1. Consider the following two statements.
S1: If a candidate is known to be corrupt, then he will not be elected
S2: If a candidate is kind, he will be elected
WhichoneofthefollowingstatementsfollowsfromS1and S2aspersoundinferencerulesoflogic?
A. If a person is known to be corrupt, he is kind
B. If a person is not known to be corrupt, he is not kind
C. If a person is kind; he is not known to be corrupt
D. If a person is not kind, he is not known to be corrupt
Answer ||| C
Solution |||
S1: If a candidate is known to be corrupt, then He will not be elected.
S2: If a candidate is kind, he will be elected.
If p → q, then ¬q → ¬p
So from S1, elected → not corrupt
And S2 is, kind → elected
Therefore, kind → not corrupt

2. The cardinality of the power set of {0, 1, 2, 10} is_____.
A. 8
B. 10
C. 16
D. 14
Answer ||| C
Solution |||
Set has 4 elements. So power set will contain 2^4 elements.

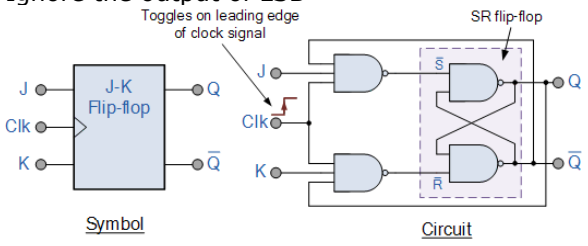
3. Let R be the relation on the set of positive integers such that aRb if and only if bear distinct and have a common divisor other than 1. Which one of the following statements about R is true?
A. Rissymmetricandreflexivebutnottransitive
B. Risreflexivebutnotsymmetricandnottransitive
C. Ristransitivebutnotreflexiveandnotsymmetric
D. Rissymmetricbutnotreflexiveandnottransitive
Answer ||| D
Solution |||
R cannot be reflexive as 'a' and 'b' have to be distinct in aRb. R is symmetric if a and b have a common divisor, then b and a also have. R is not transitive as aRb and bRc doesn't mean aRc. For example 3 and 15 have common divisor, 15 and 5 have common divisor, but 3 and 5 don't have.

4. The number of divisors of 2100 is.
A. 42
B. 36
C. 78
D. 72
Answer ||| B
Solution |||
$2100 = 2^2 * 3^1 * 5^2 * 7^1$
There are 3 choices for powers of 2; the choices are 0, 1, and 2
Similarly, there are 2 choices for powers of 3, 3 choices for power of 5
and 2 choices for power of 7.
So total number of divisors is 3 * 2 * 3 * 2 = 36

5. The larger of the two Eigen values of the matrix $\begin{bmatrix} 4 & 5 \\ 2 & 1 \end{bmatrix}$ is _____.
A. 5
B. 6
C. 7
D. 8
Answer ||| B
Solution |||
The character equation for given matrix is
$| 4-\lambda \quad 5 | = 0$
$| 2 \quad 1-\lambda |$
$(4-\lambda)*(1-\lambda) - 10 = 0$
$\lambda^2 - 5\lambda - 6 = 0$
$(\lambda+1)*(\lambda-6) = 0$
$\lambda = -1, 6$
Greater of two Eigen values are 6

6. An unordered list contains n distinct elements. The number of comparisons to find an element in this list that is neither maximum nor minimum is
A. θ(n log n)
B. θ(n)
C. θ(log n)
D. θ(1)
Answer ||| D
Solution |||
We only need to consider any 3 elements and compare them. So the number of comparisons is constants, that makes time complexity as Θ (1) The catch here is, we need to return any element that is neither maximum not minimum. Let us take an array {10, 20, 15, 7, and 90}. Output can be 10 or 15 or 20 Pick any three elements from given liar. Let the three elements be 10, 20 and 7. Using 3 comparisons, we can find that the middle element is 10.

7. The minimum number of JK flip-flops required to construct a synchronous counter with the count sequence (0,0,1,1,2,2,3,3,0,0,…) is_____.
A. 0
B. 1
C. 2
D. 3
Answer ||| D
Solution |||
Answer = 3, mod 8 up counter using 3 JK flip flops. Ignore the output of LSB



8. Assume that for a certain processor, a read request takes 50 nanoseconds on a cache miss and 5 nanoseconds on a cache hit. Suppose while running a program, it was observed that 80% of the processor's read requests result in a cache hit. The average read access time in nanoseconds is _____.
A. 10
B. 12
C. 13
D. 14
Answer ||| D
Solution |||
The average read access time in nanoseconds = 0.8 * 5 + 0.2*50 = 14. The time a program or device takes to locate a single piece of information and make it available to the computer for processing. *DRAM (dynamic random access memory)* chips for personal computers have access times of 50 to 150 nanoseconds (billionths of a second). *Static RAM (SRAM)* has access times as low as 10 nanoseconds. Ideally, the access time of memory should be fast enough to keep up with the CPU. If not, the CPU will waste a certain number of clock cycles, which makes it slower.

9. A computer system implements a 40-bit virtual address, page size of 8 kilobytes, and a 128-entry translation look-aside buffer (TLB) organized into 32 sets each having four ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is_____.
A. 20
B. 10
C. 11
D. 22
Answer ||| C
Solution |||
Total virtual address size = 40
Since there are 32 sets, set offset = 5
Since page size is 8kilobytes, word offset = 13
Minimum tag size = 40 - 5- 13 = 22

10. Consider the following statements.
I. The complement of every Turing decidable language is Turing decidable
II. There exists some language which is in NP but is not Turing decidable
III. If L is a language in NP, L is Turing decidable
Which of the above statements is/are true?
A. Only II
B. Only III
C. Only I and II
D. Only I and III
Answer ||| D
Solution |||
1 is true: Complement of Turing decidable is Turing Decidable.
3 are true. All NP problems are Turing decidable
2 is false: The definition of NP itself says solvable in
Polynomial time using non-deterministic Turing machine.
An algorithm is said to be solvable in polynomial time if the number of steps required to complete the algorithm for a given input is for some nonnegative integer, where is the complexity of the input. Polynomial-time algorithms are said to be "fast."

