1. 

If $g(x)=1-x$ and $h(x)=\frac{x}{x-1}$, then $\frac{g(h(x))}{h(g(x))}$ is:
A. $\frac{h(x)}{g(x)}$
-1
B. $x$
C. $\frac{g(x)}{h(x)}$
D. $\frac{x}{(1-x)^{2}}$

Answer ||| A
Solution |||
$g(h(x))=g(x /(x-1))=1-x /(x-1)=-1 /(x-1)$
$h(g(x))=h(1-x)=(1-x) /((1-x)-1)=-(1-x) / x$
$g(h(x)) / h(g(x))=[-1 /(x-1)] /[-(1-x) / x]=-x /(x-1)^{2}-$
$x /(x-1)^{2}$ is same as $h(x) / g(x)$
2. $\lim _{x \rightarrow \infty} x^{1 / x}$ is
A. $\infty$
B. 0
C. 1
D. Not defined

Answer ||| C
3. Match the following:

List - I
(P) Prim's algorithm for minimum spanning tree
(Q) Floyd-Warshall algorithm for all pairs shortest paths
(R) Mergesort
(S) Hamiltonian circuit

List - II
(i) Backtracking
(ii) Greedy method
(iii) Dynamic programming
(iv) Divide and conquer
A. P-iii, Q-ii, R-iv, S-i
B. P-i, Q-ii, R-iv, S-iii
C. P-ii, Q-iii, R-iv, S-i
D. P-ii, Q-i, R-iii, S-iv

Answer ||| C
Solution |||
Prim's Algorithm is a greedy algorithm where greedy choice is to pick minimum weight edge from cut that divides already picked vertices and vertices yet to be picked. Floyd-Warshall algorithm for all pair's shortest paths is a Dynamic Programming algorithm where we keep updating the distance matrix in bottom up manner. Merge Sort is clearly divide and conquer which follows all steps of divide and conquer. It first divides the array in two halves, then conquers the two halves and finally combines the conquered results. Hamiltonian circuit is a NP complete problem that can be solved using Backtracking
4. Which one of the following is the recurrence equation for the worst case time complexity of the Quicksort algorithm for sorting $n(\geq 2)$ numbers? In the recurrence equations given in the options below, cis a constant.
A. $T(n)=2 T(n / 2)+c n$
B. $T(n)=T(n-1)+T(1)+c n$
C. $T(n)=2 T(n-1)+c n$
D. $T(n)=T(n / 2)+c n$

Answer ||| B
Solution |||
In worst case, the chosen pivot is always placed at a corner position and recursive call is made for following.
a) for sub array on left of pivot which is of size $n-1$ in worst case.
b) for sub array on right of pivot which is of size 0 in worst case.
5. The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are
A. 63 and 6 , respectively
B. 64 and 5, respectively
C. 32 and 6, respectively
D. 31 and 5, respectively

Answer ||| A
Solution |||
Number of nodes is maximum for a perfect binary tree. A perfect binary tree of height $h$ has $2^{h+1}-1$ nodes Number of nodes is minimum for a skewed binary tree. A perfect binary tree of height $h$ has $h+1$ nodes.
6. Match the following:

List - I
(P) Condition coverage
(Q) Equivalence class partitioning
(R) Volume testing
(S) Alpha testing

List - II
(i) Black-box testing
(ii) System testing
(iii) White-box testing
(iv) Performance testing
A. P-ii, Q-iii, R-i, S-iv
B. P-iii, Q-iv, R-ii, S-i
C. P-iii, Q-i, R-iv, S-ii
D. P-iii, Q-i, R-ii, S-iv

Answer ||| C
Solution |||
White Box Testing tests internal structures or workings of an application. It covers following Control flow testing Data flow testing Branch testing Statement coverage Decision coverage Modified condition/decision coverage Prime path testing Path testing So conditional coverage must be in White-Box Testing. Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. Typical black-box test design techniques include: Decision table testing All-pairs testing Equivalence partitioning Boundary value analysis Cause-effect graph Error guessing Volume Testing Does performance testing for specific size. Alpha testing is system testing by potential users/customers or an independent test team at the developers' site.
7. Which of the following is/are correct in order traversal sequence(s) of binary search tree(s)?
I. $3,5,7,8,15,19,25$
II. $5,8,9,12,10,15,25$
III. $2,7,10,8,14,16,20$
IV. $4,6,7,918,20,25$
A. I and IV only
B. II and III only
C. II and IV only
D. II only

Answer ||| A
Solution |||
An In order traversal of a Binary Search Tree must be in increasing order.
In-order Traverse the left sub tree by recursively calling the in-order function. Display the data part of the root (or current node). Traverse the right sub tree by recursively calling the in-order function. In a search tree, in-order traversal retrieves data in sorted order
8. Which one of the following is TRUE at any valid state in shift-reduce parsing?
A. Viable prefixes appear only at the bottom of the stack and not inside
B. Viable prefixes appear only at the top of the stack and not inside
C. The stack contains only a set of viable prefixes
D. The stack never contains viable prefixes

Answer ||| C
Solution |||
The prefixes of right sentential forms that can appear on the stack of a shift-reduce parser are called viable prefixes. By definition, a viable prefix is a prefix of a right sentential form that does not continue past the right end of the rightmost handle of that sentential form.
9. Which one of the following is NOT equivalent to $p \leftrightarrow q$ ?
A. $\left({ }^{\prime} p \vee q\right) \wedge(p \vee \neg q)$
B. $(\neg p \vee q) \wedge(q \rightarrow p)$
C. $(\neg p \wedge q) \vee(p \wedge \neg q)$
D. $\left({ }_{\neg} p \wedge \neg q\right) \vee(p \wedge q)$

Answer ||| C
Solution ||l
$\mathrm{p} \leftarrow \rightarrow \mathrm{q}$ means both $\mathrm{p} \rightarrow \mathrm{q}$ and $\mathrm{q} \rightarrow \mathrm{p} \mathrm{p} \rightarrow \mathrm{q}$ is equivalent to p $\vee q$ So $A$ and $B$ are fine.
$D$ is a different way of writing $A p \leftrightarrow q$
$(p \rightarrow q) \wedge(q \rightarrow p)(l p \vee q) \wedge(q \rightarrow p)($ As $p \rightarrow q=l p \vee q)(p$
$\vee q) \wedge(l q \vee p)(l p \wedge \mid q) \vee(p \wedge q)$ So, answer C
10. For a set $A$, the power set of $A$ is denoted by $2^{A}$. If $A=\{5$, $\{6\},\{7\}\}$, which of the following options are TRUE?
I. $\varnothing \in 2^{A}$
II. $\varnothing \subseteq 2^{A}$
III. $\{5,\{6\}\} \in 2^{A}$
IV. $\{5,\{6\}\} \subseteq 2^{A}$
A. I and III only
B. II and III only
C. I, II and III only
D. I, II and IV only

