



Kunal Jha

Course: GATE
Computer Science Engineering(CS)

HOME

MY TEST

BOOKMARKS

MY PROFILE

REPORTS

BUY PACKAGE

NEWS

TEST SCHEDULE

TOPICWISE : PROGRAMMING AND DATA STRUCTURES-1 (GATE - 2020) - REPORTS

OVERALL ANALYSIS

COMPARISON REPORT

SOLUTION REPORT

ALL(17)

CORRECT(5)

INCORRECT(7)

SKIPPED(5)

Q. 1

Have any Doubt ?



What is the output of the following C-code? (Assume that the address of X is 2000 (in decimal) and one integer requires 4B of memory).

```
int main()
{
    unsigned int X[4][3] = {[1, 2, 3],
                           {4, 5, 6},
                           {7, 8, 9},
                           {10, 11, 12}};
    printf("%u %u %u", X + 3, *(X + 3), *(X + 2) + 3);
}
```

A 2036, 2036, 2036

Your answer is Correct

Solution :
(a)

2000	2000	2008
1	2	3
2012	2012	2020
4	5	6
2024	2028	2032
7	8	9
2036	2040	2044
10	11	12

$$X + 3 = 2000 + 3 * (12) = 2036$$

$$*(X + 3) = *(2000 + 3 * (12)) = 2036$$

$$*(X + 2) + 3 = *(2000 + 2 * (12)) + 3 = 2036$$

B 2012, 4, 2024

C 2036, 10, 10

D 2012, 4, 6

QUESTION ANALYTICS



Q. 2

Have any Doubt ?



Which of the following permutations can be obtained in the output (in the same order) using a stack assuming that the input is the sequence 5, 7, 8, 4, 6 in that order?

A 6, 8, 4, 7, 5

B 6, 4, 5, 7, 8

C 6, 4, 7, 8, 5

Your answer is Wrong

D 7, 8, 4, 6, 5

Correct Option

Solution :
(d)

(a) 6, 8, 4, 7, 5

6
4
8
7
5

After popping element 6, only 4 can be popped, hence this permutation is not possible.

(b) 6, 4, 5, 7, 8

6
4
8
7
5

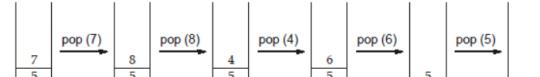
After performing pop operation on element 6, 4 now only element 8 can be popped.

(c) 6, 4, 7, 8, 5

6
4
8
7
5

After 6, 4 elements are popped, now element 7 can only be popped iff 8 has already been popped.

(d) 7, 8, 4, 6, 5



Q. 3

Have any Doubt ?



Which of the following is correct output for the program code given below?
code given below?

```
main ()
{
    void fun ();
    fun();
    fun();
}
void fun ()
{
    static int i = 1;
    auto int j = 5;
    printf ("%d", (i++));
    printf ("%d", (j++));
}
```

A 1 5 2 6 3 7

B 2 6 3 7 4 8

C 1 5 2 5

Your answer is Correct

Solution :

(c)

An object whose storage class is auto, is reinitialized at every function call whereas an object whose storage class static persist its value between different function calls.
When the function fun () is called for the first time, value of i and j are printed and sequentially incremented. During the second function call, i retains its incremented value whereas j is reinitialized, hence i will print 2 and j will print 5 again.

D 1 5 2 5 3 5

Q. 4

Have any Doubt ?



Consider the following program:

```
main ()
{
    int i, j;
    int A[m][n] = {{1, 2, 3}, {4, 5, 6}};
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            print f("%d", * (A[j] + i));
}
```

For the output printed by the above program.

A 1 2 3 4 5 6

Your answer is Wrong

B 1 4 2 5 3 6

Correct Option

Solution :

(b)

Here m represent the number of rows and n represents the number of column.

$$m = 2, n = 3$$

$$* (A[0] + 0) = A[0][0] = 1$$

$$* (A[1] + 0) = A[1][0] = 4$$

Similarly it will access all the element.

$\therefore 1 \ 4 \ 2 \ 5 \ 3 \ 6$ is the output printed by the program.

C 4 5 6 7 8 9

D 4 5 6 4 5 6

Q. 5

Have any Doubt ?



Which of the following statements is correct:

A Void pointers can be used for dereferencing.

B Arithmetic operations can be applied on void pointers.

C The default value of extern storage class is garbage value.

D A particular extern variable can be declared many times, but can be initialized at only 1 time.

Your answer is Correct

Solution :

- (d)
 - Void pointers can't be used for dereferencing because each variable type takes different amount of memory.
 - Since compiler can not know, after how many bytes is the next variables located, hence arithmetic operations can't be performed.
 - Default value of extern storage class is 0.

 QUESTION ANALYTICS



Q. 6





What will be the output of the following C program:

```
# include <stdio.h>
int print 1 (void)
{
    static int x = 10;
    x += 5;
    return x;
}
int print 2 (void)
{
    static int x;
    x = 10;
    x += 5;
    return x;
}
int main ( )
{
    int x;
    x = print 1();
    x += print 1();
    x += print 2();
    x += print 2 ();
    printf("%d", x);
    return 0;
}
```

The output of the program is _____.

 65

Correct Option

Solution :

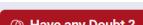
- 65
- `print 1()`: $x = 10 + 5 = 15$; since the variable is of static storage class, hence it will retain its value between different function calls.
 - `print 1()`: $x = 15 + 5 = 20$; since it has retained its value 15.
 - `print 2()`: x is defined again inside the function and hence will print, $x = x + 5 = 10 + 5 = 15$. Again when the function will be called, $x = 10 + 5 = 15$. Here second time also $x = 10$ will be there because it is not initialized at the time of definition.
- $$x = 15 + 20 + 15 + 15 = 65$$

Your Answer is 70

 QUESTION ANALYTICS



Q. 7





Consider the following program segment:

```
int main ( )
{
    char * str = "GATECS";
    printf ("%d", test (str));
    return 0;
}
int test (char * p1)
{
    char * p2 = p1;
    while (*+p1);
    return (p1 - p2);
}
```

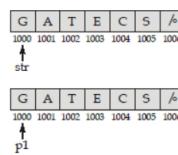
The output of the above program will be _____.

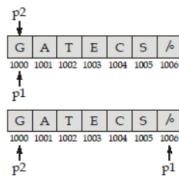
 6

Correct Option

Solution :

6





$$\frac{\text{Address of } p1 - \text{Address of } p2}{\text{Size}} = \frac{1006 - 1000}{1} = 6$$

Your Answer is 4

QUESTION ANALYTICS

Q. 8

Have any Doubt?



Consider the following function:

```
Void Test (int arr[ ], int s, int e)
{
    int temp;
    if(s ≥ e)
        return;
    temp = arr[s + 1];
    arr[s + 1] = arr[e];
    arr[e] = temp;
    Test(arr, s + 1, e - 1)
}
```

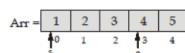
Let arr is an array [0 to 4] which initially holds elements {1, 2, 3, 4, 5}. Test (arr[], 0, 3) is called. The number of elements that will maintain their initial position after the end of the code are _____.

3

Your answer is Correct

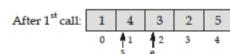
Solution :

3



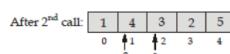
Test (arr [], 0, 3)

```
temp = arr [1] ⇒ temp = 2
arr [1] = arr [3] ⇒ arr [1] = 4
arr [3] = 2
```



Test (arr [], 1, 2)

```
temp = arr [2] ⇒ temp = 3
arr [2] = arr [2] = 3
arr [2] = 3
```



Index element 0, 2 and 4 do not change their positions.

QUESTION ANALYTICS



Q. 9

Have any Doubt?



Consider the following statements :

S_1 : Two stacks can be implemented by 1 array that grow from each end towards the middle.

S_2 : A queue can be implemented using two stacks with worst case time $O(n)$ for each enqueue and dequeue operation.

The number of statements correct are _____.

1

Correct Option

Solution :

1

Considering, both the statements:



- With a single array, two stacks can be implemented.
- Implement a queue using 2 stacks. Denote the two stacks S_1 and S_2 . The enqueue operation is simply implemented as a push on S_1 which will be in $O(1)$ time. But, while performing dequeue operation, element of S_1 will be moved to S_2 , then pop from S_2 and push to S_1 , each element of S_2 which takes $O(n)$ time. It can be vice versa too i.e. enqueue in $O(n)$ and dequeue $O(1)$ time.

QUESTION ANALYTICS



Q. 10

Have any Doubt?