11. Consider the following function written in the C programming language.

```
void foo(char *a){
    if ( *a && *a != ' '){
        foo(a+1);
        putchar(*a);
    }
}
```

The output of the above function on input "ABCD EFGH" is
A. ABCD EFGH
B. ABCD
C. HGFE DCBA
D. DCBA
Answer ||| D
Solution |||
The program prints all characters before ' ' or '\0' (whichever comes first) in reverse order. The first line of the program #includes <stdio.h> is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation. The next line int main() is the main function where the program execution begins.

12. Consider a complete binary tree where the left and the right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is
A. Ω (log n)
B. Ω (n)
C. Ω (n log n)
D. Ω (n^2)
Answer ||| A
Solution |||
The answer to this question is simply max-heapify function. Time complexity of max-heapify is O(Log n) as it recourses at most through height of heap.

```
// A recursive method to heapify a sub tree with root at given index
// this method assumes that the sub trees are already heapified
Void MinHeap::MaxHeapify(int i)
{
Int l = left(i);
Int r = right(i);
Int largest = i;
If (l < heap size && harr[l] < harr[i])
Largest = l;
If (r < heap size && harr[r] < harr[smallest])
Largest = r;
If (largest != i)
{
Swap(&harr[i], &harr[largest]);
MinHeapify(largest);
}
}
```

13. A binary tree T has 20 leaves. The number of nodes in T having two children is _____.
A. 18
B. 19
C. 20
D. 17
Answer ||| B
Solution |||
The number of nodes with two children is always one less than the number of leaves. In computer science, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. Binary trees are seldom used solely for their structure. Much more typical is to define a labeling function on the nodes, which associates some value to each node. Binary trees labeled this way are used to implement binary search trees and binary heaps, and are used for efficient searching and sorting. The designation of non-root nodes as left or right child even when there is only one child present matters in some of these applications, in particular it is significant in binary search trees. In mathematics, what is termed binary tree can vary significantly from author to author. Some use the definition commonly used in computer science, but others define it as every non-leaf having exactly two children and don't necessarily order (as left/right) the children either.

14. Consider the following C function.

```
int fun(int n){
      int x=1,k;
      if (n==1) return x;
      for (k=1; k<n; ++k)
            x = x + fun(k) * fun(n-k);
      return x;
}
```

The return value of fun (5) is _____.
A. 0
B. 26
C. 51
D. 22
Answer ||| C
Solution |||
Fun(5) = 1 + fun(1) * fun(4) + fun(2) * fun(3) +
Fun(3) * fun(2) + fun(4) * fun(1)
= 1 + 2*[fun(1)*fun(4) + fun(2)*fun(3)]
Substituting fun(1) = 1
= 1 + 2*[fun(4) + fun(2)*fun(3)]
Calculating fun(2), fun(3) and fun(4)
Fun(2) = 1 + fun(1)*fun(1) = 1 + 1*1 = 2
Fun(3) = 1 + 2*fun(1)*fun(2) = 1 + 2*1*2 = 5
Fun(4) = 1 + 2*fun(1)*fun(3) + fun(2)*fun(2)
= 1 + 2*1*5 + 2*2 = 15
Substituting values of fun(2), fun(3) and fun(4)
Fun(5) = 1 + 2*[15 + 2*5] = 51

15. A software requirements specification (SRS) document should avoid discussing which one of the following?
A. User interface issues
B. Non-functional requirements
C. Design specification
D. Interfaces with third party software
Answer ||| C
Solution |||
Software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have with the software. Design Specification should not be part of SRS.

16. Consider two decision problems Q_1, Q_2 such that Q_1 reduces in polynomial time to 3-SAT and 3-SAT reduces in polynomial time to Q_2. Then which one of the following is consistent with the above statement?
A. Q_1 is in NP, Q_2 is NP hard.
B. Q_2 is in NP, Q_1 is NP hard.
C. Both Q_1 and Q_2 are in NP.
D. Both Q_1 and Q_2 are NP hard.
Answer ||| A
Solution |||
Q1 reduces in polynomial time to 3-SAT
==> Q1 is in NP
3-SAT reduces in polynomial time to Q2
==> Q2 is NP Hard. If Q2 can be solved in P, then 3-SAT
Can be solved in P, but 3-SAT is NP-Complete, that makes Q2 NP Hard

17. Match the following:
List – I
P. Lexical analysis
Q. Parsing
R. Register allocation
S. Expression evaluation
List – II
I. Graph coloring
II. DFA minimization
III. Post-order traversal
IV. Production tree
A. P-II, Q-III, R-I, S-IV
B. P-II, Q-I, R-IV, S-III
C. P-II, Q-IV, R-I, S-III
D. P-II, Q-III, R-IV, S-I
Answer ||| C
Solution |||
Register allocation is a variation of Graph Coloring problem.
Lexical Analysis uses DFA.
Parsing makes production tree
Expression evaluation is done using tree traversal.
In graph theory, graph coloring is a special case of graph labeling; it is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color; this is called a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that no two adjacent edges share the same color, and a face coloring of a planar graph assigns a color to each face or region so that no two faces that share

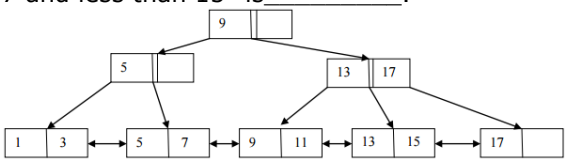
a boundary have the same color. Vertex coloring is the starting point of the subject, and other coloring problems can be transformed into a vertex version. For example, an edge coloring of a graph is just a vertex coloring of its line graph, and a face coloring of a plane graph is just a vertex coloring of its dual. However, non-vertex coloring problems are often stated and studied as is. That is partly for perspective, and partly because some problems are best studied in non-vertex form, as for instance is edge coloring.