Answer ||| C
Solution |||
$A=\{5,\{6\},\{7\}\}$
Below is Power set of $A$ ?
$\{\varphi$,
5,
\{6\},
\{7\},
$\{5,\{6\}\}$,
$\{5,\{7\}\}$,
$\{\{6\},\{7\}\}$,
$\{5,\{6\},\{7\}\}$
\}
I am true, as $\varphi$ belongs to Power set. II is true, as an empty set is a subset of every set. III is true as $\{5,\{6\}\}$ belongs to Power set. IV is false, $\{5,\{6\}\}$ is not a subset, but $\{\{5,\{6\}\}\}$ is.
11. Consider a 4-bit Johnson counter with an initial value of 0000 . The counting sequence of this counter is
A. $0,1,3,7,15,14,12,8,0$
B. $0,1,3,5,7,9,11,13,15,0$
C. $0,2,4,6,8,10,12,14,0$
D. $0,8,12,14,15,7,3,1,0$

Answer ||| D
Solution |||
The four bit Johnson's counter connects the complement of the output of the last shift register to the input of the first register with shift distance=1 i.e. 1 bit will shift/cycle It will work as follows: 0000 //Last 0 complemented and fed as input to first register 1000110011101111 //Last 1 complemented and fed as input to first register 0111 001100010000
12. For computers based on three-address instruction formats, each address field can be used to specify which of the following:
(S1) A memory operand
(S2) A processor register
(S3) An implied accumulator register
A. Either S1 or S2
B. Either S2 or S3
C. Only S2 and S3
D. All of S1, S2 and S3

Answer ||| A
Solution |||
In Three address instruction format, each operand specifies either a memory address or a register.
13. Suppose two hosts use a TCP connection to transfer a large file. Which of the following statements is/are FALSE with respect to the TCP connection?
I. If the sequence number of a segment is $m$, then the sequence number of the subsequent segment is always $m+1$.
II. If the estimated round trip time at any given point of time is $t$ sec, the value of the retransmission timeout is always set to greater than or equal to $t$ sec. III. The size of the advertised window never changes during the course of the TCP connection.
IV. The number of unacknowledged bytes at the sender is always less than or equal to the advertised window.
A. III only
B. I and III only
C. I and IV only
D. II and IV only

Answer ||| B
Solution |||
TCP sequence number of a segment is the byte number of the first byte in the segment. For example, if the segment contains 500 bytes which are from 1000 to

1499, then sequence number of the segment would be 1000 and sequence number of next segment would be 1500. Receiver window changes when TCP data is processed by application layer of receiver side.
14. Suppose that everyone in a group of $N$ people wants to communicate secretly with the $N-1$ others using symmetric key cryptographic system. The communication between any two persons should not be decodable by the others in the group. The number of keys required in the system as a whole to satisfy the confidentiality requirement is
A. 2 N
B. $N(N-1)$
C. $N(N-1) / 2$
D. $(N-1)^{2}$

Answer ||| C
Solution |||
In Symmetric Key Cryptography, access of key is with both the parties. It implies every person needs to communicate $\mathrm{N}-1$ other users using different keys i.e. $1+2+3 \ldots \mathrm{~N}-2+\mathrm{N}-1$ This is like number of edges needed in a complete graph with N vertices is $\mathrm{N}(\mathrm{N}-1) / 2$. Answer is therefore $\mathbf{C}$
15. Which of the following statements is/are FALSE? I. XML overcomes the limitations in HTML to support a structured way of organizing content.
II. XML specification is not case sensitive while HTML specification is case sensitive.
III. XML supports user defined tags while HTML uses predefined tags.
IV. XML tags need not be closed while HTML tags must be closed.
A. II only
B. I only
C. II and IV only
D. III and IV only

Answer ||| C
Solution |||

1) TRUE- XML is a structured way of organizing content.
2) FALSE- XML is CASE SENSITIVE whereas

HTML is NOT case sensitive.
3) TRUE- XML facilitates User Defined tags whereas HTML has only Pre-Defined tags
4) FALSE- XML tags MUST be closed while HTML tags may NOT be closed.

16 Which one of the following fields of an IP header is
NOT modified by a typical IP router?
A. Checksum
B. Source address
C. Time to Live (TTL)
D. Length

Answer ||| B
Solution |||
Length and checksum can be modified when IP fragmentation happens. Time To Live is reduced by every router on the route to destination. Only Source Address is what IP address cannot change SO $B$ is the answer.
17. In one of the pairs of protocols given below, both the protocols can use multiple TCP connections between the same client and the server. Which one is that?
A. HTTP, FTP
B. HTTP, TELNET
C. FTP, SMTP
D. HTTP, SMTP

Answer ||| A
Solution |||
HTTP may use different TCP connection for different objects of a webpage if non-persistent connections are used. FTP uses two TCP connections, one for data and another control. TELNET and FTP can only use ONE connection at a time.
18. For any two languages $L_{1}$ and $L_{2}$ such that $L_{1}$ is context-free and $L_{2}$ is recursively enumerable but not recursive, which of the following is/are necessarily true?
I. $\bar{L}_{1}\left(\right.$ complement of $\left.L_{1}\right)$ is recursive
II. $\bar{L}_{2}$ (complement of $L_{2}$ ) is recursive
III. $\bar{L}_{1}$ is context-free
IV. $\bar{L}_{1} \cup \mathrm{~L}_{2}$ is recursively enumerable
A. I only
B. III only
C. III and IV only
D. I and IV only

Answer ||| D
Solution |||

1) L1' (complement of L1) is recursive is true L 1 is context free. Every context free language is also recursive and recursive languages are closed under complement.
2) L1' $\cup \mathbf{L 2}$ is recursively enumerable is true Since L1' is recursive, it is also recursively enumerable and recursively enumerable languages are closed under union. Recursively enumerable languages are known as type-0 languages in the Chomsky hierarchy of formal languages. All regular, context-free, context-sensitive and recursive languages are recursively enumerable. (Source: Wiki)
3) L1' is context-free: Context-free languages are not closed under complement, intersection, or difference.
4) $\mathbf{L 2}$ ' (complement of $\mathbf{L 2}$ ) is recursive is false: Recursively enumerable languages are not closed under set difference or complementation
19. Consider a system with byte-addressable memory, 32-bit logical addresses, 4 kilobyte page size and page table entries of 4 bytes each. The size of the page table in the system in megabytes is $\qquad$ -.
A. 2
B. 4
C. 8
D. 16

Answer ||| B
Solution |||
Number of entries in page table $=2^{32} / 4$ Kbyte
$=2^{32} / 212$
$=2^{20}$
Size of page table $=(\text { No. page table entries })^{*}($ Size of an entry)
$=2^{20} * 4$ bytes
$=2^{22}$