The following C function takes a singly linked list as input argument. It prints its elements from the end using with the help of some other data structure. Some part of the code is left blank.

```
typedef struct node
{
    int value;
    struct node * next;
}
Node;
void printlist (node * head)
{
    if (! head) return;
```

Choose the correct alternative to replace the blank line.

A `printf ("%d", head → data);
printlist (head → next);`

B `while (node! = Null)
{ Node = Node → next
 printf("%d", node → data);
}`

C `printlist (head → next);
printf("%d", head → data);`

Your answer is Correct

Solution :

(c) Applying recursion, traverse the linked list till the last element, hence at every step, the element of the linked list will be stored in the stack. While coming back, print the elements, hence the elements will be start printing from the end.
if (!head) return;
 printlist (head → next);
 printf("%d", head → data);

D None of these

 QUESTION ANALYTICS





Kunal Jha
 Course: GATE
 Computer Science Engineering(CS)

- HOME
- MY TEST
- BOOKMARKS
- MY PROFILE
- REPORTS
- BUY PACKAGE
- NEWS
- TEST SCHEDULE

TOPICWISE : PROGRAMMING AND DATA STRUCTURES-1 (GATE - 2020) - REPORTS

OVERALL ANALYSIS COMPARISON REPORT **SOLUTION REPORT**

ALL(17) CORRECT(5) INCORRECT(7) SKIPPED(5)

Q. 11

Have any Doubt ?



Consider the following C program segment:

```
# include <stdio.h>
main()
{
    static char *s[ ] = {"madeeasy", "online", "test", "series"};
    char **ptr[ ] = {s + 3, s + 2, s + 1, s}, ***p;
    p = ptr;
    ++p;
    printf("%s", * -- *++p + 3)
}
```

What will be printed by the program?

A line

B ies

C test

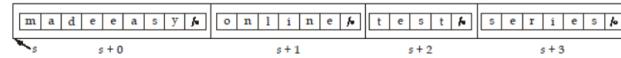
D easy

Correct Option

Solution :

(d)

In this problem we have an array of char pointers pointing to start of 4 strings i.e.,



We have ptr which is pointer to a pointer of type char and a variable p which is a pointer to a pointer of type char.



$p = \text{ptr}; \quad p[\text{ptr}]$

$\text{++}p; \quad p[\text{ptr}+1]$

$\text{printf}("%s", * -- * \text{++}p + 3);$

In printf statement the expression is evaluated $\text{++}p$ cause gets value $(s + 1)$ then now pre-decrement is executed and we get $(s + 1) - 1 = s$. The indirection pointer now gets the value from the array of s and add 3 to the starting address. The string is printed starting from this position. Thus, the output is 'eeasy'.

QUESTION ANALYTICS



Q. 12

Have any Doubt ?



Consider an efficient implementation of a data structure STACK-MAX that support an operation "max()" that reports the current maximum among all elements in the stack. Normal stack operations i.e., push, pop are also to be supported. What is the size of above data structure after performing following operation push (5), push (6), push (7), pop, max, push (6), push (8), pop, pop, max, push (5)? (Assume that an integer can be stored in 4 bytes)

A 20 bytes

B 24 bytes

Correct Option

Solution :

(b)

Implement the stack where each entity stores two values:

1. Value = Current number.
2. CurMax = maximum of current number and numbers below the current number.

To implement:

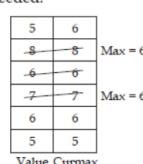
Push: If stack size is 0, add an entry with value = current number and curmax = current. If stack size >0 add an entry with value = current number and curmax = max (current number, curmax of top value on stack).

Pop: Same as normal stack.

Max: Return curmax of top entry on stack.

Every entry will be of 8 B.

After all the operation 24 B are needed.



C 18 bytes

D 28 bytes

QUESTION ANALYTICS

Q. 13

Have any Doubt ?

Consider the following C program:

```
# include <stdio.h>
int main()
{
    int a = 1;
    int b = 1;
    int c = a++ || b--;
    printf("%d %d", c, b)
}
```

Which of the following represents output of above C program?

A 1, 0

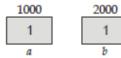
B 2, 0

C 1, 1

Correct Option

Solution :

(c)



int c = a++ || b--;
means c = a [since a++ is return true]
So no need to calculate b--, called short circuiting.
So, print will be c = a. So, c = 1 and b = 1.
So, output is 1, 1.

D 2, 1

Your answer is Wrong

QUESTION ANALYTICS

Q. 14

Have any Doubt ?



A circular array based queue 'q' is capable of holding 6 elements. If the array is initially empty and array has indices from 0 to 4.

```
for (x = 1; x <=5; x++)
q.enqueue (x);
for (x = 1; x <= 4; x++)
{
    q.dequeue( );
    q.enqueue (q.dequeue( ))
}
```

Assume enqueue and dequeue are circular queue operations for insertion and deletion respectively. The number of elements that will remain in the queue after the complete execution of the program is/are

A 0

B 1

Correct Option

Solution :

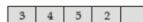
(b)

Initially, on the execution of first for loop, all the elements will be enqueued.



At the second loop execution,

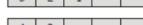
x = 1 \Rightarrow dequeue 1; enqueue (2)



x = 2 \Rightarrow dequeue 2; enqueue (3)



x = 3 \Rightarrow dequeue 3; enqueue (4)



x = 4 \Rightarrow dequeue 4; enqueue (5)



Element '2' is present in the queue.

C 2

D 3

QUESTION ANALYTICS

Q. 15

Have any Doubt ?



Consider the following program along with push and pop operations on stack which can contain atmost 8 element at a time:

```
void main ()
{
    stack S;
    int num;
    printf ("enter the input");
    scanf ("%d", & num);
    while (num != 0)
    {
        if (!full (S))
        {
            push (S, num % 2);
            num = num / 2;
        }
        else
        {
            printf ("stack overflow");
            exit (0);
        }
    }
}
```

The value 156 is given as input to the program then value present in stack from top to bottom will be ____.

10011100

Correct Option

Solution :

10011100

The given program compute the binary value of decimal number 156.
Hence, the output received will be 10011100.



Your Answer is 0011100

QUESTION ANALYTICS



Q. 16

Have any Doubt ?



The output of following program is _____.

```
# include <stdio.h>
int main()
{
    static int a[ ] = {90, 98, 99, 96, 84, 70};
    static int *p[ ] = {a + 2, a + 1, a, a + 3, a + 4, a + 5};
    static int **S = {p + 4, p + 5, p + 1, p, p + 2, p + 3};
    int ***ptr; ptr = S + 2;
    printf("%d", ***ptr + 3) - ***(ptr + 1));
}
```

-2

Correct Option

Solution :

-2

a	90	98	99	96	84	70
	a+0	a+1	a+2	a+3	a+4	a+5
p	a+2	a+1	a	a+3	a+4	a+5
	p+0	p+1	p+2	p+3	p+4	p+5
S	p+4	p+5	p+1	p	p+2	p+3
	S+0	S+1	S+2	S+3	S+4	S+5
Ptr	S+2					

$$\begin{aligned} ***(\text{ptr} + 3) - ***(\text{ptr} + 1) &= (*(*(*(\text{S} + 3 + 2)))) - (*(*(\text{p} + 1))) \\ &= (*(*(\text{p} + 3))) - (*(*(\text{a} + 1))) \\ &= 96 - 98 = -2 \end{aligned}$$

QUESTION ANALYTICS



Q. 17

Have any Doubt ?



What is the output of the following program?

```
# include <stdio.h>
# define MUL (a, b) a * b
# define pow (a) a * a
int main ()
{
    int a = 3;
    int b = 2;
    printf ("%", MUL (MUL (a + 1, b), pow (b + 1)));
    return 0;
}
```

10

Correct Option

Solution :

10

MUL (MUL (a + 1, b), pow (b + 1))
MUL ([a + 1 * b], [b + 1 * b + 1])

$$\begin{aligned}\Rightarrow & \quad a + 1 * b * b + 1 * b + 1 \\ \Rightarrow & \quad 3 + 1 * 2 * 2 + 1 * 2 + 1 \\ \Rightarrow & \quad 3 + 4 + 2 + 1 = 10\end{aligned}$$

Your Answer is 72

 QUESTION ANALYTICS

+

Item 11-17 of 17 « previous 1 2 next »



Kunal Jha
 Course: GATE
 Computer Science Engineering(CS)

HOME

MY TEST

BOOKMARKS

MY PROFILE

REPORTS

BUY PACKAGE

NEWS

TEST SCHEDULE

TOPICWISE : PROGRAMMING AND DATA STRUCTURES-2 (GATE - 2020) - REPORTS

OVERALL ANALYSIS COMPARISON REPORT **SOLUTION REPORT**

ALL(17) CORRECT(7) INCORRECT(3) SKIPPED(7)

Q. 1

Have any Doubt ?