18. In the context of abstract-syntax-tree (AST) and control-flow-graph (CFG), which one of the following is TRUE?
A. In both AST and CFG, let node N_2 be the successor of node N_1. In the input program, the code corresponding to N_2 is present after the code corresponding to N_1
B. For any input program, neither AST nor CFG will contain a cycle
C. The maximum number of successors of a node in an AST and a CFG depends on the input program
D. Each node in AST and CFG corresponds to at most one statement in the input program
Answer ||| C
Solution |||
(A) is false, In CFG (Control Flow Graph), code of N2 May be present before N1 when there is a loop or goto.
(B) is false, CFG (Control Flow Graph) contains cycle when input program has loop.
(C) is true, successors in AST and CFG depedend on input program
(D) is false, a single statement may belong to a block of statements.

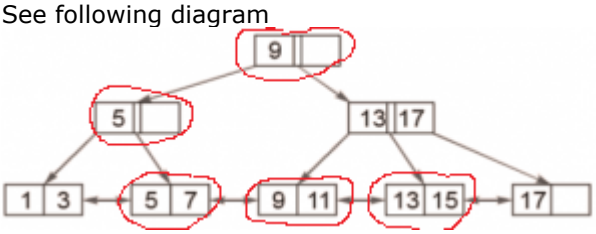
19. Consider the basic COCOMO model where E is the effort applied in person-months, D is the development time in chronological months, KLOC is the estimated number of delivered lines of code (in thousands) and a_b, b_b, c_b, d_b have their usual meanings. The basic COCOMO equations are of the form
A. $E = a_b (KLOC)exp(b_b)$, $D = c_b (E) exp(d_b)$
B. $D = a_b (KLOC)exp(b_b)$, $E = c_b (D) exp(d_b)$
C. $E = a_b exp(b_b)$, $D = c_b (KLOC) exp(d_b)$
D. $E = a_b exp(d_b)$, $D = c_b (KLOC) exp(b_b)$
Answer ||| A
Solution |||
In Basic COCOMO, following are true.
Effort Applied (E) = $a_b(KLOC)^{b_b}$ [person-months]
Development Time (D) = $c_b(\text{Effort Applied})^{d_b}$ [months]
People required (P) = Effort Applied / Development Time [count]
20. A system has 6 identical resources and *N* processes competing for them. Each process can request atmost 2 resources. Which one of the following values of *N* could lead to a deadlock?
A. 1
B. 2
C. 3
D. None
Answer ||| D
Solution |||
P1 --> 1
P2 --> 1
P3 --> 1
P4 --> 1
Even 4 processes can be in safe with 5 resources. So upto 5 processes there will be no deadlock.

21. Consider the following transaction involving two bank accounts xand y. Read (x); x: = x - 50; write(x); read(y); y: = y + 50; write(y)
The constraint that the sum of the accounts *x* and *y* should remain constant is that of
A. Atomicity
B. Consistency
C. Isolation
D. Durability
Answer ||| B
Solution |||
Consistency in database systems refers to the requirement that any given database transaction must only change affected data in allowed ways, that is sum of x and y must not change. Consistency in database systems refers to the requirement that any given database transaction must change affected data only in allowed ways. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof.

22. With reference to the B+ tree index of order 1 shown below, the minimum number of nodes (including the Root node) that must be fetched in order to satisfy the following query: "Get all records with a search key greater than or equal to 7 and less than 15" is_____.



A. 4
B. 5
C. 6
D. 7
Answer ||| B
Solution |||
We can get all values in range from 7 to 59 by accessing 5 nodes.
1) First search 7 in a leaf node.
2) Once 7 is found, linearly traverse till 15 is found.
See following diagram



23. Identify the correct order in which a server process must invoke the function calls accept, bind, listen, and recv according to UNIX socket API.
A. listen, acc ept, bind, recv
B. bind, listen, accept, recv
C. bind, acce pt, listen, recv
D. accept, listen, bind, recv
Answer ||| B
Solution |||
bind, listen, accept and recv are server side socket API functions.
bind() associates a socket with a socket address structure,
i.e. a specified local port number and IP address.
listen() causes a bound TCP socket to enter listening state.
accept() accepts a received incoming attempt to create a new
TCP connection from the remote client,
recv() is used to receive data from a remote socket.
A server must first do bind() to tell operating system the port number on which it would be listing, then it must listen to receive incoming connection requests on the bound port number. Once a connection comes, the server accepts using accept(), then starts receiving data using recv().

24. A link has a transmission speed of 10^6 bits/sec. It uses data packets of size 1000 bytes each. Assume that the acknowledgment has negligible transmission delay, and that its propagation delay is the same as the data propagation delay. Also assume that the processing delays at nodes are negligible. The efficiency of the stop-and-wait protocol in this setup is exactly 25%. The value of the one-way propagation delay (in milliseconds) is _____.
A. 4
B. 8
C. 12
D. 16
Answer ||| C
Solution |||
In stop and wait, protocol next packet is sent only when acknowledgement of previous packet is received. This
causes poor link utilization.
Transmission speed = 10^6
Time to send a packet = (1000 * 8) bits / 10^6
= 8 milliseconds
Since link utilization or efficiency is 25%, total time
Taken for 1 packet is 8 * 100/25 = 32 milliseconds.
Total time is twice the one way propagation delay plus transmission delay. Propagation delay has to be considered for packet and ack both.
Transmission delay is considered only for packet as the question says that trans. time for ack is negligible.
Let propagation delay be x.
2x + 8 = 32
x = 12

25. Which one of the following statements is NOT correct about HTTP cookies?
A. A cookie is a piece of code that has the potential to compromise the security of an Internet user
B. A cookie gains entry to the user's work area through an HTTP header
C. A cookie has an expiry date and time
D. Cookies can be used to track the browsing pattern of a user at a particular site
Answer ||| A
Solution |||
Cookies are not piece of code, they are just strings typically in the form of key value pairs.
Cookies are data, stored in small text files, on your computer.
When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.
Cookies were invented to solve the problem "how to remember information about the user":
When a user visits a web page, his name can be stored in a cookie.
Next time the user visits the page, the cookie "remembers" his name.