## $=4$ Megabytes

20. The following two functions P1 and P2 that share a variable $B$ with an initial value of 2 execute concurrently.
```
P1( ) {
                                    P2( ) {
    C = B - 1; D = 2 * B;
    B = 2 * C;
    B = D - 1;
}
}
```

The number of distinct values that $B$ can possibly take after the execution is $\qquad$ .
A. 3
B. 2
C. 5
D. 4

Answer ||| A
Solution |||
There are following ways that concurrent processes can follow.
$C=B-1 ; / / C=1$
$B=2 * C ; / / B=2$
$D=2$ * $B ; / / D=4$
$B=D-1 ; / / B=3$
$C=B-1 ; / / C=1$
$D=2 * B ; / / D=4$
$B=D-1 ; / / B=3$
$B=2 * C ; / / B=2$
$C=B-1 ; / / C=1$
$D=2 * B ; / / D=4$
$B=2 * C ; / / B=2$
$B=D-1 ; / / B=3$
$D=2 * B ; / / D=4$
$C=B-1 ; / / C=1$
$B=2 * C ; / / B=2$
$B=D-1 ; / / B=3$
$D=2 * B ; / / D=4$
$B=D-1 ; / / B=3$
$C=B-1 ; / / C=2$
$B=2 * C ; / / B=4$
There are 3 different possible values of $B$ : 2, 3 and 4
21. SELECT operation in SQL is equivalent to
A. the selection operation in relational algebra
B. the selection operation in relational algebra, except
that SELECT in SQL retains duplicates
C. the projection operation in relational algebra
D. the projection operation in relational algebra, except
that SELECT in SQL retains duplicates
Answer ||| D
Solution |||
Select operation is equivalent to the projection operation in relational algebra, except that select in SQL retains duplicates and on the contrary projection removes the duplicates.
22. A file is organized so that the ordering of data records is the same as or close to the ordering of data entries in some index. Then that index is called
A. Dense
B. Sparse
C. Clustered
D. Unclustered

Answer ||| C
Solution |||
In Clustered Index, data blocks are stored in a way to match the index. Therefore, only one clustered index can be created on a given database table.
23. In the LU decomposition of the matrix

$$
\left[\begin{array}{ll}
2 & 2 \\
4 & 9
\end{array}\right]
$$

if the
diagonal elements of $U$ are both 1 , then the lower
diagonal entry $I_{22}$ of $L$ is $\qquad$ —.
A. 4
B. 5
C. 6
D. 7

Answer ||| B
Solution |||
LU decomposition (where 'LU' stands for 'lower upper', and also called LU factorization) factors a matrix as the product of a lower triangular matrix and an upper triangular matrix.

```
| \(22\left|=\left|I_{11} 0\right| *\right| 1 u_{12} \mid\)
| 49 || \(I_{21} I_{22}| | 01\) |
\(I_{21} * u_{12}+I_{22} * 1=9\)------ (1)
We need to find \(\mathrm{I}_{21}\) and \(\mathrm{u}_{12}\)
\(I_{21} * 1+I_{22} * 0=4\)
\(l_{21}=4\)
\(\mathrm{I}_{11} * \mathrm{U}_{12}+0 * 1=2\)
\(l_{11}=2\)
\(\mathbf{U}_{12}=1\)
Putting value of \(\mathrm{I}_{21}\) and \(\mathrm{u}_{12}\) in (1), we get
\(4 * 1+I_{22} * 1=9\)
\(\mathrm{I}_{22}=5\)
```

24. The output of the following C program is $\qquad$ -
void f1(int a, int b) \{
int c;
$\mathrm{c}=\mathrm{a} ; \mathrm{a}=\mathrm{b} ; \mathrm{b}=\mathrm{c}$;
\}
void f2(int *a, int *b) \{
int c;
$\mathrm{c}=\star \mathrm{a}$; *a=*b; *b=c;
\}
int main() \{
int $a=4, b=5, c=6$;
f1 (a,b) ;
f2 (\&b, \&c) ;
printf("\%d", c-a-b) ;
\}
A. -5
B. -4
C. 3
D. 8

Answer ||| A
Solution |||
The function call to $f 1(a, b)$ won't have any effect as the values are passed by value. The function call f2(\&b, \&c) swaps values of $b$ and $c$. So b becomes 6 and $c$ becomes 5. Value of $\mathrm{c}-\mathrm{a}-\mathrm{b}$ becomes $5-4-6$ which is -5
25. What are the worst-case complexities of insertion and deletion of a key in a binary search tree?
A. $\theta(\log n)$ for both insertion and deletion
B. $\theta(n)$ for both insertion and deletion
C. $\theta(n)$ for insertion and $\theta(\log n)$ for deletion
D. $\theta(\log n)$ for insertion and $\theta(n)$ for deletion

Answer ||| B
Solution |||
The time taken by search, insert and delete on a BST is always proportional to height of BST. Height may become $\mathrm{O}(\mathrm{n})$ in worst case.
26. Suppose that the stop-and-wait protocol is used on a link with a bit rate of 64 kilobits per second and 20 milliseconds propagation delay. Assume that the transmission time for the acknowledgement and the processing time at nodes are negligible. Then the minimum frame size in bytes to achieve a link utilization of at least $50 \%$ is $\qquad$ _.
A. 160
B. 320
C. 640
D. 220

Answer ||| A
Solution |||
Transmission or Link speed $=64 \mathrm{~kb}$ per sec
Propagation Delay $=20$ milisec
Since stop and wait is used, a packet is sent only when previous one is acknowledged.
Let $x$ be size of packet, transmission time $=x / 64$ milisec
Since utilization is at least $50 \%$, minimum possible total time for one packet is twice of transmission delay, which means
$\mathrm{x} / 64 * 2=\mathrm{x} / 32$
$x / 32>x / 64+2 * 20$
$x / 64>40$
$x>2560$ bits $=320$ bytes
27 Consider a max heap, represented by the array: 40, $30,20,10,15,16,17,8,4$

| Array Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | 40 | 30 | 2 | 10 | 15 | 16 | 17 | 8 | 4 |

Now consider that a value 35 is inserted into this heap.
After insertion, the new heap is
A. $40,30,20,10,15,16,17,8,4,35$
B. $40,35,20,10,30,16,17,8,4,15$
C. $40,30,20,10,35,16,17,8,4,15$
D. $40,35,20,10,15,16,17,8,4,30$

Answer ||| B
Solution |||
The array $40,30,20,10,15,16,17,8,4$ represents following heap


After insertion of 35 , we get

| 40 |  |
| :---: | :---: |
| / | \} |
| 30 | 20 |
| / | / |
| 1015 | 1617 |
| / \/ |  |
| 8435 |  |