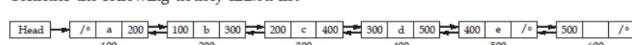
In a doubly linked list organization, insertion of a record in end involves modification of _____ for existing list.

A one pointer

Your answer is Correct

Solution :

(a) Consider the following doubly linked list



After the insertion, only one pointer that is the next of node 'e' is modified to 600.

Two more pointers are added in the list, but in the existing list only one pointer is modified.

B two pointer

C multiple pointer

D no pointer

QUESTION ANALYTICS



Q. 2

Have any Doubt ?



A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

A $\log_2 n$

B $n - 1$

Correct Option

Solution :

(b)

Because for 2 degree node, every time 2 leafs are added then number of nodes with degree 2 increases by 1.
 Thus, number of nodes with degree 2 is always one less than number of leafs present in tree.

C n

Your answer is Wrong

D 2^n

QUESTION ANALYTICS



Q. 3

Have any Doubt ?



Consider M_1 and M_2 be two complete binary tree which satisfy max-heap property, each of size ' n '. What is the time complexity to combine both M_1 and M_2 such that combine tree will be min heap tree?

A $O(n \log n)$

B $O(n)$

Correct Option

Solution :

(b)

Copy both M_1 and M_2 's element in new array of size $2n$, then apply Build heap method to make min heap tree which take $O(2n) \cong O(n)$ time.

C $O(n^2)$

D $O(n^2 \log n)$

QUESTION ANALYTICS



Q. 4

Have any Doubt ?



Consider the following C program segment where node represents a node in binary search tree.

```
struct node  
{  
    int data;  
    struct node * left;  
    struct node * right;  
};  
int BST (struct node * node)  
{  
    if (node == NULL)  
        return 0;  
    If (node → left == NULL && node → right == NULL)  
        return (node → data);  
    else  
        return (BST (node → left) + BST (node → right));  
}
```

The value returned by BST when a pointer to root of a binary search tree is passed as its argument is

A Sum of all nodes data

B Sum of leaf nodes data

Your answer is Correct

Solution :

(b)

Consider the statement If(node==NULL) then it returns 0 i.e. tree does not contain any node. If(node → left==NULL && node → right==NULL) node → data it means if a node does not contain any left or right child, i.e., leaf node it returns that leaf node value. Consider else statement return (BST (node → left) + BST (node → right)) means if a node contains left and right subtree then recursively, find the leaf node value in the left subtree as well as right subtree and return the sum of values of leaf nodes.

C Sum of internal nodes data

D None of these

QUESTION ANALYTICS



Q. 5

Have any Doubt ?



Consider the following recursive function.

```
bool f(Struct node *P)  
{  
    if(P = NULL) return TRUE;  
    if(P → Left != NULL && Max(P → Left) > P → data)  
        return FALSE;  
    if(P → right != NULL && Min(P → right) <= P → data)  
        return FALSE;  
    if(!f(P → Left)||!f(P → right))  
        return FALSE;  
    return TRUE;  
}
```

Assume Max(q) function returns the maximum value from q and subtrees of q, Min(q) function returns the minimum value from q and subtrees of q. If root of the binary tree is passed to the function f(), then what is the functionality of the above code?

A It checks if a given tree is a binary search tree or not.

Your answer is Correct

Solution :

(a)

Given function f() checks whether a given tree is BST or not. It returns 'TRUE' if a tree is BST, otherwise it returns 'FALSE'.

B It checks if a given tree is a heap tree or not.

C It checks if given tree is AVL tree or not.

D None of these

QUESTION ANALYTICS



Q. 6

Have any Doubt ?



The number of BST possible with 6 nodes numbered 1, 2, 3, 4, 5 and 6 with exactly one leaf node _____.

32

Correct Option

Solution :

32

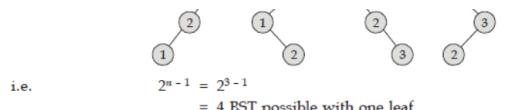
When n = 2 then



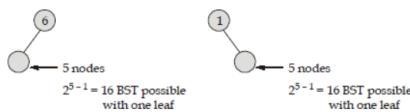
i.e. $2^{n-1} = 2^2 - 1$
 $= 2$ BST possible with one leaf

When n = 3 then





Similarly when $n = 6$



Total $2^{5-1} + 2^{5-1} = 2^{5-1}[1+1] = 2^5 = 32$

Your Answer is 2

QUESTION ANALYTICS

+

Q. 7

Have any Doubt?

Bookmark

A 4-ary i.e., either has 0 children or has 4 children tree has 20 leaf nodes. Then the total number of nodes in the tree are _____.

27

Correct Option

Solution:
27

$$\text{The total number of nodes} = \frac{nm-1}{n-1}$$

where, m is number of leaf nodes and n be ary of the tree.
Hence, substituting the values,

$$\begin{aligned} \text{Number of nodes} &= \frac{20 \times 4 - 1}{4 - 1} = \frac{80 - 1}{3} = \frac{79}{3} = 26.3 \\ &\approx 27 \text{ nodes} \end{aligned}$$

Your Answer is 28

QUESTION ANALYTICS

+

Q. 8

Have any Doubt?

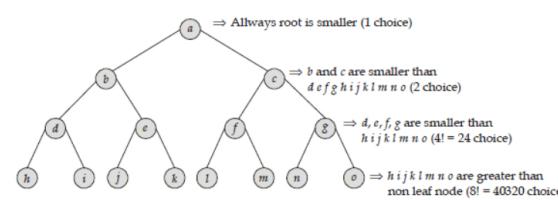
Bookmark

The number of min heap trees are possible with 15 distinct elements such that every leaf node must be greater than all non-leaf nodes of the tree are _____.

1935360

Correct Option

Solution:
1935360



$$\therefore \text{All possibility} = 40320 \times 24 \times 2 \times 1$$

$$= 1935360$$

QUESTION ANALYTICS

+

Q. 9

Have any Doubt?

Bookmark

Consider the following array of elements <40, 35, 20, 10, 15, 16, 17, 8, 4, 30>. The minimum number of interchanges needed to convert it into a min-heap is _____ using built heap method.

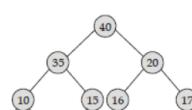
6

Correct Option

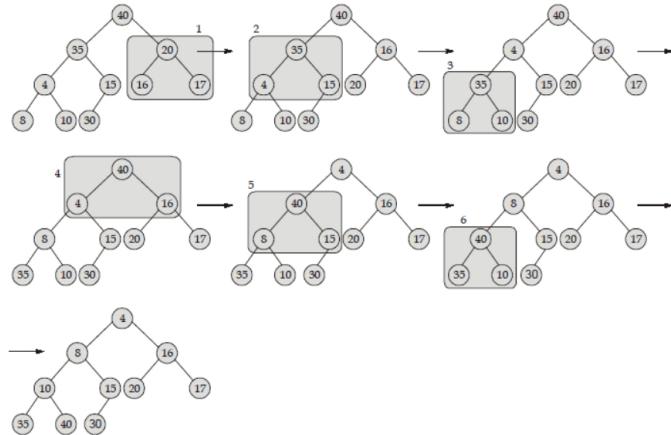
Solution:
6

40	35	20	10	15	16	17	8	4	30
----	----	----	----	----	----	----	---	---	----

Constructing the Binary Tree,



In Build-heap method, it is assumed that the leaf nodes are satisfying heap property and heapify procedure starts from second last level.



QUESTION ANALYTICS



Q. 10

Have any Doubt ?



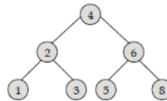
Assume that node l , whose key is k_l , is a leaf node of a Binary Search Tree 'T' having total number of nodes greater than 2 and that its parent is node P with key k_p then

- A k_p is the smallest key greater than k_l
- B k_p is the longest key smaller than k_l
- C Either (a) or (b)

Your answer is **Correct**

Solution :

(c)
Consider the following Binary Search Tree,



Consider 2 Scenarios :

Scenario 1: $k_l = 1; k_p = 2$

Here, k_p is the smallest key greater than k_l

Scenario 2: $k_l = 3; k_p = 2$

Here, k_p is the longest key which is smaller than k_l

Hence, either of the two is possible depending on the key.