26. Consider the following routing table at an IP router:

Network No.	Net Mask	Next Hop
128.96.170.0	255.255.254.0	Interface 0
128.96.168.0	255.255.254.0	Interface 1
128.96.166.0	255.255.254.0	R2
128.96.164.0	255.255.252.0	R3
0.0.0.0	Default	R4

For each IP address in Group I identify the correct choice of the next hop from Group II using the entries from the routing table above.
Group I
i) 128.96.171.92
ii) 128.96.167.151
iii) 128.96.163.151
iv) 128.96.165.121
Group II
a) Interface 0
b) Interface 1
c) R2
d) R3
e) R4
A. i-a, ii-c, iii-e, iv-d
B. i-a, ii-d, iii-b, iv-e
C. i-b, ii-c, iii-d, iv-e
D. i-b, ii-c, iii-e, iv-d

Answer ||| A
Solution |||
The next hop is decided according to the longest prefix matching. Next hop is a routing term that refers to the next closest router a packet can go through. The next hop is among the series of routers that are connected together in a network and is the next possible destination for a data packet. The Internet consists of thousands of different networks of every size and shape. Routers

are among the most important and significant network devices in this network in that they hold the key to the rapid growth of the Internet worldwide, enabling communication among the devices. Therefore, a router has to manage the information related to its topological surroundings, specifically about nearby routers. Whenever a router maintains information about the routers in its routing table, the lowest metric among them is known as the next hop or the next optimal router.

27. Host A sends a UDP datagram containing 8880 bytes of user data to host B over an Ethernet LAN. Ethernet frames may carry data up to 1500 bytes (i.e. MTU=1500 bytes). Size of UDP header is 8 bytes and size of IP header is 20 bytes. There is no option field in IP header. How many total number of IP fragments will be transmitted and what will be the contents of offset field in the last fragment?
A. 6 and 925
B. 6 and 7400
C. 7 and 1110
D. 7 and 8880
Answer ||| C
Solution |||
UDP data = 8880 bytes
UDP header = 8 bytes
IP Header = 20 bytes
Total Size excluding IP Header = 8888 bytes.
Number of fragments = ⌈ 8888 / 1480 ⌉
= 7
Refer the Kurose book slides on IP (Offset is always scaled by 8)
Offset of last segment = (1480 * 6) / 8 = 1110

28. Assume that the bandwidth for a TCP connection is 1048560 bits /sec. Let α be the value of RTT in milliseconds (rounded off to the nearest integer) after which the TCP window scale option is needed. Let β be the maximum possible window size with window scale option. Then the values of α and β are
A. 63 milliseconds, 65535 x 2^{14}
B. 63 milliseconds, 65535 x 2^{16}
C. 500 milliseconds, 65535 x 2^{14}
D. 500 milliseconds, 65535 x 2^{16}
Answer ||| C
Solution |||
Since sequence number in TCP header is limited to 16 bits, the maximum window size is limited. When bandwidth delay product of a link is high, scaling is required to efficiently use link. TCP allows scaling of windows when bandwidth delay product is greater than 65,535 (Refer this). The bandwidth delay product for given link is 1048560 * α. Window scaling is needed when this value is more than 65535 bytes, i.e., when α is greater than 65535 * 8 / 1048560 or 0.5 seconds. Scaling is done by specifying a one byte shift count in the header options field. The true receive window size is left shifted by the value in shift count. A maximum value of 14 may be used for the shift count value. Therefore maximum window size with scaling option is 65535 × 2^{14}.

29. Consider a simple checkpointing protocol and the following set of operations in the log.
(start, T4); (write, T4, *y*, 2, 3); (start, T1); (commit, T4); (write, T1, *z*, 5, 7); (checkpoint); (start, T2); (write, T2, *x*, 1, 9); (commit, T2); (start, T3), (write, T3, *z*, 7, 2);
If a crash happens now and the system tries to recover using both undo and redo operations, what are the contents of the undo list and the redo list?
A. Undo: T3, T1; Redo: T2
B. Undo: T3, T1; Redo: T2, T4
C. Undo: none; Redo: T2, T4, T3, T1
D. Undo: T3, T1, T4; Redo: T2
Answer ||| A
Solution |||
Since T1 and T3 are not committed yet, they must be undone. The transaction T2 must be redone because it is after the latest checkpoint.
A transaction symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database. Transactions in a database environment have two main purposes:
To provide reliable units of work that allow correct recovery from failures and keep a database consistent even in cases of system failure, when execution stops (completely or partially) and many operations upon a database remain uncompleted, with unclear status.
To provide isolation between programs accessing a database concurrently. If this isolation is not provided, the programs' outcomes are possibly erroneous.

30. Consider two relations R_1(A, B) with the tuples (1,5), (3,7) and R_2 (A, C) = (1,7), (4,9). Assume that R(A, B, C) is the full natural outer join of R_1 and R_2. Consider the following tuples of the form (A, B, C): *a* = (1, 5, *null*), *b* = (1, *null,* 7), *c* = (3, *null,* 9), *d* = (4, 7, *null), e* = (1,5,7), *f* = (3,7,*null), g* = (4,*null,*9). Which one of the following statements is correct?
A. R contains *a, b, e, f, g* but not *c, d.*
B. R contains all of *a, b, c, d, e, f, g.*
C. R contains *e, f, g* but not *a, b.*
D. R contains *e* but not *f, g.*
Answer ||| C
Solution |||
Below is R1

A	B
1	5
3	7

Below is R2

A	C
1	7
4	9

Full outer join of above two is

A	B	C
1	5	7
3	7	NULL
4	NULL	9

So the full outer join contains e = (1, 5, 7), f = (3, 7, null), g = (4, null, 9).