After swapping 35 with 15 and swapping 35 again with 30, we get

40
$1 \quad 1$
$35 \quad 20$
/ \ /
$\begin{array}{llll}10 & 30 & 16 & 17\end{array}$
/ \/
8415
28. Consider the following $C$ program segment.

```
while(first <= last)
{
    if (array[middle] < search)
    first = middle + 1;
        else if (array[middle] == search)
            found = TRUE;
        else last = middle - 1;
        middle = (first + last)/2;
}
if (first > last) notPresent = TRUE;
```

The cyclomatic complexity of the program segment is
A. 3
B. 4
C. 5
D. 6

Answer ||| C
Solution |||
The cyclomatic complexity of a structured program[a] is defined with reference to the control flow graph of the program, a directed graph containing the basic blocks of the program, with an edge between two basic blocks if control may pass from the first to the second. The complexity $M$ is then defined as
$M=E-N+2 P$,
where
$\mathrm{E}=$ the number of edges of the graph.
$N=$ the number of nodes of the graph.
$\mathrm{P}=$ the number of connected components.
For a single program (or subroutine or method), $P$ is always equal to 1 . So, a simpler formula for a single subroutine is
$M=E-N+2$
For the given program, the control flow graph is:

$E=13, N=10$.
Therefore, $\mathrm{E}-\mathrm{N}+2=5$.
29. Consider a LAN with four nodes $S_{1}, S_{2}, S_{3}$ and $S_{4}$. Time is divided into fixed-size slots, and a node can begin its transmission only at the beginning of a slot. A collision is said to have occurred if more than one node transmit in the same slot. The probabilities of generation of a frame in a time slot by $S_{1}, S_{2}, S_{3}$ and $S_{4}$ are $0.1,0.2$, 0.3 and 0.4 , respectively. The probability of sending a frame in the first slot without any collision by any of these four stations is $\qquad$ _.
A. 0.462
B. 0.711
C. 0.456
D. 5.903

Answer ||| A
Solution |||
The probability of sending a frame in the first slot without any collision by any of these four stations is sum of following 4 probabilities
Probability that S1 sends a frame and no one else does + Probability thatS2 sends a frame and no one else does + Probability thatS3 sends a frame and no one else does + Probability thatS4 sends a frame and no one else does
$=0.1 *(1-0.2) *(1-0.3) *(1-0.4)+$
$(1-0.1) * 0.2 *(1-0.3) *(1-0.4)+$
$(1-0.1) *(1-0.2) * 0.3 *(1-0.4)+$
$(1-0.1) *(1-0.2) *(1-0.3) * 0.4=0.4404$
30. The binary operator $\neq$ is defined by the following truth table.

| $p$ | $q$ | $p \neq q$ |
| :---: | :---: | :---: |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Which one of the following is true about the binary operator $\neq$ ?
A. Both commutative and associative
B. Commutative but not associative
C. Not commutative but associative
D. Neither commutative nor associative

Answer ||| A
Solution |||
The operation is basically XOR which is both commutative and associative. XOR is true only when an odd number of inputs are true. A chain of XORs-a XOR b XOR c XOR d (and so on)-is true whenever an odd number of the inputs are true and is false whenever an even number of inputs are true.
31. $\sum_{x=1}^{99} \frac{1}{x(x+1)}=$ $\qquad$
A. 0.99
B. 1
C. 99
D. 99.9

Answer ||| A
Solution |||
$\mathrm{S}=1 / 1^{*} 2+1 / 2 * 3+1 / 3 * 4+\ldots . .+1 / 99 * 100$
$=(1-1 / 2)+(1 / 2-1 / 3)+(1 / 3-1 / 4)+\ldots+(1 / 99-$
1/100)
$=(1+1 / 2+1 / 3 \ldots .1 / 99)-(1 / 2+1 / 3+1 / 4 \ldots 1 / 100)$
$=1-1 / 100$
= 99/100
$=0.99$
32. Suppose $\mathcal{L}=\{p, q, r, s, t\}$ is a lattice represented by the following Hasse diagram:


For any $x, y \in \mathcal{L}$, not necessarily distinct, $x \vee y$ and $x \wedge$ $y$ are join and meet of $x, y$, respectively. Let $\mathcal{L}^{3}=\{(x$, $y, z): x, y, z \in \mathcal{L}\}$ be the set of all ordered triplets of the elements of $\mathcal{L}$. Let $p_{r}$ be the probability that an element
$(x, y, z) \in \mathcal{L}^{3}$ chosen equiprobably satisfies $x \vee(y \wedge z)=$ $(x \vee y) \wedge(x \vee z)$. Then
A. $p_{r}=0$
B. $p_{r}=1$
C. $0<p_{r} \leq \frac{1}{5}$
D. $\frac{1}{5}<p_{r}<1$

Answer ||| D
Solution |||
Number of triplets in $L^{3}=$ Number of ways in which we can choose 3 elements from 5 with repetition
$=5 * 5 * 5$
$=125$
Now, when we take $\mathbf{x}=\mathbf{t}$, then the given condition for L is satisfied for any $y$ and $z$. Here, $y$ and $z$ can be taken in $5 * 5=25$ ways.
Take $\mathbf{x}=\mathbf{r}, \mathrm{y}=\mathrm{p}, \mathrm{z}=\mathrm{p}$. Here also, the given condition is satisfied. So, pr > $25 / 125>1 / 5$
For $\mathbf{x}=\mathbf{q}, \mathrm{y}=\mathrm{r}, \mathrm{z}=\mathrm{s}$, the given condition is not satisfied as $q \vee(r \wedge s)=q \vee p=q$, while $(q \vee r) \wedge(q \vee s)$
$=\mathrm{t} \wedge \mathrm{t}=\mathrm{t}$.
So, $\mathrm{pr} \neq 1$
Hence D choice.
33. Consider the operations
$f(x, y, z)=x^{\prime} y z+x y^{\prime}+y^{\prime} z^{\prime}$ and $g(x, y, z) x^{\prime} y z+x^{\prime} y z^{\prime}+$ $x y$.
Which one of the following is correct?
A. Both $\{f\}$ and $\{g\}$ are functionally complete
B. Only $\{f\}$ is functionally complete
C. Only $\{g\}$ is functionally complete
D. Neither $\{f\}$ nor $\{g\}$ is functionally complete

