	(c)	12.	(d)	18.	(b)	24.	(c)	30.	(b)

DETAILED EXPLANATIONS

1. (c)

Here structure creates the memory for array and union, but union creates the memory for only 'long z' which is maximum among all data types in union

$$u = \max(2, 4, 8) = 8$$

$$\therefore \quad t = 20 + 8 = 28$$

2. (c)

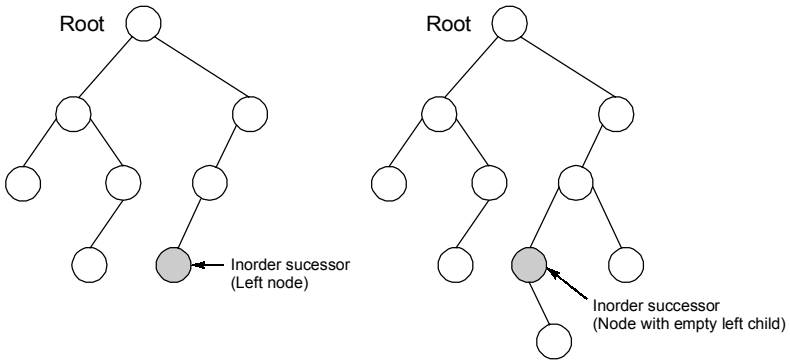
P1 traverses node by node

P2 traverses by skipping one node. The node pointed by P1 is the middle node in the linked list when P2 reaches to NULL.

$\therefore$ Middle element of the list is printed by P1 $\rightarrow$ value.

3. (d)

Successor of Root element is always the smallest element of the Right subtree. Because it will be the next largest element after the elemetn to be deleted.



4. (a)

Iterations of for statement

for:	$i = 0$		$i = 1$		$i = 2$		$i = 3$		$i = 4$	
	x	y	x	y	x	y	x	y	x	y
x << 1;	1	2	1	1	1	2	1	3	1	4
$y = x + i$;	1	1	1	2	1	3	1	4	1	5

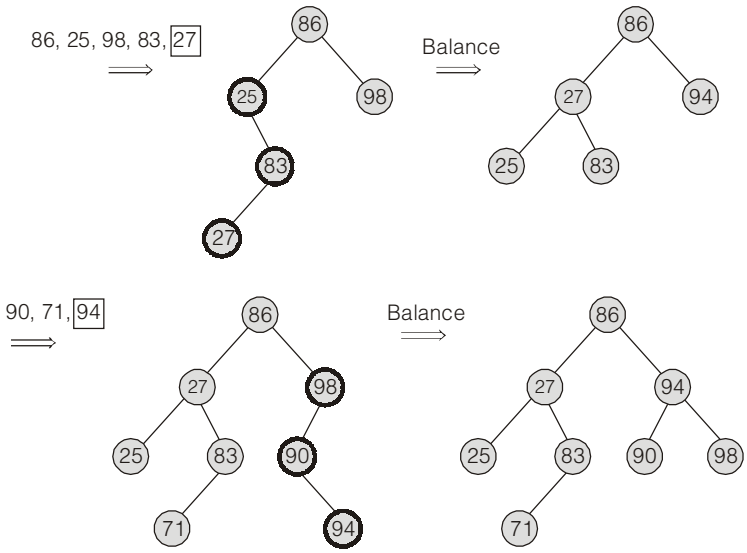
$x = 1, \quad y = 5$

[**Note:** x << 1 will not change the value of x, but $x = x$ << 1 will changes the value of x]

5. (d)

Both (a) and (b) are true to keep the first in first out order, a queue can be implemented using linked list in any of the given two ways.

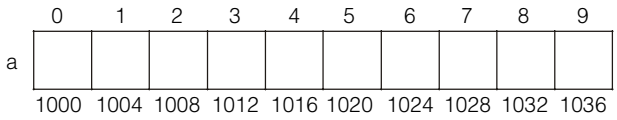
6. (c)



The order: 86, 25, 98, 83, 27, 90, 71, 94 will result the given AVL

[**Note:** Option (a) and obtion (b) will generate different AVL trees]

7. (c)



Assume first element address is 1000.

p holds 1020

q holds 1028

p + 1 can point 1024

q – 3 can point 1016

q – p can result displacement between them.

p + q can lead segmentation fault hence addition of this pointers is invalid.

8. (a)

Linker error: Undefined symbol-*i*

Extern int *i*; Specifies to the compiler that the memory for *i* is allocated in some other program and that address will be given to the current program at the time of linking. But linker finds that no other variable of name '*i*' is available in any other program with memory space allocated for it. Hence linker error occurred.