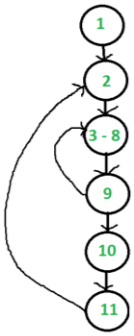
31. Consider six memory partitions of sizes 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. If the best fit algorithm is used, which partitions are NOT allotted to any process?
A. 200 KB and 300 KB
B. 200 KB and 250 KB
C. 250 KB and 300 KB
D. 300 KB and 400 KB
Answer ||| A
Solution |||
Best fit allocates the smallest block among those that are large enough for the new process. So the memory blocks are allocated in below order.
357 ---> 400
210 ---> 250
468 ---> 500
491 ---> 600

32. Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50×10^6 bytes/sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512-byte sector of the disk is _____.
A. 6.1
B. 4
C. 5.2
D. 5
Answer ||| A
Solution |||
Disk latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead
Seek Time? Depends no. tracks the arm moves and seek speed of disk
Rotation Time? depends on rotational speed and how far the sector is from the head
Transfer Time? depends on data rate (bandwidth) of disk (bit density) and the size of request
Disk latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead
Average Rotational Time = (0.5)/(15000 / 60) = 2 milliseconds
[On average half rotation is made]
It is given that the average seek time is twice the average rotational delay
So Avg. Seek Time = 2 * 2 = 4 milliseconds.
Transfer Time = 512 / (50×10^6 bytes/sec)
= 10.24 microseconds
Given that controller time is 10 times the average transfer time
Controller Overhead = 10 * 10.24 microseconds = 0.1 milliseconds
Disk latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead
= 4 + 2 + 10.24 * 10^{-3} + 0.1 milliseconds = 6.1 milliseconds

33. A computer system implements 8 kilobyte pages and a 32-bit physical address space. Each page table entry contains a valid bit, a dirty bit, three permission bits, and the translation. If the maximum size of the page table of a process is 24 megabytes, the length of the virtual address supported by the system is_____bits.
A. 36
B. 56
C. 30
D. 42
Answer ||| A
Solution |||
Max size of virtual address can be calculated by calculating maximum number of page table entries. Maximum Number of page table entries can be calculated using given maximum page table size and size of a page table entry.
Given maximum page table size = 24 MB
Let us calculate size of a page table entry.
A page table entry has following number of bits.
1 (valid bit) +
1 (dirty bit) +
3 (permission bits) +
x bits to store physical address space of a page.
Value of x = (Total bits in physical address) - (Total bits for addressing within a page)
Since size of a page is 8 kilobytes, total bits needed within
a page is 13.
So value of x = 32 - 13 = 19
Putting value of x, we get size of a page table entry =
1 + 1 + 3 + 19 = 24bits.
Number of page table entries
= (Page Table Size) / (An entry size)
= (24 megabytes / 24 bits)
= 2^{23}
Virtual address Size
= (Number of page table entries) * (Page Size)
= 2^{23} * 8 kilobits
= 2^{36}
Therefore, length of virtual address space = 36

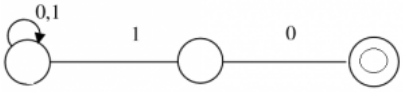
34. Consider the intermediate code given below.
(1) i = 1
(2) j = 1
(3) t1 = 5 * i
(4) t2 = t1 + j
(5) t3 = 4 * t2
(6) t4 = t3
(7) a[t4] = -1
(8) j = j + 1
(9) if j<=5 goto (3)
(10) i=i+1
(11) if i<5 goto (2)

The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are
A. 5 and 7
B. 6 and 7
C. 5 and 5
D. 7 and 8
Answer ||| B
Solution |||

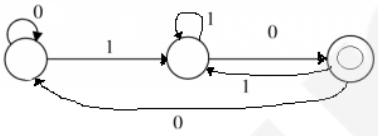


The above is the control flow graph.

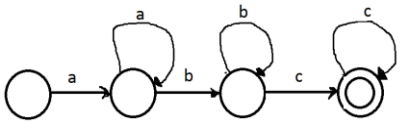
35. The number of states in the minimal deterministic finite automaton corresponding to the regular expression $(0 + 1)^*(10)$ is_____.
A. 2
B. 3
C. 4
D. 5
Answer ||| B
Solution |||
Below is NFA for the given regular expression (0 + 1)*(10)



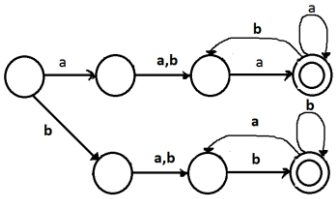
Below is DFA for the same.



36. Which of the following languages is/are regular?
L_1: $\{wxw^R \mid w, x \in \{a, b\}^*$ and $|w|, |x| > 0\}$, w^R is the reverse of string w
L_2: $\{a^n b^m \mid m \neq n$ and $m, n \geq 0\}$
L_3: $\{a^p b^q c^r \mid p, q, r \geq 0\}$
A. L_1 and L_3 only
B. L_2 only
C. L_2 and L_3 only
D. L_3 only
Answer ||| A
Solution |||
L3 is simple to guess, it is regular.
Below is DFA for L1.



L1 is interesting. The important thing to note about L1 is length of x is greater than 0, i.e., |x| > 0.
So any string than starts and ends with same character
is acceptable by language and remaining string becomes w.
Below is DFA for L1.



37. Given below are some algorithms, and some algorithm design paradigms.
List - I
P. Dijkstra's Shortest Path
Q. Floyd-Warshall algorithm to compute all pairs shortest path
R. Binary search on a sorted array
S. Backtracking search on a graph
List – II
i. Divide and Conquer
ii. Dynamic Programming
iii. Greedy design
iv. Depth-first search
v. Breadth-first search
Match the above algorithms on the left to the corresponding design paradigm they follow.
A. P-i, Q-iii, R-i, S-v.
B. P-iii, Q-iii, R-i, S-v.
C. P-iii, Q-ii, R-i, S-iv.
D. P-iii, Q-ii, R-i, S-v.
Answer ||| C
Solution |||
Dijkstra's Shortest Path is a Greedy Algorithm.
Floyd-Warshall algorithm is Dynamic Programming.
Binary search is a Divide and Conquer.
Backtracking is Depth-first search.
Mark visited (set to red) when done with neighbors. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks

38. A Young tableau is a 2D array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with ∞, and hence there cannot be any entry to the right of, or below a ∞. The following Young tableau consists of unique entries.

1	2	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled in with a ∞). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is _____.

A. 2
B. 5
C. 6
D. 8
Answer ||| B
Solution |||
Initially

1	2	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When 1 is removed, it is replaced by the smallest adjacent which is 2.