- D Only (a) but not (b)

QUESTION ANALYTICS



Item 1-10 of 17 « previous 1 2 next »



Kunal Jha

 Course: GATE
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

TOPICWISE : PROGRAMMING AND DATA STRUCTURES-2 (GATE - 2020) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(17\)](#)
[CORRECT\(7\)](#)
[INCORRECT\(3\)](#)
[SKIPPED\(7\)](#)
Q. 11
[Have any Doubt ?](#)


Consider the following program implemented on tree T where T is

```

typedef struct tnode {
    int key;
    struct tnode * left, * right;
} * Node;

Void search (Node * node)
{
    struct Node * ptr, * q;
    if (node == Null) return;
    search (node → left);
    search (node → right);
    ptr = node → right;
    node → right = new Node (node → data);
    node → right → right = ptr;
}
    
```

If the root of the above tree is passed to the above function, what is the level order traversal of output tree produced by above function?

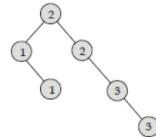
Note: newNode is a function which creates new node.

A 2 2 1 1 3 3

B 2 2 3 3 1 1

C 2 1 2 1 3 3

Correct Option

Solution :
 (c)


This will be the tree after execution of above code.

The level order traversal of the tree will be 2, 1, 2, 1, 3, 3.

D 2 1 2 1 2 3

[QUESTION ANALYTICS](#)

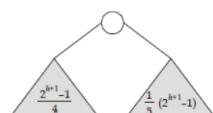
Q. 12
[Have any Doubt ?](#)


Consider the AVL tree 'T' in which left subtree contain quarter of the maximum number of nodes possible in the balanced AVL tree of height 'h' and right subtree consist of one fifth of the maximum number of nodes possible in AVL tree of height 'h'. Assume that tree 'T' may or may not be height balanced at present. What is the total maximum possible number of nodes in 'T'.

A $\frac{7}{20}(2^{h+1}-1)-1$

B $\frac{9}{20}(2^{h+1}-1)+1$

Your answer is Correct

Solution :
 (b)


$$\begin{aligned}
 \text{Total} &= \left(\frac{2^{h+1}-1}{4}\right) + \frac{1}{5}(2^{h+1}-1) + 1 \\
 &= \frac{9}{20}(2^{h+1}-1)+1
 \end{aligned}$$

C $\frac{7}{30}(2^{h+1}+1)$

D $\frac{9}{30}(2^{h+1}+1)-1$

Q. 13

Have any Doubt?



The following C function takes single linked list as parameter and rearrange the elements of the list. The function is called with list containing integers 1, 2, 3, 4, 5, 6 in order.

```
struct node
{
    int val;
    struct node * next;
}

void arrange (struct node * list)
{
    struct node * p, * q;
    int temp;
    if(!list || !list->next) return;
    p = list;
    q = list->next;
    while (P)
    {
        temp = p->val;
        q->val = temp;
        p->val = q->val;
        q = p? p->next: 0;
        p = q? q->next;
    }
}
```

What will be the content of the list after function completes execution?

A 2, 1, 4, 3, 6, 5

B 1, 3, 3, 5, 5, 6

Your answer is Correct

Solution:

(b)

Consider the linked list.



1st function call: temp = 1

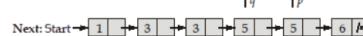
q->val = 1

p->val = q->val = 1

q = p, true

q = p->next

p = q->next



Final output: 1, 3, 3, 5, 5, 6.

C 1, 1, 1, 1, 1, 6

D 1, 1, 1, 1, 1, 1

Q. 14

Have any Doubt?



Consider the following C program segment:

```
struct node
{
    struct node * left, right;
    int data;
}

int Target (struct node * root)
{
    if(root != NULL)
    {
        int a = root->data;
        int b = Target (root->left);
        int c = Target (root->right);
        if (b > a)
            a = b;
        if(c > a)
            a = c;
    }
    return a;
}
```

The value returned by the function Target when a pointer to the root of a non-empty binary tree is passed as an argument is

A Maximum number of nodes in the left subtree

B Height of the tree

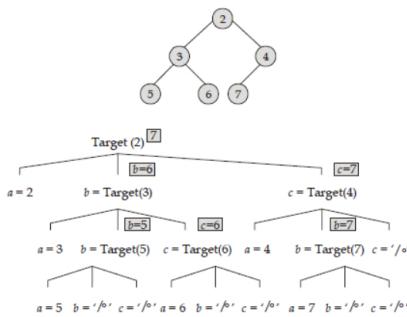
C Diameter of the tree

D None of the above

Correct Option

Solution :
(d)

Consider the following tree:



Output is 7, which is the maximum data of all the tree nodes data.

QUESTION ANALYTICS

+

Q. 15

Have any Doubt ?

?

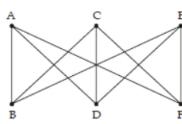
The maximum possible height of BFS tree, if BSF is run on a complete bipartite graph $K_{m,n}$, where $m \geq 1, n \geq 1$ with starting vertex 'S' is _____.

C 2

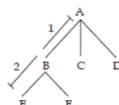
Correct Option

Solution :
2

Since bipartite graph can be grouped into 2 group of vertices i.e., $K_{3,3}$.



Take start vertex is A.



Max height of above BFS traversal is 2.

QUESTION ANALYTICS

+

Q. 16

Have any Doubt ?

?

Consider a binary tree whose pre-order and in-order traversals are CBAFDEHGJI AND ABCDEFGHIJ. What will be the minimum number of rotations to convert the tree into an AVL tree?

C 1

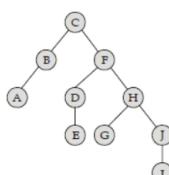
Correct Option

Solution :
1

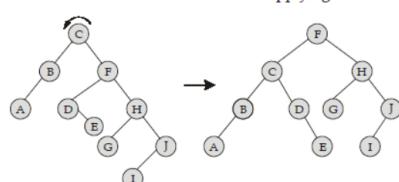
Constructing BST with the help of pre-order and in-order traversals:

Pre-order: CBAFDEHGJI

In-order: ABCDEFGHIJ



The tree become unbalanced at the root node. Hence applying RR rotation, we get



Hence after 1 rotation the tree travers into an AVL tree.

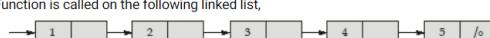
Q. 17

[Have any Doubt ?](#)

The following C function takes a singly linked list as input argument.

```
struct node {int value, struct node * next};  
int find (Node * head)  
{  
    Node * P1 = head, * P2 = head;  
    if (head -> next != NULL)  
    {  
        P1 = head -> next;  
        P2 = (head -> next)? head -> next -> next : NULL;  
    }  
    while ((P1 != NULL) && (P2 != NULL))  
    {  
        if (P1 == P2) return 1;  
        P1 = P1 -> next;  
        P2 = (P2 -> next != NULL)? P2 -> next -> next : NULL;  
    }  
    return 0;  
}
```

Function is called on the following linked list,



The value returned by the code is _____.

 0

Your answer is Correct!

Solution :

0

Above code returns '1' if there is a cycle in a linked list. Since, the linked list which is given as an argument does not contain any cycle. Hence, value '0' will be returned.



Kunal Jha

 Course: GATE
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

SINGLE SUBJECT : PROGRAMMING AND DATA STRUCTURES (GATE - 2020) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(33\)](#)
[CORRECT\(14\)](#)
[INCORRECT\(9\)](#)
[SKIPPED\(10\)](#)
Q. 1
[Have any Doubt ?](#)


Consider ' T ' to be a ' n ' - ary tree (each internal node has at most ' n ' children). Suppose the depth of ' T ' is ' k '. Which of the following correctly represents the maximum number of leaves that ' T ' can have?

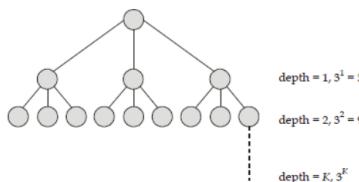
A n^k

 Your answer is **Correct**
Solution :

(a)

 Number of children by every node = n

 depth of tree = k

 Let $n = 3$


Hence the maximum number of leaves that ' T ' can have is n^k .