9. (a)

x > y ? y > z ? 1 : z > x ? 2 : 3 : y > z ? 4 : z > x ? 5 : 6;

Number of ? = 5

Number of : = 5

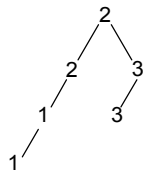
- If both are equal there check the condition.
- Every colon should match with just before question mark.
- Every question mark followed by colon not immediately but following.

10. (c)

The function counts number of characters in the input string. Gate2026 has 8 characters. P1 points to null character and P2 points to first character.

11. (b)

Tree after execution of above code



Level order traversal is 2 2 3 1 3 1.

12. (d)

The correct output is "BCD" when the function pr ( ) is first called the value of i is initialized to 1. After the pr ( ) completes it execution $i = 2$ is retained for it's next call as "i" is static variable.

$\therefore$ 65 + 1 = 66 (B)

65 + 2 = 67(C)

65 + 3 = 68(D)

$\therefore$ BCD is the correct output.

13. (a)

<< and >> are bitwise operators used to multiply and divide by power of 2 respectively (shift operators)

$\therefore$ $i << 3 \Rightarrow i * 8$

$j >> 2 \Rightarrow j / 4$

14. (d)

Both (a) and (b) are true to keep the first in first out order, a queue can be implemented using linked list in any of the given two ways.

15. (c)

f(n – 1) + f(n – 2) + 2 values printed by f(n), where 2 indicate number of print statements.

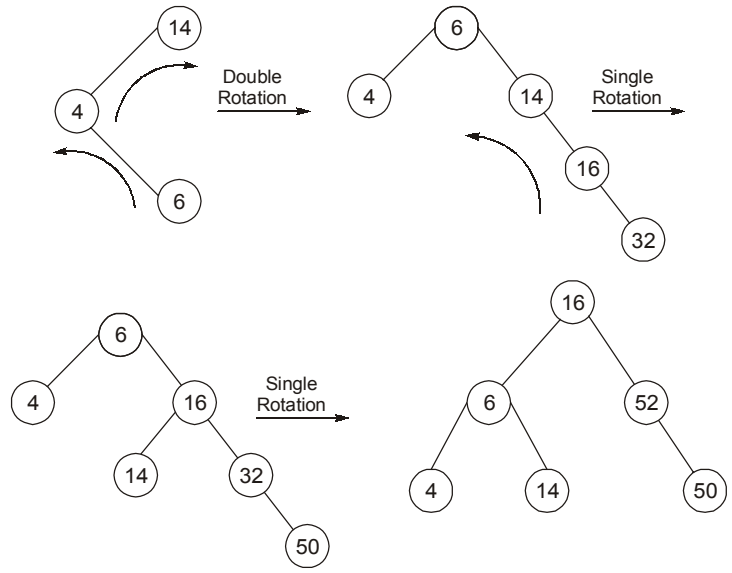
16. (b)

This program finds a repeated number (duplicate) in the list. This program adds all the values in the variable "value" and subtract "k(k–1)/2" from it. Array has 0 to (k – 1) locations and numbers are from 1 to (k – 1). So one number will be repeated in the array.

17. (d)

f1 (3) = 2
f1 (4) = 3
f1 (5) = 6
f1 (6) = 9
2 + 3 + 6 + 9 = 20

18. (b)



One double and two single rotations are required. So total 3 rotations.

19. (a)

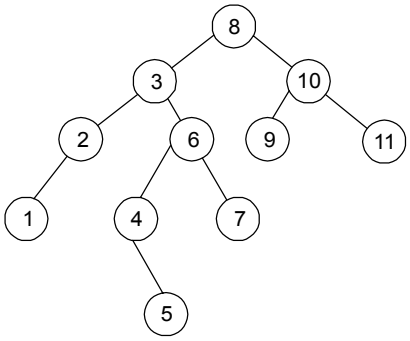
g(15) = 2 + g(10) + g(13)

g(10) = 13
g(13) = 25
} Compute recursively

∴ g(15) = 2 + 13 + 25 = 40.

20. (c)

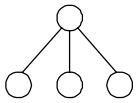
Final tree:

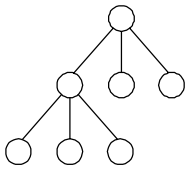


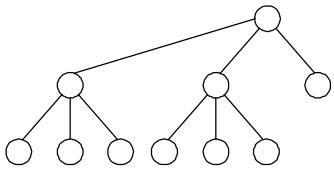
1, 4, 5 and 7 have height greater then node 6.