2	2	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When 2 is moved in place of 1, it is replaced by smallest adjacent which is 4

2	4	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When 4 is moved in place of 2, it is replaced by smallest adjacent which is 6

2	4	5	14
3	6	6	23
10	12	18	25
31	∞	∞	∞

When 6 is moved in place of 4, it is replaced by smallest adjacent which is 18.

2	4	5	14
3	6	18	23
10	12	18	25
31	∞	∞	∞

When 18 is moved in place of 6, it is replaced by smallest adjacent which is 25.

2	4	5	14
3	6	18	23
10	12	25	25
31	∞	∞	∞

When 25 is moved in place of 18, it is replaced by smallest adjacent which is ∞.

2	4	5	14
3	6	18	23
10	12	25	∞
31	∞	∞	∞

Shifted numbers are 2, 4, 6, 18, 25

39. Suppose you are provided with the following function declaration in the C programming language.
int partition(int a[], int n);
The function treats the first element of a [] as a pivot, and rearranges the array so that all elements less than or equal to the pivot is in the left part of the array, and all elements greater than the pivot is in the right part. In addition, it moves the pivot so that the pivot is the last element of the left part. The return value is the number of elements in the left part.
The following partially given function in the C programming language is used to find the K^{th} smallest element in an array a [] of size n using the partition function. We assume k ≤ n.

```
int kth_smallest(int a[], int n, int k)
{
   int left_end = partition(a,n);

   if ( left_end+1 == k ){
       return a[left_end];
   }

   if ( left_end+1 > k ){
       return kth_smallest( _____ );
   } else {
       return kth_smallest( _____ );
   }
}
```

The missing argument lists are respectively
A. (a, left_end, k) and (a+left_end+1, n-left_end-1, k-left_end-1)
B. (a, left_end, k) and (a, n-left_end-1, k-left_end-1)
C. (a+left_end+1, n-left_end-1, k-left_end-1) and (a, left_end, k)
D. (a, n-left_end-1, k-left_end-1) and (a, left_end, k)
Answer ||| A
Solution |||
This is an optimization over method 1 if Quick Sort is used as a sorting algorithm in first step. In Quick Sort, we pick a pivot element, and then move the pivot element to its correct position and partition the array around it. The idea is, not to do complete quick sort, but stop at the point where pivot itself is k'th smallest element. Also, not to recur for both left and right sides of pivot, but recur for one of them according to the position of pivot. The worst case time complexity of this method is $O(n^2)$, but it works in O(n) on average.

```
#include<iostream>
#include<climits>
using namespace std;
int partition(int arr[], int l, int r);
// This function returns k'th smallest element in arr[l..r] using
// QuickSort based method. ASSUMPTION: ALL ELEMENTS IN ARR[] ARE DISTINCT
int kthSmallest(int arr[], int l, int r, int k)
{
// If k is smaller than number of elements in array
if (k > 0 && k <= r - l + 1)
{
// Partition the array around last element and get
// position of pivot element in sorted array
int pos = partition(arr, l, r);
// If position is same as k
if (pos-l == k-1)
return arr[pos];
if (pos-l > k-1) // If position is more, recur for left sub array
return kthSmallest(arr, l, pos-1, k);
// Else recur for right sub array
return kthSmallest(arr, pos+1, r, k-pos+l-1);
}
// If k is more than number of elements in array
return INT_MAX;
}
void swap(int *a, int *b)
```

```
{
int temp = *a;
*a = *b;
*b = temp;
}
// Standard partition process of QuickSort(). It considers the last
// element as pivot and moves all smaller element to left of it
// and greater elements to right
int partition(int arr[], int l, int r)
{
int x = arr[r], i = l;
for (int j = l; j <= r - 1; j++)
{
if (arr[j] <= x)
{
swap(&arr[i], &arr[j]);
i++;
}
}
swap(&arr[i], &arr[r]);
return i;
}
// Driver program to test above methods
int main()
{
int arr[] = {12, 3, 5, 7, 4, 19, 26};
int n = size of(arr)/size of(arr[0]), k = 3;
cout << "K'th smallest element is " << kthSmallest(arr, 0, n-1, k);
return 0;
}
```

Output:
K'th smallest element is 5

40. Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?
A. $h(i) = i^2 \bmod 10$
B. $h(i) = i^3 \bmod 10$
C. $h(i) = (11 * i^2) \bmod 10$
D. $h(i) = (12 * i) \bmod 10$
Answer ||| B
Solution |||
Since mod 10 is used, the last digit matters. If you do cube all numbers from 0 to 9, you get following

Number	Cube	Last Digit in Cube
0	0	0
1	1	1
2	8	8
3	27	7
4	64	4
5	125	5
6	216	6
7	343	3
8	512	2
9	729	9

Therefore all numbers from 0 to 2020 are equally divided in 10 buckets. If we make a table for square, we don't get equal distribution. In the following table. 1, 4, 6 and 9 are repeated, so these buckets would have more entries and buckets 2, 3, 7 and 8 would be empty.

Number	Square	Last Digit in Cube
0	0	0
1	1	**1**
2	4	**4**
3	9	**9**
4	16	**6**
5	25	5
6	36	**6**
7	49	**9**
8	64	**4**
9	81	**1**

41. The secant method is used to find the root of an equation f(x) = 0. It is started from two distinct estimates x_a and x_b for the root. It is an iterative procedure involving linear interpolation to a root. The iteration stops if $f(x_b)$ is very small and then x_b is the solution. The procedure is given below. Observe that there is an expression which is missing and is marked by ?. Which is the suitable expression that is to be put in place of ? so that it follows all steps of the secant method?