Answer ||| B
Solution |||
A function is considered as functionally complete if it does not belong to T0, T1, L, M, S which are Property 1: We say that Boolean function $f$ preserves zero, if on the 0 input it produces 0 . By the 0 -input we mean such an input, where every input variable is 0 (this input usually corresponds to the first row of the truth table). We denote the class of zero-preserving Boolean functions as TO and write $f \in$ TO. Property 2: Similarly to TO, we say that Boolean function $f$ preserves one, if on 1 -input, it produces 1 . The 1 -input is the input where all the input variables are 1 (this input usually corresponds to the last
row of the truth table). We denote the class of onepreserving Boolean functions as T 1 and write $\mathrm{f} \in \mathrm{T} 1$. Property 3: We say that Boolean function $f$ is linear if one of the following two statements holds for $f$ :
For every 1 -value of $f$, the number of 1 ' $s$ in the corresponding input is odd, and for every 0 -value of $f$, the number of 1 's in the corresponding input is even. or
For every 1 -value of $f$, the number of 1 ' $s$ in the corresponding input is even, and for every 0 -value of f , the number of 1 's in the corresponding input is odd.
If one of these statements holds for $f$, we say that $f$ is linear1. We denote the class of linear Boolean functions with $L$ and write $f \in L$. Property 4: We say that Boolean function $f$ is monotone if for every input, switching any input variable from 0 to 1 can only result in the function's switching its value from 0 to 1 , and never from 1 to 0 . We denote the class of monotone Boolean functions with $M$ and write $f \in M$. Property 5: We say that Boolean function $f(x 1, \ldots, x n)$ is self-dual if $f(x 1, \ldots, x n)=$ $\neg f(\neg x 1, \ldots, \neg x n)$. The function on the right in the equality above (the one with negations) is called the dual of $f$. We will call the class of self-dual Boolean functions $S$ and write $f \in S$. As in our case we can see on giving all I/p to 0 ( g )produce 0 so it preserving 0 and can't be functionally complete. But $f$ is neither preserving 0 nor 1. $F$ is not linear(see defn. of linear above)
$F$ is not monotone(see defn. of monotone above)
$F$ is not self dual as $f(x, y, z)$ is not equal to $-f(-x,-y,-z)$
So $f$ is functionally complete. Hence answer is ( $B$ ) part
34. Let $G$ be a connected planar graph with 10 vertices. If the number of edges on each face is three, then the number of edges in $G$ is $\qquad$ _.
A. 24
B. 4
C. 32
D. 56

Answer ||| A
Solution |||
Euler's formula states that if a finite, connected, planar graph is drawn in the plane without any edge intersections, then
$v-e+f=2$.
$v->$ Number of vertices
e -> Number of edges
f -> Number of faces
As per the question
$\mathrm{v}=10$
And number of edges on each face is three
Therefore, $2 e=3 f$ [Note that every edge is shared by 2 faces]
Putting above values in $v-e+f=2$
$10-e+2 e / 3=2$
$e=3 * 10-6=24$
35 Let $a_{n}$ represent the number of bit strings of length $n$ containing two consecutive 1 s . What is the recurrence relation for $a_{n}$ ?
A. $a_{n-2}+a_{n-1}+2^{n-2}$
B. $a_{n-2}+2 a_{n-1}+2^{n-2}$
C. $2 a_{n-2}+a_{n-1}+2^{n-2}$
D. $2 a_{n-2}+2 a_{n-1}+2^{n-2}$

Answer ||| A

Solution |||
A string of length $n(n>=2)$ can be formed by following 4 prefixes

1) 11 followed by a string of length $n-2$
2) 00 followed by a string of length $n-2$
3) 01 followed by a string of length $n-2$
4) 10 followed by a string of length $n-2$

Number 1 has already two consecutive 1's so number of binary strings beginning with number 3 is $2^{n-2}$ as remaining $n-2$ bits can have any value.
Number 2 has two 0 's so remaining $\mathrm{n}-2$ bits must have two consecutive 1's. Therefore, number of binary strings that can be formed by number 2 is $a_{n-2}$.
Number 3 and Number 4 together form all strings of length n-1 and two consecutive 1's.
36. A variable $x$ is said to be live at a statement $S_{i}$ in a program if the following three conditions hold simultaneously:
i. There exists a statement $\mathrm{S}_{\mathrm{j}}$ that uses $x$
ii. There is a path from $S_{i}$ to $S_{j}$ in the flow graph corresponding to the program
iii. The path has no intervening assignment to $x$ including at $S_{i}$ and $S_{j}$


The variables which are live both at the statement in basic block 2 and at the statement in basic block 3 of the above control flow graph are
A. $p, s, u$
B. $r, s, u$
C. $r, u$
D. $q, v$

Answer ||| C
Solution |||
Live variable analysis is useful in compilers to find variables in each program that may be needed in future. As per the definition given in question, a variable is live if it holds a value that may be needed in the future. In other words, it is used in future before any new assignment.
37. The least number of temporary variables required to create a three-address code in static single assignment form for the expression $q+r / 3+s-t^{*} 5+u^{*} v / w$ is
A. 4
B. 8
C. 7
D. 9

Answer ||| B
Solution |||
The correct answer is 8 . Three address code is an intermediate code generated by compilers while optimizing the code. Each three address code instruction can have almost three operands (constants and variables) combined with an assignment and a binary operator. The point to be noted in three address code is
that the variables used on the left hand side (LHS) of the assignment cannot be repeated again in the LHS side. Static single assignment (SSA) is nothing but a refinement of the three address code.
So, in this question, we have
$\mathrm{t} 1=\mathrm{r} / 3$;
$\mathrm{t} 2=\mathrm{t} * 5$;
$\mathrm{t} 3=\mathrm{u}^{*} \mathrm{v}$;
$\mathrm{t} 4=\mathrm{t} 3 / \mathrm{w}$;
$\mathrm{t} 5=\mathrm{q}+\mathrm{t} 1$;
t6 $=$ t5 +s ;
$\mathrm{t} 7=\mathrm{t} 6-\mathrm{t} 2$;
$\mathrm{t} 8=\mathrm{t} 7+\mathrm{t} 4 ;$
Therefore, we require 8 temporary variables ( t 1 to t 8 ) to create the three address code in static single assignment form.
38. Consider an Entity-Relationship (ER) model in which entity sets $E_{1}$ and $E_{2}$ are connected by an $m: n$ relationship $R_{12} . E_{1}$ and $E_{3}$ are connected by a 1 : $n$ (1 on the side of $E_{1}$ and $n$ on the side of $E_{3}$ ) relationship $\mathrm{R}_{13}$.
$\mathrm{E}_{1}$ has two single-valued attributes $\mathrm{a}_{11}$ and $\mathrm{a}_{12}$ of which $a 11$ is the key attribute. $\mathrm{E}_{2}$ has two single- valued attributes $\mathrm{a}_{21}$ and $\mathrm{a}_{22}$ of which $\mathrm{a}_{21}$ is the key attribute. $E_{3}$ has two single-valued attributes $a_{31}$ and $a_{32}$ of which $a_{31}$ is the key attribute. The relationships do not have any attributes.
If a relational model is derived from the above ER model, then the minimum number of relations that would be generated if all the relations are in 3 NF is $\qquad$ _.
A. 2
B. 3
C. 4
D. 5

Answer ||| C
Solution |||
Entity E1
a1 a12
--------
a11 is key

## Entity E2

a21 a22
a22 is key

Entity E3
a31 a32
a31 is key
R12 is m:n Relationship between E1 and E2
R12
a11 a22
(a11, a22) is key.
R13 is 1:n Relationship between E1 and E3

R13
a11 a31
(a11, a31) is key.
We need minimum no. of tables.
Can we remove any of the above tables without loosing information and keeping the relations in 3NF?
We can combine R13 and R12 into one.
a11 a31 a22
(a11, a31, a22) is key.
The relation is still in $3 N F$ as for every functional dependency $X->A$, one of the following holds

1) $X$ is a super key or
2) A-X is prime attribute
39. 