21. (a)

n → number of internal nodes :

Let n = 1  ⇒ 3 ⇒ 2 (1 – 1) + 3

Let n = 2  ⇒ 5 ⇒ 2 (2 – 1) + 3

Let n = 3  ⇒ 7 ⇒ 2 (3 – 1) + 3

Number of internal nodes =2 (n – 1) + 3 = 2 (19 – 1) + 3 = 39

22. (c)

S_1: Since, k items have already been inserted in the hash table. So, $\left(\frac{k}{n}\right)$ denotes that the key, hit a location out of 'k' location in which insertion have already been made.

$$\text{Probability} = \left(1-\frac{k}{n}\right)$$

S_2: This is correct because in case of collision, key probes to the next free location and to retrieve also. Offer not getting the element on its expected position, the key need to be searched further in the table.

23. (d)

The given lower triangular matrix can be represented as

$$\begin{array}{c|cccccc} & -6 & -5 & -4 & \ldots & +8 \\ -6 & a_{11} & & & & \\ -5 & a_{21} & a_{22} & & & \\ -4 & a_{31} & a_{32} & a_{33} & & \\ . & . & . & & & \\ . & . & . & & & \\ . & . & . & & & \\ . & . & . & & & \\ +8 & a_{81} & a_{82} & \ldots & \ldots & a_{88} \end{array}$$

Let (i, j) be the element to be accessed.

We must cross upto $(i-1)^{th}$ row.

Number of elements upto $(i-1)^{th}$ row or 10^{th} row

$$= 1 + 2 + 3 + \ldots\ldots + [(i-1) - (l_{bi}) + 1] \quad [l_{bi} \rightarrow \text{lower bound of } i]$$

$$= 1 + 2 + 3 + \ldots\ldots (3 - (-6) + 1)$$
$$= 1 + 2 + 3 + \ldots\ldots + (10)$$
$$= \frac{10 \times 11}{2} = 55$$

In i^{th} row we must cross $(j - l_{bj})$ elements. $[l_{bj} \rightarrow$ lower bound of j]

$$= 2 - (-6) = 8$$

$\therefore$ In total = 55 + 8 = 63 elements need to be crossed.

Resulted address = Base address + Number of element crossed

1000 + 63 = 1063

24. (c)

Number of (minimum) nodes=S (h – 1) + S (h – 2) + 1

Number of (maximum) nodes=$2^{h+1} - 1$

25. (c)

26. (b)

The while loop may some time leads to segmentation error. Segmentation error occurs when we access data from an address that does not exist. This can happen when we reach last node and assign the NULL value to temp in while loop.

So again when we access temp → data. This creates segmentation error.

```
While(temp→data!=y && temp→next!=NULL).
{
    q=temp;
     temp=temp →next;
}
```

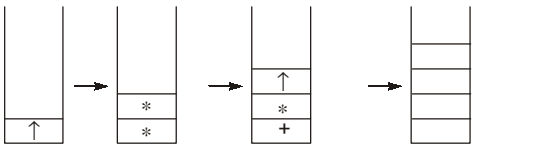
The above code is correct implementation of while Loop.

27. (d)

The output sequence printed by the given code is: 8, 9, 3, 5, 0, 6, 1, 7, 2, 8

Last output is : 8

28. (b)

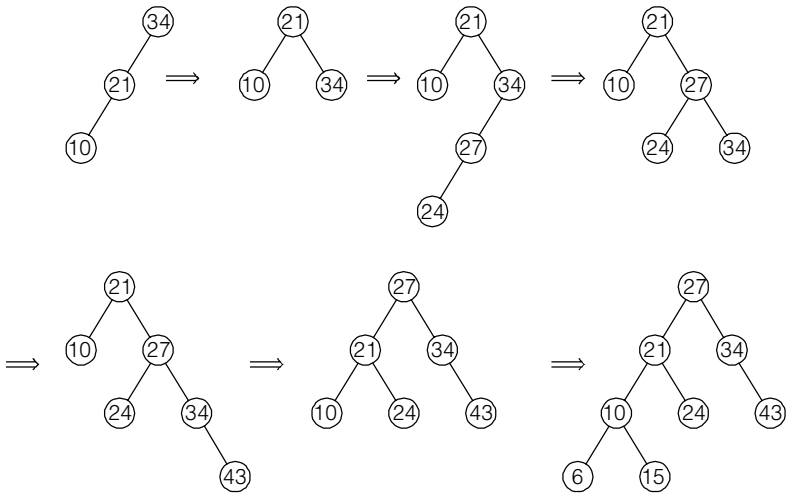


max size of stack = 3

29. (d)

- Random access is not allowed in linked list but allowed in array.
- Insertion and deletion in array is difficult, since we need to shifts other elements, while insertion and deletion in linked list in easy (only few pointers need to be charged in worst case both take $O(n)$ time).
- Since array are contiguous memory blocks, so large chunk of them will be loaded into the cache upon cache upon first access. Which make cache access faster for future elements of array i.e. gives better cache locality which is not possible in case of linked list.

30. (b)



∴ Root is 27.

■■■■