Secant

```
Initialize: xa, xb, ε, N // ε = convergence indicator
                         // N = maximum no. of iterations
fb = f(xb)
i = 0
while (i < N and  |fb| >  ε) do
     i = i + 1            // update counter
     xt = ?               // missing expression for
                          // intermediate value
     xa = xb              // reset xa
     xb = xt              //  reset xb
     fb = f(xb)           // function value at new xb
   end while
   if  |fb| >  ε then     // loop is terminated with i=N
     write "Non-convergence"
   else
         write "return xb"
   end if
```

A. $x_b - (f_b - f(x_a))f_b / (x_b - x_a)$
B. $x_a - (f_a - f(x_a))f_a / (x_b - x_a)$
C. $x_b - (x_b - x_a)f_b / (f_b - f(x_a))$
D. $x_a - (x_b - x_a)\, f_a / (f_b - f(x_a))$
Answer ||| D
Solution |||
The secant method can be interpreted as a method in which the derivative is replaced by an approximation and is thus a Quasi-Newton method. If we compare Newton's method with the secant method, we see that Newton's method converges faster (order 2 against $\alpha \approx 1.6$). However, Newton's method requires the evaluation of both f and its derivative f'at every step, while the secant method only requires the evaluation of f. Therefore, the secant method may occasionally be faster in practice. For instance, if we assume that evaluating f takes as much time as evaluating its derivative and we neglect all other costs, we can do two steps of the secant method (decreasing the logarithm of the error by a factor $\alpha^2 \approx 2.6$) for the same cost as one step of Newton's method

(decreasing the logarithm of the error by a factor 2), so the secant method is faster. If however we consider parallel processing for the evaluation of the derivative, Newton's method proves its worth, being faster in time, though still spending more steps.

42. Consider the C program below.

```
#include <stdio.h> int *A, stkTop;
int stkFunc(int opcode, int val)
{
static int size=0, stkTop=0;
switch (opcode) {
case -1: size = val; break;
case 0: if (stkTop < size) A[stkTop++] = val;
break; default: if (stkTop) return A[--stkTop];
}
return -1;
}
int main()
{
int B[20]; A = B; stkTop = -1;
stkFunc (-1, 10);
stkFunc ( 0, 5);
stkFunc ( 0, 10);
printf ("%d\n", stkFunc(1, 0) + stkFunc(1, 0));
}
```

The value printed by the above program is _____.

A. 5
B. 10
C. 15
D. 20

Answer ||| C

Solution |||

The code in main, basically initializes a stack of size 10, then pushes 5, then pushes 10. Finally the printf statement prints sum of two pop operations which is 10 + 5 = 15.

```
stkFunc (-1, 10); // Initialize size as 10
stkFunc (0, 5); // push 5
stkFunc (0, 10); // push 10
// print sum of two pop
printf ("%d\n", stkFunc(1, 0) + stkFunc(1, 0));
```

43. Consider the sequence of machine instructions given below:

```
MUL R5, R0, R1
DIV R6, R2, R3
ADD R7, R5, R6
SUB R8, R7, R4
```

In the above sequence, R0 to R8 are general purpose registers. In the instructions shown, the first register stores the result of the operation performed on the second and the third registers. This sequence of instructions is to be executed in a pipelined instruction processor with the following 4 stages: (1) Instruction Fetch and Decode (IF), (2) Operand Fetch (OF), (3) Perform Operation (PO) and (4) Write back the result (WB). The IF, OF and WB stages take 1 clock cycle each for any instruction. The PO stage takes 1 clock cycle for ADD or SUB instruction, 3 clock cycles for MUL instruction and 5 clock cycles for DIV instruction. The pipelined processor uses operand forwarding from the PO stage to the OF stage. The number of clock cycles taken for the execution of the above sequence of instructions is_____.

A. 11
B. 12
C. 13
D. 14

Answer ||| C

Solution |||

```
1  2  3  4  5  6  7  8  9  10  11  12  13
 IF OF PO PO PO WB
    IF OF      PO PO PO PO PO  WB
       IF      OF           PO  WB
          IF      OF           PO  WB
```

44. Consider a processor with byte-addressable memory. Assume that all registers, including Program Counter (PC) and Program Status Word (PSW), are of size 2 bytes. A stack in the main memory is implemented from memory location $(0100)_{16}$ and it grows upward. The stack pointer (SP) points to the top element of the stack. The current value of SP is $(016E)_{16}$. The CALL instruction is of two words, the first word is the op-code and the second word is the starting address of the subroutine (one word = 2 bytes). The CALL instruction is implemented as follows:

- Store the current value of PC in the stack
- Store the value of PSW register in the stack
- Load the starting address of the subroutine in PC

The content of PC just before the fetch of a CALL instruction is $(5FA0)_{16}$. After execution of the CALL instruction, the value of the stack pointer is

A. $(016A)_{16}$
B. $(016C)_{16}$
C. $(0170)_{16}$
D. $(0172)_{16}$

Answer ||| D

Solution |||

The current value of SP is $(016E)_{16}$

The value of SP after following operations is asked in question

- Store the current value of PC in the stack.

This operation increments SP by 2 bytes as size of PC is given 2 bytes in question.

So becomes $(016E)_{16} + 2 = (0170)_{16}$

- Store the value of PSW register in the stack.

This operation also increments SP by 2 bytes as size of PSW is also given 2 bytes.

So becomes $(0170)_{16} + 2 = (0172)_{16}$

- Load the starting address of the subroutine in PC.

The Load operation doesn't change SP.

So new value of SP is $(016E)_{16}$

45. The number of min-terms after minimizing the following Boolean expression is_____.

[D' + AB' + A'C + AC'D + A'C'D]'

A. 1
B. 46
C. 56
D. 76

Answer ||| A

Solution |||
Given Boolean expression is:
[D' + AB' + A'C + AC'D + A'C'D]'
Step 1: [D' + AB' + A'C + C'D (A + A')]'
(taking C'D as common)
Step 2: [D' + AB' + A'C + C'D]'
(as, A + A' = 1)
: [D' + DC' + AB' + A'C]' (Rearrange)
Step 3: [D' + C' + AB' + A'C]'
(Rule of Duality, A + A'B = A + B)
: [D' + C' + CA' + AB']' (Rearrange)
Step 4: [D' + C' + A' + AB']'
(Rule of Duality)
: [D' + C' + A' + AB']' (Rearrange)
Step 5: [D' + C' + A' + B']'
(Rule of Duality)
:[(D' + C')'.(A' + B')']
(Demorgan's law, (A + B)'=(A'. B'))
:[(D''.C'').(A''.B'')] (Demorgan's law)
:[(D.C).(A.B)] (Idempotent law, A'' = A)
: ABCD
Hence only 1 minterm after minimization.