Consider the DFAs M and N given above. The number of states in a minimal DFA that accepts the language $L(M) \cap$ $L(N)$ is
A. 0
B. 1
C. 2
D. 3

Answer ||| B
Solution |||
In DFA M: all strings must end with 'a'. In DFA $N$ : all strings must end with 'b'. So, the intersection is empty. For an empty language, only one state is required in DFA. The state is non-accepting and remains on itself for all characters of alphabet.

40. Consider the NPDA $\left\langle Q=\left\{\mathrm{q}_{0}, \mathrm{q}_{1}, \mathrm{q}_{2}\right\}, \Sigma=\{0,1\}, \Gamma=\right.$ $\left.\{0,1, \perp\}, \delta, q_{0}, \perp, F=\left\{q_{2}\right\}\right\rangle$, where (as per usual convention) Q is the set of states, $\Sigma$ is the input alphabet, $\Gamma$ is the stack alphabet, $\delta$ is the state transition function, $q_{0}$ is the initial state, $\perp$ is the initial stack symbol, and F is the set of accepting states. The state transition is as follows:


Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automaton?
A. 10110
B. 10010
C. 01010
D. 01001

Answer ||| B
Solution |||
In q0 state for ' 1 ', a ' 1 ' is pushed and for a ' 0 ' a ' 0 ' is pushed. In q1 state, for a ' 0 ' a ' 1 ' is popped and for a ' 1 ' a ' 0 ' is popped. So, the given PDA is accepting all strings of the form x0x'r or x1x'r or xx'r, where $x$ 'r is the reverse of the 1 's complement of $x$. The given string 101100 has 6 letters and we are given 5 letter strings. So, x0 is done, with $x=10110$. So, $x^{\prime} r=(01001) r=10010$.
Hence option $B$ is correct.
41. Let $G=(V, E)$ be a simple undirected graph, and $s$ be a particular vertex in it called the source. For $x \in V$, let $d(x)$ denote the shortest distance in $G$ from $s$ to $x$. A breadth first search (BFS) is performed starting at $s$. Let $T$ be the resultant BFS tree. If ( $u, v$ ) is an edge of $G$ that is not in $T$, then which one of the following CANNOT be the value of $d(u)-d(v)$ ?
A. -1
B. 0
C. 1
D. 2

Answer ||| D
Solution |||
Note that the given graph is undirected, so an edge ( $u, v$ ) also means ( $v, u$ ) is also an edge. Since a shorter path can always be obtained by using edge ( $u, v$ ) or ( $v, u$ ), the difference between $d(u)$ and $d(v)$ cannot be more than 1
42. Consider a uniprocessor system executing three tasks $T_{1}, T_{2}$ and $T_{3}$, each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period, and the available tasks are scheduled in order of priority, with the highest priority task scheduled first. Each instance of $T_{1}$, $T_{2}$ and $T_{3}$ requires an execution time of 1,2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the $1^{\text {st }}$ millisecond and task preemptions are allowed, the first instance of $T_{3}$ completes its execution at the end of milliseconds.
A. 5
B. 10
C. 12
D. 15

Answer ||| C
Solution |||
Periods of T1, T2 and T3 are $3 \mathrm{~ms}, 7 \mathrm{~ms}$ and 20 ms
Since priority is inverse of period, T1 is the highest priority task, then T2 and finally T3
Every instance of T 1 requires 1 ms that of T 2 requires 2 ms and that of T 3 requires 4 ms
Initially all T1, T2 and T3 are ready to get processor,
T1 is preferred
Second instances of T1, T2, and T3 shall arrive at 3, 7, and 20 respectively.
Third instance of T1, T2 and T3 shall arrive at 6, 14, and 49 respectively.

Time-Interval Tasks

| $0-1$ | T1 |
| :---: | :--- |
| $1-2$ | T2 |
| $2-3$ | T2 |
| $3-4$ | T1 [Second Instance of T1 arrives] |
| $4-5$ | T3 |
| $5-6$ | T3 |
| $6-7$ | T1 [Third Instance of T1 arrives] |
|  | [Therefore T3 is preempted] |
| $7-8$ | T2 [Second instance of T2 arrives] |
| $8-9$ | T2 |
| $9-10$ | T1 [Fourth Instance of T1 arrives] |
| $10-11$ | T3 |
| $11-12$ | T3 [First Instance of T3 completed] |

43. A positive edge-triggered $D$ flip-flop is connected to a positive edge-triggered JK flip-flop as follows. The $Q$ output of the D flip-flop is connected to both the J and K inputs of the JK flip-flop, while the $Q$ output of the JK flip-flop is connected to the input of the D flip-flop. Initially, the output of the D flip-flop is set to logic one and the output of the JK flip-flop is cleared. Which one of the following is the bit sequence (including the initial state) generated at the $Q$ output of the JK flip-flop when the flip-flops are connected to a free-running common clock? Assume that $\mathrm{J}=\mathrm{K}=1$ is the toggle mode and J $=\mathrm{K}=0$ is the state-holding mode of the JK flip-flop.
Both the flip-flops have non-zero propagation delays.
A. 0110110...
B. 0100100...
C. 011101110...
D. 011001100...

Answer ||| A
Solution |||


Initially Q output of D - FF = 1 Initially Q output of JK FF $=0$ Now with the help of present state and next state table we can see what is happening in circuit.
Toggle: $J=K=1$
Hold: J = K = 0

| PRESENT STATE D INPUT | Q(OUT PUT OF D-FF) | PREVIOUS <br> STATE Q <br> OF JK-FF | INPUT TO JK-FF |  | NEXT STATE Q OF JK-FF |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  |  |  |
|  |  |  | J | K |  |
| O(initially) | 0 | 0 | 1 | 1 | 1(toggled) |
| 1 | 1 | 1 | 0 | 0 | 1(held) |
| 1 | 1 | 1 | 1 | 1 | 0(toggled) |
| 0 | 0 | 0 | 1 | 1 | 1(toggled) |
| 1 | 1 | 1 | 0 | 0 | 1(held) |
| 1 | 1 | 1 | 1 | 1 | 0(toggled) |