B k^n
C nk
D None of these

QUESTION ANALYTICS

Q. 2
[Have any Doubt ?](#)


An $n \times n$ where ' n ' ranging from {1 to n } matrix ' r ' is defined as follows:

 $r[i, j] = i$, if ($i = j$)

 $r[i, j] = i^2 - j^2$, if ($i < j$)

 $r[i, j] = j^2 - i^2$, if ($i > j$)

 The sum of the elements of the array ' r ' is

A 0

 Your answer is **Wrong**
B n^2
C $\frac{n(n+1)}{2}$
D None of these

Correct Option

Solution :

(d)

Let

 $n = 3$

$$r = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & -6 \\ -3 & 2 & -5 \\ -8 & -5 & 3 \end{bmatrix}$$

$$1 + 2 + 3 + 2 \times (-3 - 8 - 5) = -26 \neq \frac{n(n+1)}{2}$$

Alternate Method:

$$r[i, j] = \begin{cases} i^2 - j^2; & \text{if } (i < j) \\ j^2 - i^2; & \text{if } (i > j) \end{cases}$$

Both these statements will generate -ve values.

or, we can also say

$$r[i, j] = -|i^2 - j^2|$$

 Further $|i^2 - j^2| > i$

$$\Rightarrow \sum_{-ve} |i^2 - j^2| > \sum_{+ve} i$$

So, summation of all elements will always be -ve.

QUESTION ANALYTICS


Q. 3

[Have any Doubt ?](#)

Consider a C language variables a and b with initial values as 0 and 1 respectively. What will be the value of b after executing the following expression?
 $((a > 0) \&\& (b = b + 1))$

 A 2 B 1

Your answer is Correct

Solution :

(b)
 In 'AND' operation second condition is executed only when first condition is true.
 Here a is 0 and $(0 > 0)$ is false so $(b = b + 1)$ is not executed. So b will be 1.

 C 0 D 3

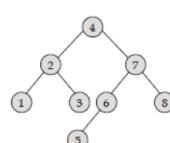
Q. 4

[Have any Doubt ?](#)

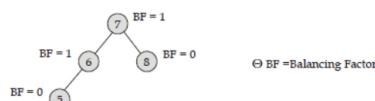
Let ' T ' be a rooted Binary tree above vertices are labelled with symbols 1, 2, 3, 4, 5, 6, 7, 8.
 Suppose the inorder and postorder traversal of T produces the following sequences.
 In-order: 1, 2, 3, 4, 5, 6, 7, 8
 Post-order: 1, 3, 2, 5, 6, 8, 7, 4
 Which of the following is correct with respect to T ?

 A ' T ' has 5 leaf nodes B Right subtree of the root node satisfies AVL property

Your answer is Correct

Solution :
 (b)
 The tree constructed will be,

- ' T ' has 4 leaf nodes.
- Subtree rooted at node '7' satisfies the AVL property.



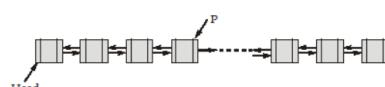
- In a heap, nodes are started from most left pointer, hence node '5' is violating the basic heap property.

 C ' T ' satisfies basic heap properties D None of these

Q. 5

[Have any Doubt ?](#)

Consider a link list containing n nodes given below:

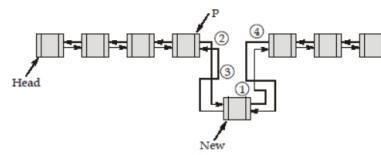


We want to insert a new node in the given linked list after node P . Which of the following sequence of operation is correct?

- | | |
|-----------------------------------------------------|-----------------------------------------------------|
| (a) 1. $new \rightarrow next = P \rightarrow next;$ | (b) 1. $new \rightarrow next = P \rightarrow next;$ |
| 2. $new = P \rightarrow next;$ | 2. $new \rightarrow next = P;$ |
| 3. $new \rightarrow prev = P;$ | 3. $new \rightarrow prev \rightarrow P;$ |
| 4. $(P \rightarrow next) \rightarrow prev = next;$ | 4. $(new \rightarrow next) \rightarrow prev = new;$ |
- | | |
|-----------------------------------------------------|---------------------------------------------------|
| (c) 1. $new \rightarrow next = P \rightarrow next;$ | (d) 1. $new \rightarrow next \rightarrow next;$ |
| 2. $P \rightarrow next = new;$ | 2. $P \rightarrow next = new;$ |
| 3. $new \rightarrow prev = P;$ | 3. $new \rightarrow prev = P;$ |
| 4. $(new \rightarrow next) \rightarrow prev = new;$ | 4. $(P \rightarrow next) \rightarrow prev = new;$ |

 A a B b

Solution :
(c)



1. $\text{new} \rightarrow \text{next} = \text{P} \rightarrow \text{next};$
2. $\text{P} \rightarrow \text{next} = \text{new};$
3. $\text{new} \rightarrow \text{prev} = \text{P};$
4. $(\text{new} \rightarrow \text{next}) \rightarrow \text{prev} = \text{new};$

D d

QUESTION ANALYTICS

+

Q. 6

Have any Doubt ?

Q

A stack is implemented using a circular doubly linked list such that PUSH and POP operations are performed efficiently. Which one of the following statements can be correct?

A Both operations can be performed in $O(1)$ time.

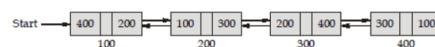
Your answer is Correct

Solution :
(a)

Consider a stack

4
3
2
1

The characteristic of the stack is both insertions and deletions are performed from one end.
If, it is implemented with a link lists, then both insertions and deletions are needed to be performed from the end.
Since, the linked list is a doubly circular linked list, hence the start node will have address of last node.



So, both the operations can be performed in $O(1)$ time.

B The worst case time complexity for both operations will be $\Omega(n)$.

C Stack can not be implemented with gives data structure.

D None of these

QUESTION ANALYTICS

+

Q. 7

Have any Doubt ?

Q

Consider the following C program:

```
main ()
{
    int p = -3, q = 2, r = 0, s, t;
    s = ++p && ++q || r++;
    t = p + q + s++;
    printf("\n %d %d", s, t);
}
```

Which of the following represents output of above program?

A 1 2

Correct Option

Solution :
(a)

Initial value are $p = -3$
 $q = 2$
 $r = 0$
 $\&&$ has more priority than $++$
 $++p = -2$
 $++q = 3$

Since, both are non zero, hence expression becomes true. $r++$ need not be checked for calculating 's' because it's an OR operation so $s = 1$ i.e. the truth value of the expression.

$$\begin{aligned} t &= p + q + s++ \\ &= -2 + 3 + 1 = 2 \end{aligned}$$

B 0 2

Your answer is Wrong

C 1 3

D 0 3

Q. 8

Have any Doubt?



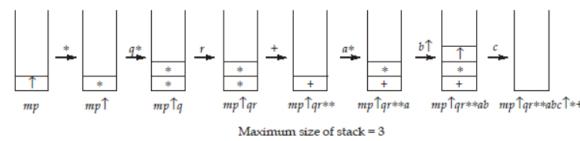
The maximum size of operator stack, when converting the following infix expression to postfix expression?
(Assume that \uparrow as the highest precedence and follows right associativity)

Infix: $m \uparrow p * q * r + a * b \uparrow c$

A 2**B** 3

Your answer is Correct

Solution :
(b)

**C** 4**D** 5

Q. 9

Have any Doubt?



Consider the following elements inserted into an empty AVL tree in the following order:

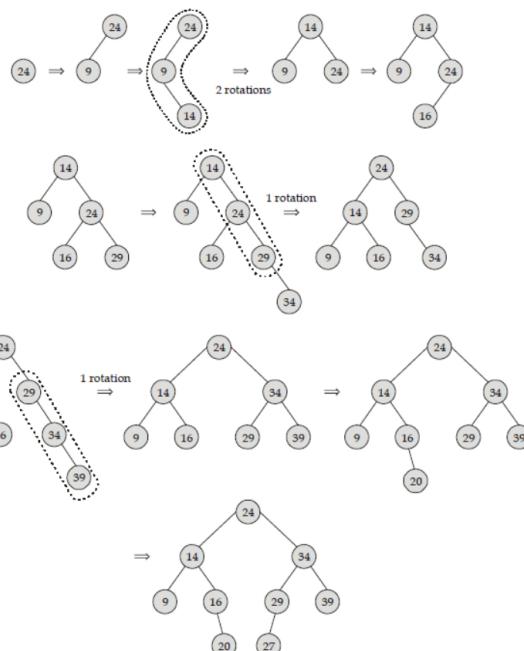
24, 9, 14, 16, 29, 34, 39, 20, 27

Total number of rotation performed for the construction are _____. [Consider RL and LR as 2 rotation each]

A 2**B** 3**C** 4

Your answer is Correct

Solution :
(c)

**D** 5

Q. 10

Have any Doubt?



Consider the following program segment prints:

```
# include <stdio.h>
void fun (int *a, int *b)
{
    a = b;
    *a += 2;
    a = b;
}
int p = 50, q = 20;
int main()
{
    fun (&p, &q);
    fun(&p, &p);
    fun(&q, &q);
    printf ("%d %d", p, q);
    return 0;
}
```

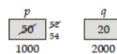
Which of the following represents the values printed by above program segments?