We see from table Q output of D-FF is going to next state input of JK-FF and the bits sequence produced is like 110110..... Including initial condition (0) we get output as $0110110110 . .$. . Hence answer is (A) part.
44. Consider a disk pack with a seek time of 4 milliseconds and rotational speed of 10000 rotations per minute (RPM). It has 600 sectors per track and each sector can store 512 bytes of data. Consider a file stored in the disk. The file contains 2000 sectors. Assume that
every sector access necessitates a seek, and the average rotational latency for accessing each sector is half of the time for one complete rotation. The total time (in milliseconds) needed to read the entire file is
A. 14020
B. 14000
C. 25030
D. 15040

Answer ||| A
Solution |||
Seek time (given) $=4 \mathrm{~ms}$
RPM $=10000$ rotation in 1 min [60 sec]
So, 1 rotation will be $=60 / 10000=6 \mathrm{~ms}$ [rotation speed]
Rotation latency $=1 / 2 * 6 \mathrm{~ms}=3 \mathrm{~ms}$
\# To access a file,
total time includes $=$ seek time + rot. latency +transfer time
TO calc. transfer time, find transfer rate
Transfer rate $=$ bytes on track /rotation speed
so, transfer rate $=600 * 512 / 6 \mathrm{~ms}=51200 \mathrm{~B} / \mathrm{ms}$
transfer time $=$ total bytes to be transferred/ transfer rate so, Transfer time $=2000 * 512 / 51200=20 \mathrm{~ms}$
Given as each sector requires seek tim + rot. Latency $=$ $4 \mathrm{~ms}+3 \mathrm{~ms}=7 \mathrm{~ms}$
Total 2000 sector takes $=2000 * 7 \mathrm{~ms}=14000 \mathrm{~ms}$
To read entire file ,total time $=14000+20$ (transfer time) $=14020 \mathrm{~ms}$
45. Consider a non-pipelined processor with a clock rate of 2.5 gigahertz and average cycles per instruction of four. The same processor is upgraded to a pipelined processor with five stages; but due to the internal pipeline delay, the clock speed is reduced to 2 gigahertz. Assume that there are no stalls in the pipeline. The speed up achieved in this pipelined processor is
A. 3.2
B. 3.0
C. 2
D. 2.5

Answer ||| A
Solution |||
Speedup $=$ ExecutionTime ${ }_{o l d} / E^{\text {ExecutionTime }}{ }_{\text {New }}$
ExecutionTime ${ }_{\text {old }}=\mathrm{CPI}_{\text {old }} *$ CycleTime ${ }_{\text {old }}$
[Here CPI is Cycles Per Instruction]
$=$ CPI $_{\text {old }} *$ CycleTimeold
$=4 * 1 / 2.5$ Nanoseconds
$=1.6 \mathrm{~ns}$
Since there are no stalls, $C P U_{\text {new }}$ can be assumed 1 on average.
ExecutionTime $_{\text {New }}=$ CPI $_{\text {new }} *$ CycleTime ${ }_{\text {new }}$
$=1 * 1 / 2$
$=0.5$
Speedup $=1.6 / 0.5=3.2$
46. Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80, 25, 70. Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming
that SCAN algorithm moves towards 100 when it starts execution) is $\qquad$ tracks.
A. 8
B. 9
C. 10
D. 11

Answer ||| C
Solution |||
In Shortest seek first (SSTF), closest request to the current position of the head, and then services that request next. In SCAN (or Elevator) algorithm, requests are serviced only in the current direction of arm movement until the arm reaches the edge of the disk. When this happens, the direction of the arm reverses, and the requests that were remaining in the opposite direction are serviced, and so on.
Given a disk with 100 tracks
And Sequence 45, 20, 90, 10, 50, 60, 80, 25, 70
Initial position of the R/W head is on track 50

| In SSTF, requests are served |  |
| :---: | :---: |
| Next Served | Distance Traveled |
| 50 | 0 |
| 45 | 5 |
| 60 | 15 |
| 70 | 10 |
| 80 | 10 |
| 90 | 10 |
| 25 | 65 |
| 20 | 5 |
| 10 | 10 |
| $-\cdots-\cdots-\cdots-\cdots$ |  |
| Total Dist | $=130$ |

## If Simple SCAN is used, requests are served as

 following| Next Served | Distance Traveled |
| :---: | :---: |
| 50 | 0 |
| 60 | 10 |
| 70 | 10 |
| 80 | 10 |
| 90 | 10 |
| 45 | 65 [disk arm goes to 100, then to 45] |
| 25 | 20 |
| 20 | 5 |
| 10 | 10 |

Extra Distance traveled in SSTF $=140-120=-10$
If SCAN with LOOK is used, requests are served as following

| Next Served | Distance Traveled |
| :---: | :---: |
| 50 | 0 |
| 60 | 10 |
| 70 | 10 |
| 80 | 10 |
| 90 | 10 |
| 45 | 45 [disk arm comes back from 90] |
| 25 | 20 |
| 20 | 5 |
| 10 | 10 |

Total Dist $=120$
Extra Distance traveled in SSTF $=130-120=10$
47. Consider a main memory with five page frames and the following sequence of page references: $3,8,2,3$, $9,1,6,3,8,9,3,6,2,1,3$. Which one of the following is true with respect to page replacement
policies First In First Out (FIFO) and Least Recently Used (LRU)?
A. Both incur the same number of page faults
B. FIFO incurs 2 more page faults than LRU
C. LRU incurs 2 more page faults than FIFO
D. FIFO incurs 1 more page faults than LRU

Answer ||| A
Solution |||
$3,8,2,3,9,1,6,3,8,9,3,6,2,1,3$
In both FIFO and LRU, we get following after considering 3, 8, 2, 3, 9, 1, 38291
FIFO
6 replace 3
82916

3 replace 8
29163

8 replace 2
91638

2 replace 9
16382

No more page faults
LRU
6 replace 8
32916

8 replace 2
39168

2 replace 1
39682

1 replaces 8
39621
$\int_{1 / \pi}^{2 / \pi} \frac{\cos (1 / x)}{x^{2}} d x=$
A. 0
B. -1
C. 1
D. infinite

Answer ||| B
49. Consider the following $2 \times 2$ matrix $A$ where two elements are unknown and are marked by $a$ and $b$. The eigenvalues of this matrix are -1 and 7 . What are the values of $a$ and $b$ ?
$A=\left(\begin{array}{ll}1 & 4 \\ b & a\end{array}\right)$.
A. $a=6, b=4$
B. $a=4, b=6$
C. $a=3, b=5$
D. $a=5, b=3$