A 50, 22

B 52, 24

Your answer is Correct

Solution :
(b)



1st function call pass parameters as call by reference



$a = b \Rightarrow$ Now a is also pointing to 2000 address.

$*a += 2 \Rightarrow *a = *a + 2$

$\Rightarrow *a = 20 + 2 = 22$

$a = b \Rightarrow$ Now a is also pointing to 2000 address.

2nd function call by reference



$a = b$ 'a' will now store b value which is 1000, already contain by 'a'.

• $*a += 2;$

$*a = *a + 2;$

$*a = 50 + 2$

$*a = 52$

• $a = b$

'a' will now store b value which is 1000, already contain by 'a'.



$a = b \Rightarrow$ Now a is also pointing to 2000 address.

$*a += 2 \Rightarrow *a = *a + 2$

$\Rightarrow *a = 22 + 2 = 24$

$a = b \Rightarrow$ Now a is also pointing to 2000 address.

So, output will be 52 and 24.

C 54, 20

D None of these

QUESTION ANALYTICS





Kunal Jha

 Course: GATE
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

SINGLE SUBJECT : PROGRAMMING AND DATA STRUCTURES (GATE - 2020) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(33\)](#)
[CORRECT\(14\)](#)
[INCORRECT\(9\)](#)
[SKIPPED\(10\)](#)
Q. 11
[Have any Doubt ?](#)


Consider the following 'rec' function.

```
rec (int x)
{
    static int f;
    if (x == 1)
        return (1);
    else
        f+= x * rec (x - 1);
    return (f);
}
```

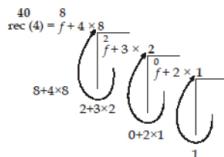
The value returned by rec (4) is _____.

40

 Your answer is **Correct** 40

Solution :

40



Returning from x = 1 to 5, since 'f' is declared as static variable so memory will be allocated at compiled time, so updated value = 40 + 5 * 40 = 240.

QUESTION ANALYTICS

Q. 12
[Have any Doubt ?](#)


Consider the following function:

```
int CALL (int K)
{
    if (K < 3)
        return 1;
    else
        return CALL (K - 1) + CALL (K - 3) + 1;
}
```

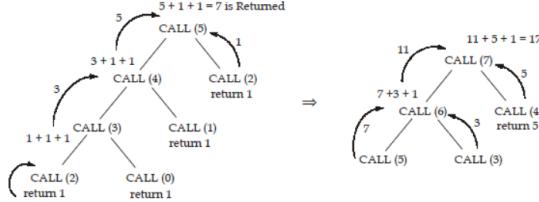
The value returned by CALL (CALL (5)) is _____.

17

Correct Option

Solution :

17



Your Answer is 19

QUESTION ANALYTICS

Q. 13
[Have any Doubt ?](#)


Consider a two-dimensional array with elements stored in the form of lower triangular matrix.

 The elements must be crossed to read A[4, 2] from the array A[-6, ..., +8, -6, ..., +8] whose base address 1000 is _____.
 (Assume elements are stored in row major order).

1063

Correct Option

Solution :

1063

The given lower triangular matrix can be represented as

$$\begin{bmatrix} -6 & -5 & -4 & \dots & +8 \\ -6 & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \vdots \end{bmatrix}$$

$$\begin{array}{c} \sim \\ -5 \\ -4 \\ \vdots \\ +8 \end{array} \left| \begin{array}{cccccc} & a_{11} & & & & \\ a_{21} & a_{22} & & & & \\ a_{31} & a_{32} & a_{33} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ a_{81} & a_{82} & \dots & \dots & a_{88} & \end{array} \right|$$

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

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

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

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

$$= 1 + 2 + 3 + \dots + (3 - (-6)) + 1$$

$$= 1 + 2 + 3 + \dots + (10)$$

$$= \frac{10 \times 11}{2} = 55$$

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

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

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

Resulted address = Base address + Number of element crossed

$$1000 + 63 = 1063$$

QUESTION ANALYTICS



Q. 14

Have any Doubt ?



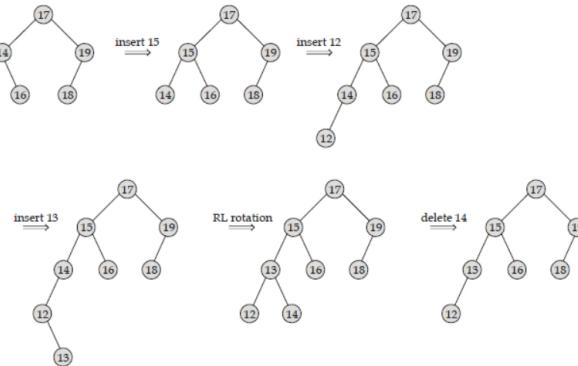
An AVL tree is constructed by inserting the following sequence of elements into empty AVL tree. After building AVL tree, if an element 14 is deleted from it, then number of levels present in AVL tree is _____. (Assume root is at 1 level)
17, 14, 19, 16, 18, 15, 12, 13

Correct Option

4

Solution :

4



Deletes 14 does not decrease any level so 4 levels present.

Your Answer is 3

QUESTION ANALYTICS



Q. 15

Have any Doubt ?



The output of following program is _____.

```
# include <stdio.h>
int main()
{
    static int a[] = {155, 13, 45, 98, 48, 110};
    static int *p[] = {a + 2, a + 1, a, a + 3, a + 4, a + 5};
    static int **S = {p + 4, p + 5, p + 1, p, p + 2, p + 3};
    int ***ptr; ptr = S + 3;
    printf("%d", ***ptr + 2) - **(p + 2));
}
```

-57

Your answer is Correct-57

Solution :

-57

a	155	13	45	98	48	110
	a+0	a+1	a+2	a+3	a+4	a+5
p	a+2	a+1	a	a+3	a+4	a+5
	p+0	p+1	p+2	p+3	p+4	p+5
S	p+4	p+5	p+1	p	p+2	p+3
	S+0	S+1	S+2	S+3	S+4	S+5
Ptr	S+3					

$$\begin{aligned} ***(\text{ptr} + 2) - **(p + 2) &= (*(*(*S + 3 + 2))) - (*(*p + 2)) \\ &= (*(*(*p + 3))) - (*(*a)) \\ &= 155 + 211 - 155 \end{aligned}$$

$$= 98 - 155 = -57$$

 QUESTION ANALYTICS



Q. 16

Have any Doubt ?



Consider 3 dimensional Array A[90] [30] [40] stored in linear array in column major order. If the base address starts at 10. The location of A[20] [20] [30] is _____. [Assume the first element is stored at A[1][1][1] and each element take 1 B].

 23699

Correct Option

Solution:
23699

Let $A[r_1 \ i \ j \ k]$
 $\begin{array}{ccc} 90 & 30 & 40 \\ \text{(planes)} & \text{(rows)} & \text{(columns)} \end{array}$

For column major order

$$\begin{aligned} \text{loc}(A(i, j, k)) &= \text{Base Address} + [(i-1)r_2r_3 + (K-1)r_2 + (j-1)] \times 1 \text{ B} \\ &= 10 + [19 \times (30)(40) + 29 \times (30) + 19] \times 1 \text{ B} \\ &= 23699 \end{aligned}$$

Q. 17

Have any Doubt ?



Consider the following code:

```
int Do (char * gate)
{
    char * gate1 = gate;
    char * gate2 = gate + strlen (gate) - 1;
    while (gate1 < gate2)
    {
        if (* gate1 ++ != * gate2 --)
            return 0;
    }
    return 1;
}
```

What is the functionality of above function Do ()?

- A Check whether string is odd palindrome
- B Check whether the string is even palindrome
- C Check whether the string is palindrome

Correct Option

Solution:

(c)
Here two pointers are used gate1 and gate2. gate1 pointer points to beginning and gate2 points to the end. Loop is set up that compares the characters pointed by these two pointers. If the characters do not match, then it's returning 0 i.e. it is not a palindrome. It returns 1 for both even and odd palindrome (i.e., when match occurs for entire loop).

- D None of the above

 QUESTION ANALYTICS



Q. 18

Have any Doubt ?



Consider the following C program segment:

```
struct node
{
    int value;
    struct node * left;
    struct node * right;
};

void find (node * root, int K)
{
    if (root == NULL || K < 0)
        return 0;
    if (K == 0)
        printf ("%d", root -> value);
    else
        {
            find (root -> left, K - 1);
            find (root -> right, K - 1);
        }
}
```

What is the output of above C program when it runs on a root node of a binary tree?

- A Print nodes at K distance from root node.

Correct Option

Solution :

(a)
The above program segment print the node value when it is at distance 'K' from root node otherwise it return '0'. If length of binary tree from root node is less than 'K' then program return '0'. Only when height is greater than 'K' the program prints the node value of distance 'K' from root node.

B Print nodes of Kth level of binary tree.

Your answer is Wrong

C Both (a) and (b)**D** None of these

QUESTION ANALYTICS

+

Q. 19

Have any Doubt ?

Consider the following C program execute on a singly linked list numbered from 1 to n containing atleast 2 nodes:

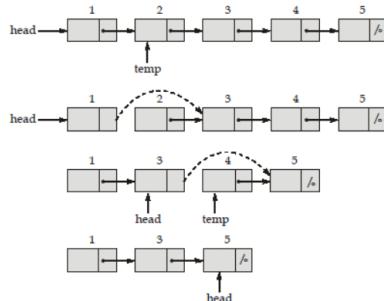
```
struct Listnode
{
    int data;
    struct Listnode *next;
};

void fun (struct Listnode *head)
{
    if (head == NULL || head -> next ==NULL) return;
    struct Listnode *tmp = head -> next;
    head -> next = tmp -> next;
    free (tmp);
    fun (head -> next);
}
```

Which of the following represents the output of above function 'fun'?

A It reverses the every 2 adjacent nodes linked list**B** Every odd number nodes of given linked list will be deleted**C** Every even number nodes of given linked list will be deleted

Your answer is Correct

Solution :
(c)

The above program deletes every even number node in the linked list (In particular second, fourth, sixth... soon nodes will be deleted)

D It reverses the linked list and delete alternate nodes

QUESTION ANALYTICS

+

Q. 20

Have any Doubt ?

Consider the following function:

```
shift (Q)
{
    while (! Q. isEmpty ( ))
        S. push (Q. dequeue ( ));
    while (! S. isEmpty ( ))
        Q.enqueue (S. pop ( ));
}
```

Assume S is stack which is initially empty and Q is queue which contain n-elements. The functions:

I. isEmpty() returns true if stack/queue is empty otherwise returns false.

II. Push() and Pop() functions are standard stack operations.

III. enqueue() and dequeue() are queue operations.

What will be the contents of Q after execution of shift(Q)? (Assume S and Q are global variables)

A No change to the contents of Q

Your answer is Wrong

B Q will be empty (No element in the queue)**C** Q will be reversed (All elements are in the reverse order)

Correct Option

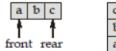
Solution :

Solution.

(c)

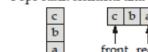
while (!Q.isEmpty ())
S.Push(Q.dequeue ());

Pushes all Q elements onto S by deleting from Q



while (!S.isEmpty ())
Q.enqueue (S.Pop ());

Pops stack elements and inserts into Q



∴ Content of queue(Q) will be reversed after the execution of shift(Q).

D None of these

QUESTION ANALYTICS



Item 11-20 of 33 « previous 1 2 3 4 next »



Kunal Jha
 Course: GATE
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

SINGLE SUBJECT : PROGRAMMING AND DATA STRUCTURES (GATE - 2020) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(33\)](#)
[CORRECT\(14\)](#)
[INCORRECT\(9\)](#)
[SKIPPED\(10\)](#)
Q. 21
[Have any Doubt ?](#)


Consider a hash table of size 11. Let $h(k) = k \bmod 11$ be the hash function used. A sequence of records with keys {33, 68, 62, 47, 48, 60, 105, 23, 19, 120, 97} is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. Which of the following is true ?

- A If, collisions are resolved by linear probing then maximum number of comparisons in worst case can be 2.
- B If chaining is used to resolve collision, the average chain length will be 1.09.
- C Both of these

- D None of these

Your answer is Correct

Solution :

(d)

Inserting keys in the hash table, we get:

33	0
23	1
68	2
47	3
48	4
60	5
105	6
62	7
19	8
97	9
120	10

Since, there is no collision in the hash table.

- Using linear probing, maximum comparison will be 1.
- Using chaining, the average chain length will be 1.

[QUESTION ANALYTICS](#)

Q. 22
[Have any Doubt ?](#)


Consider the linked list of integers represented by the following diagram.



Head

Run the following code with the above list of integers.

```

Node * Prev, * nodeToDelete;
Prev = Head → next;
nodeToDelete = (struct node *) malloc(sizeof (struct node));
nodeToDelete → item = 28;
nodeToDelete → next = Prev → next;
Prev → next = nodeToDelete;
Assume that the following structure is used to create a node in the list. struct Node
{
    int item;
    Node * next;
};
    
```

Which of the following is the effect of code?

- A Element 28 becomes the first element of the list.
- B Element 28 becomes the second element of the list.
- C One element is deleted from the list.

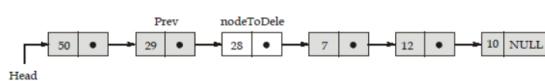
- D Element 28 becomes the 3rd element of the list.

Your answer is Correct

Solution :

(d)

The elements 28 is inserted between 29 and 7. The new list after the code is executed is shown below.


[QUESTION ANALYTICS](#)


Q. 23

Have any Doubt ?

Which of the following recurrence relation represents the number of nodes $N(h)$ in a minimal AVL tree of height ' h '?

A $N(h) = N(h - 1) + N(h - 2)$

B $N(h) = N(h + 1) + N(h + 2) + 1$

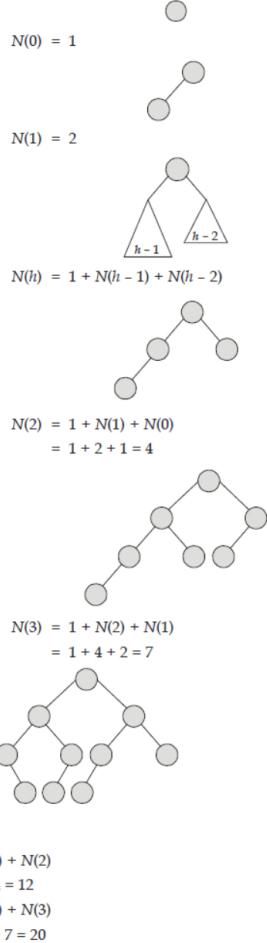
C $N(h) = 1 + N(h - 1) + N(h - 2)$

Your answer is Correct

Solution :

(c)

Let $N(h)$ be the number of nodes in a minimal AVL tree with height h .



D $N(h) = 2 * N(h - 1) * N(h - 2)$

QUESTION ANALYTICS

Q. 24

Have any Doubt ?

Consider the following code:

```
int f(Struct Node *P, Struct Node *Q)
{
    if (P == NULL && Q == NULL) return 0;
    else if (P == NULL || Q == NULL) return 1;
    else if (P → data != Q → data) return 1;
    else if (f(P → left, Q → left) || f(P → right, Q → right)) return 1;
    else return 0;
}
```

Assume Node is a structure type with 3 members as follows:

```
Struct Node
{
    int data;
    Struct Node *left;
    Struct Node *right;
};
```

If two binary tree root pointers are passed to the function $f()$, then which of the following statement is correct?

A It compares two given binary trees and returns 1 if two trees are different and it returns 0 otherwise.

Correct Option

Solution :

(a)

In given code 2nd else if statement return 1 when both tree values are different.
It returns 1, when both trees are different
It returns 0, when both trees are same recursively.

B It compares two given binary trees and returns 0 if two trees are different and it returns 0 otherwise.

C It compares two given binary trees but return value can not be used to differentiate the trees.

D None of the above

QUESTION ANALYTICS



Q. 25

Solution Video

Have any Doubt ?



Which of the following is true about linked list implementation of queue?

A In enqueue operation, if new node is inserted at the beginning of linked list, then in dequeue operation node must be removed from end.

B In enqueue operation, if new node is inserted at the end, then in dequeue operation node must be removed from the beginning.

C Both (a) and (b)

Your answer is Correct

Solution :

(c)

To keep the first in first out order a queue can be implemented using linked list in 2 ways.

1. If a new node is inserted at the beginning of linked list, then in dequeue operation node must be removed from end.
2. If new node is inserted at the end of linked list then, in dequeue operation, node must be removed from beginning.

D None of these

QUESTION ANALYTICS



Q. 26

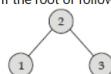
Have any Doubt ?



Consider the following program:

```
void find (struct Node *node)
{
    struct Node *ptr, *q;
    if (node == NULL) return;
    find (node -> left);
    find (node -> right);
    ptr = node -> left;
    q -> data = node -> data;
    node -> left = q;
    node -> left -> left = ptr;
}
```

If the root of following tree is passed to the above function, the sum of all keys in the resultant tree produced by find () is given by _____.