Answer ||| D
Solution |||
The character equation for given matrix is
$|1-\lambda 4|=0$
| ba- $\lambda$ |
$(1-\lambda)^{*}(a-\lambda)-4 b=0$
Putting $\lambda=-1$,
$=>(1-(-1)) *(a-(-1))-4 b=0$
$=>2 a+2-4 b=0$
$=>2 b-a=1$
Putting $\lambda=7$,
$=>(1-7) *(a-7)-4 b=0$
$=>-6 a+42-4 b=0$
$=>2 b+3 a=21$
Solving the above two equations, we get
$a=5, b=3$
50. An algorithm performs $(\log N)^{1 / 2}$ find operations, $N$ insert operations, $(\log N)^{1 / 2}$ delete operations, and (log $\mathrm{N})^{1 / 2}$ decrease-key operations on a set of data items with keys drawn from a linearly ordered set. For a delete operation, a pointer is provided to the record that must be deleted.
For the decrease-key operation, a pointer is provided to the record that has its key decreased. Which one of the following data structures is the most suited for the algorithm to use, if the goal is to achieve the best total asymptotic complexity considering all the operations?
A. Unsorted array
B. Min-heap
C. Sorted array
D. Sorted doubly linked list

Answer ||| A
Solution |||
The time complexity of insert in unsorted array is $\mathrm{O}(1)$, $\mathrm{O}($ Logn $)$ in Min-Heap, $\mathrm{O}(\mathrm{n})$ in sorted array and sorted DLL. Since number of insertion operations is asymptotically higher, unsorted array is preferred.
51. Consider the following relations:

| Student |  |
| :---: | :---: |
| Roll_No | Student_Name |
| 1 | Raj |
| 2 | Rohit |
| 3 | Raj |


| Performance |  |  |
| :---: | :---: | :---: |
| Roll_No | Course | Marks |
| 1 | Math | 80 |
| 1 | English | 70 |
| 2 | Math | 75 |
| 3 | English | 80 |
| 2 | Physics | 65 |
| 3 | Math | 80 |

Consider the following SQL query.

```
SELECT S.Student Name, sum(P.Marks)
FROM Student S, Performance P
WHERE S.Roll_No = P.Roll_No
GROUP BY S.Student_Name
```

The number of rows that will be returned by the SQL
query is $\qquad$ .
A. 0
B. 1
C. 2
D. 3

Answer ||| C
Solution |||
Below is result of given query. Note that there are only two student names and query prints sum(P. Marks) for every student.

| Student_Name | Marks |
| :---: | :---: |
| Raj | 310 |
| Rohit | 140 |

52. What is the output of the following C code? Assume that the address of $x$ is 2000 (in decimal) and an integer requires four bytes of memory.
int main () \{
unsigned int $\times[4][3]=$
$\{\{1,2,3\},\{4,5,6\},\{7,8,9\},\{10,11,12\}\}$;
printf("\%u, \%u, \%u", $\left.x+3,{ }^{*}(x+3),{ }^{*}(x+2)+3\right)$;
A. 2036, 2036, 2036
B. 2012, 4, 2204
C. 2036, 10, 10
D. 2012, 4, 6

Answer ||| A
Solution |||
$x=2000$
Since $x$ is considered as a pointer to an array of 3 integers and an integer take 4bytes, value of $x+3=$ $2000+3 * 3 * 4=2036$
The expression, $*(x+3)$ also prints same address as $x$ is 2D array
The expression $*(x+2)+3=2000+2 * 3 * 4+3 * 4=$ 2036
53. The graph shown below has 8 edges with distinct integer edge weights. The minimum spanning tree (MST) is of weight 36 and contains the edges: $\{(A, C),(B, C)$, $(B, E),(E, F),(D, F)\}$. The edge weights of only those edges which are in the MST are given in the figure shown below. The minimum possible sum of weights of all 8 edges of this graph is $\qquad$ -.

A. 66
B. 69
C. 70
D. 58

Answer ||| B
Solution |||

In every cycle, the weight of an edge that is not part of MST must by greater than or equal to weights of other edges which are part of MST. Since all edge weights are distinct, the weight must be greater. So, the minimum possible weight of ED is 7 , minimum possible weight of $C D$ is 16 and minimum possible weight of $A B$ is 10 . Therefore, minimum possible sum of weights is 69

```
54. Consider the following C function.
int fun1(int n) {
    int i,j,k,p,q=0;
    for (i=1; i<n; ++i) {
        p=0;
        for (j=n; j>1; j=j/2)
            ++p;
        for (k=1; k<p; k=k*2)
        ++q;
    }
    return q;
}
```

Which one of the following most closely approximates the
return value of the function fun1?
A. $n^{3}$
B. $n(\log n)^{2}$
C. $n \log n$
D. $n \log (\log n)$

Answer ||| D
Solution ||| int fun1 (int n)
\{
int $I, j, k, p, q=0 ;$
// This loop runs $\Theta(n)$ time
for ( $\mathrm{I}=1 ; \mathrm{I}<\mathrm{n} ;++\mathrm{I}$ )
\{
p = 0;
// This loop runs $\Theta(\log n)$ times. Refer this for ( $\mathrm{j}=\mathrm{n} ; \mathrm{j}>1 ; \mathrm{j}=\mathrm{j} / 2$ )
$++p ;$
// Since above loop runs $\Theta(\log n)$ times, $p=\Theta(\log n)$
// This loop runs $\Theta$ (Log $p$ ) times which loglogn
for ( $k=1 ; k<p ; k=k * 2$ )
$++q$;
\}
Return q;
\}
$\mathrm{T}(\mathrm{n})=\mathrm{n}(\log \mathrm{n}+\log \log \mathrm{n}) \mathrm{T}(\mathrm{n})=\mathrm{n}(\log \mathrm{n})$ dominant But please note here we are return $q$ which lies in loglogn so ans should be $T(n)=$ nloglogn

55 Consider the following pseudo code, where $x$ and $y$ are positive integers.

## begin

$\mathrm{q}:=0$
r := x
while $r \geq y$ do
begin
r := r - y
$q:=q+1$
end
end

The post condition that needs to be satisfied after the program terminates is
A. $\{r=q x+y \wedge r<y\}$
B. $\{x=q y+r \wedge r<y\}$
C. $\{y=q x+r \wedge 0<r<y\}$
D. $\{q+1<r-y \wedge y>0\}$

Answer ||| B
Solution |||
The given pseudo code does following for given $x$ and $y$ which positive integers.

1) It initializes $r$ as $x$.
2) It repeatedly subtracts $y$ from $r$ until $r$ becomes smaller than $y$. For every subtraction, it increments count q.
3) Finally $r$ contains remainder, i.e., $x \% y$ and $q$ contains [ $x / y$ ]
See below pseudo code with comments.
begin
$\mathrm{q}:=0 / / \mathrm{q}$ is going to contain floor $(\mathrm{x} / \mathrm{y})$
$r:=x / / r$ is going to contain $x \% y$
// Repeatedly subtract $y$ from $x$.
while $r>=y$ do
begin
$r:=r-y$
$\mathrm{q}:=\mathrm{q}+1$
end
end