A 7

B 8

C 12

Correct Option

Solution :

(c)

Given function duplicates each node

$$\text{Sum} = (2 + 1 + 3) \times 2 = 12$$

D 18

QUESTION ANALYTICS



Q. 27

Have any Doubt ?



The following C function takes single linked list of integers as parameter and rearrange the elements of the list. The function is called with the list containing integers 10, 20, 30, 40, 50, 60, 70 in given order and generate output as 20, 10, 40, 30, 60, 50, 70 after completion. What will be the correct options files so we get desired output?

```
struct node
{
    int val;
    struct node * next;
}
void Altswap(struct node * list)
{
    struct node *p, *q;
    int temp;
    if (! list || list -> next) return;
    p = list; q = list -> next;
    while (q)
        ,
```

```

    W
    X
    Y
    Z
    A
}

(a) W : p → val = q - val;
    X : q → val = temp;
    Y : p = q → next;
    Z : temp = p → val;
    A : q = p? p → next: 0;

(b) W : temp = p → val;
    X : q → val = temp;
    Y : p → val = q → val;
    Z : q = p? p → next: 0;
    A : p = q → next;

(c) W : q = p? p → next: 0;
    X : temp = p → val;
    Y : p → val = q → val;
    Z : q → val = temp;
    A : p = q → next;

(d) W : temp = p → val;
    X : p → val = q → val;
    Y : q → val = temp;
    Z : p = q → next;
    A : q = p? p → next: 0;

```

A a

B b

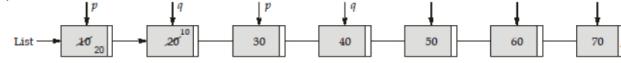
C c

D d

Correct Option

Solution :

(d)



1. temp = 10;
 2. p → value = q → value;
 3. q → value = temp;
 4. p = q → next;
 5. q = p ? p → next: 0;
- So option (d) is correct.

QUESTION ANALYTICS



Q. 28

Have any Doubt ?



Consider the following implementation of SELECTION SORT to sort the array A in a nondecreasing order. The algorithm first finds the smallest element and places in the first location. Then repeatedly selects the smallest element from the remaining elements and places in the first remaining location. The process is repeated $(n - 1)$ times.

```

begin
  for i = 1 to n - 1 do
    min = i
    for j = i + 1 to n do
      if P then min = j
    end for
    Swap (Q)
  end for
end

```

Identify the missing expressions at P and Q respectively.

A (A[j] > A[min]) and (A[i], A[min])

B (A[j] < A[min]) and (A[j], A[min])

C (A[i] < A[min]) and (A[i], A[min])

D (A[j] < A[min]) and (A[i], A[min])

Correct Option

Solution :

(d)

Whenever the if condition P : A[j] < A[min] is true then minimum index value will be updated.
After the end of 1st iteration 1st minimum will go to its correct place. This is performed by swapping (A[i], A[min]).

QUESTION ANALYTICS



Q. 29

Have any Doubt ?



Consider the following program:

```

int find (int n)
{
  int a = 1;
  for (i = 1; i <= n; i++)
    for (j = 1; j <= i; j++)
      for (k = 1; k <= j; k++)
        a = a + 1;
  return a;
}

```

The value returned by find (9) is _____.

166

Correct Option

Solution :
166

$$\begin{aligned}\text{find } (n) &= 1 \text{ (} \because 1 \text{ is initial value of a)} + \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^j 1 \\ &= 1 + \sum_{i=1}^n \sum_{j=1}^i j \quad \left[\text{As } \sum_{k=1}^j (1) = j \right] \\ &= 1 + \sum_{i=1}^n \frac{i(i+1)}{2} \quad \left[\text{As } \sum_{i=1}^n (j) = \frac{n(n+1)}{2} \right] \\ &= 1 + \frac{1}{2} \sum_{i=1}^n (i^2 + i) \\ \text{find } (n) &= 1 + \frac{1}{2} \left[\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right] \\ \text{find } (10) &= 1 + 165 = 166\end{aligned}$$

QUESTION ANALYTICS



Q. 30

Have any Doubt ?



Given finite alphabet set = {A, B, C} and stack S of size 100. There are only three stack operations we can perform as mentioned below.

- push (X): push the letter X from the alphabet on to the stack
- emit (): output the top letter from the stack.
- pop (): pop the top letter from the stack.

Stack is initially empty and we do not perform pop () on empty stack. Assume that only emit () can print output and stack may or may not be empty finally. The minimum number of stack operations to get "A B C A C B A" as output are _____.

14

Correct Option

Solution :
14

The minimum set of operation are

Push A
emit
Push B
emit
Push C
emit
Push A
emit
POP
emit
POP
emit
POP
emit

14 operations

After the above operations final content of stack is



Your Answer is 18

QUESTION ANALYTICS





Kunal Jha

 Course: GATE
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

SINGLE SUBJECT : PROGRAMMING AND DATA STRUCTURES (GATE - 2020) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(33\)](#)
[CORRECT\(14\)](#)
[INCORRECT\(9\)](#)
[SKIPPED\(10\)](#)
Q. 31
[Have any Doubt ?](#)


Consider the following program:

```
void find (struct Node *node)
{
    struct Node *ptr, *q;
    if (node == NULL) return;
    find (node->left);
    find (node->right);
    ptr = node->left;
    q = (struct node *) malloc(sizeof (struct node));
    q->data = node->data;
    node->left = q;
    node->left->left = ptr;
}
```

If the root of following tree is passed to the above function, the sum of all keys in the resultant tree produced by find () is given by _____.


12
[Correct Option](#)
Solution :

12

Given function duplicates each node

$$\text{Sum} = (2 + 1 + 3) \times 2 = 12$$

[Your Answer is 10](#)
Q. 32
[Have any Doubt ?](#)

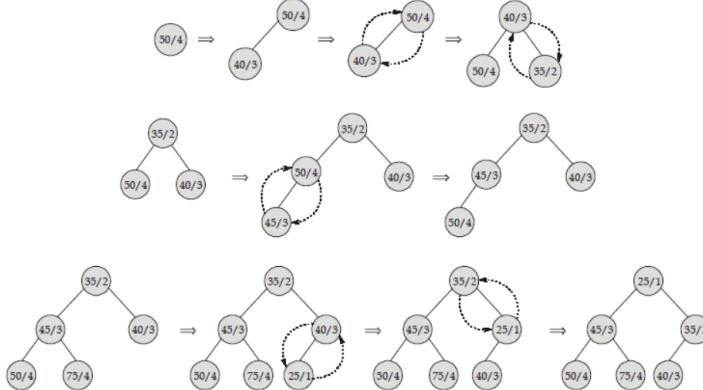

A priority queue is implemented with a heap in such a way that the parent element always have higher priority than its children. Following keys are to be inserted [(50, 4) (40, 3) (35, 2) (45, 3) (75, 4) (25, 1)]

Note: Key [k, l] depicts 'l' is the priority of 'k' key. Assume 1 to be the highest priority and 4 to the lowest priority.

The number of swaps required are _____.

5
[Correct Option](#)
Solution :

5


 Here 50/4 represents key 50 with priority 4.

QUESTION ANALYTICS

Q. 33
[Have any Doubt ?](#)


Consider the following C program:

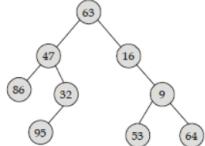
```
struct treenode
{
    int data;
    struct treenode *left;
    struct treenode *right;
```

```

} ;
void Even (struct treenode *root)
{
    If (root ==NULL) return;
    Even (root → right);
    Even (root → left);
    if (root → data %2 = 0)
        (root → data) / = 2;
    else
    {
        int Sum = 0;
        if (root → left != NULL) Sum += root → left → data;
        if (root → right != NULL) Sum += root → right → data;
        root → data += Sum;
    }
    printf ("%d", root → data);
}

```

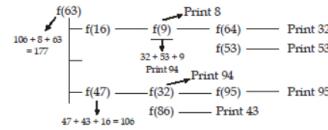
If root of the below tree is passed as a parameter to the above function then the sum of even output values produced by the above code is _____.



● 256

Correct Option

Solution :
256



32, 53, 94, 8, 95, 16, 43, 106, 177

Sum of even values: $32 + 94 + 8 + 16 + 106 = 256$

Your Answer is 361

● QUESTION ANALYTICS

+

Item 31-33 of 33 « previous 1 2 3 4 next »