46. Let $f(x) = x^{-(1/3)}$ and A denote the area of the region bounded by f(x) and the X-axis, when x varies from −1 to 1. Which of the following statements is/are TRUE?
I) f is continuous in [−1, 1]
II) f is not bounded in [−1, 1]
III) A is nonzero and finite
A. II only
B. III only
C. II and III only
D. I, II and III
Answer ||| C
Solution |||
1 is false: function is not a Continuous function. As a change of 1 in x leads to ∞ change in f(x). For example when x is changed from -1 to 0. At x = 0, f(x) is ∞ and at x = 1, f(x) is finite. **2 is True:** f(x) is not a bounded function as it becomes ∞ at x = 0. **3 is true:** A denote the area of the region bounded by f(x) and the X-axis. This area is bounded, we can calculate it by doing integrating the function

47. Perform the following operations on the matrix
$\begin{bmatrix} 3 & 4 & 45 \\ 7 & 9 & 105 \\ 13 & 2 & 195 \end{bmatrix}$.
(i) Add the third row to the second row
(ii) Subtract the third column from the first column.
The determinant of the resultant matrix is _____.
A. 0 B. 1
C. 50 D. 100
Answer ||| A
Solution |||
The given matrix is singular matrix as its last column is a multiple of first column. If we multiply 15 with first column, we get the third column.
A square matrix that is not invertible is called singular or degenerate. A square matrix is singular if and only if its determinant is 0

48. The number of onto functions (surjective functions) fromset X ={1,2,3,4}to set Y ={a, b, c}is _____.
A. 36
B. 46
C. 6
D. 56
Answer ||| A
Solution |||
A function f from X to Y is called onto if for all 'y' in Y there is an 'x' in X such that f(x) = y. In onto functions, all elements in Y are used. Every Subjective or onto function sends two elements of {1, 2, 3, 4} to the same element of {a, b, c}. There are $^4C_2$ = 6 such pairs of elements. The pairs are {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}. For a given pair {i, j} ⊂ {1, 2, 3, 4}, there are 3! Subjective functions such that f(i) = f(j). Hence there are total 6 * 6 = 36 subjective functions.

49. Let X and Y denote the sets containing 2 and 20 distinct objects respectively and F denote the set of all possible functions defined from X to Y. Let *f* be randomly chosen from F. The probability of *f* being one-to-one is _____.
A. 0.95 B. 1
C. 0.93 B. 0.69
Answer ||| A
Solution |||
X has 2 elements Y has 20 elements Number of functions from X to Y is 20*20 [Every element can take any of the 20 values] Number of one to one functions from X to Y is 20*19 [Every element takes a different value] So probability of a function being one to one = (20*19) / (20*20) = 380 / 400 = 0.95

50. Consider the alphabet Σ = {0, 1}, the null/empty string λ and the sets of strings X_0, X_1, and X_2 generated by the corresponding non-terminals of a regular grammar. X_0, X_1, and X_2 are related as follows.
$X_0 = 1\ X_1$
$X_1 = 0\ X_1 + 1\ X_2$
$X_2 = 0\ X_1 + \{\lambda\}$
Which one of the following choices precisely represents the strings in X_0?
A. 10(0* + (10)*)1
B. 10(0* + (10)*)*1
C. 1(0 +10)*1
D. 10(0 + 10)*1 + 110(0 + 10)*1
Answer ||| C
Solution |||
The smallest possible string by given grammar is "11".
X0 = 1X1
= 11X2 [Replacing X1 with 1X2]
= 11 [Replacing X2 with λ]
The string "11" is only possible with option (C).

51. A graph is self-complementary if it is isomorphic to its complement. For all self-complementary graphs on *n* vertices, *n* is
A. A multiple of 4

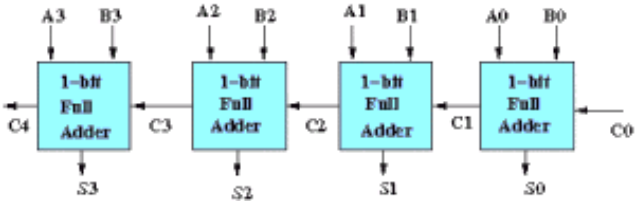
B. Even
C. Odd
D. Congruent to 0 *mod* 4, or, 1 *mod* 4.
Answer ||| D
Solution |||
An n-vertex self-complementary graph has exactly half number of edges of the complete graph, i.e., n(n − 1)/4 edges, and (if there is more than one vertex) it must have diameter either 2 or 3. Since n(n −1) must be divisible by 4, n must be congruent to 0 or 1 mod 4; for instance, a 6-vertex graph cannot be self-complementary.

52. In a connected graph, a bridge is an edge whose removal disconnects a graph. Which one of the following statements is true?
A. A tree has no bridges
B. A bridge cannot be part of a simple cycle
C. Every edge of a clique with size ≥ 3 is a bridge (A clique is any complete subgraph of a graph)
D. A graph with bridges cannot have a cycle
Answer ||| B
Solution |||
A bridge in a graph cannot be a part of cycle as removing it will not create a disconnected graph if there is a cycle.
A graph G is connected if there is a path in G between any given pair of vertices, otherwise it is disconnected. Every disconnected graph can be split up into a number of connected sub graphs, called components. Sub graph. Let G be a graph with vertex set V(G) and edge-list E(G).

53. Which one of the following well-formed formulae is a tautology?
A. $\forall x \exists y\, R(x,y) \leftrightarrow \exists y \forall x\, R(x,y)$
B. $(\forall x\, [\exists y\, R(x,y) \rightarrow S(x,y)]) \rightarrow \forall x \exists y\, S(x,y)$
C. $[\forall x \exists y\, (P(x,y) \rightarrow R(x,y)] \leftrightarrow [\forall x \exists y\, (\neg P(x,y) \vee R(x,y)]$
D. $\forall x \forall y\, P(x,y) \rightarrow \forall x \forall y\, P(y,x)$
Answer ||| C
Solution |||
Since p -> q = ¬q ∨ p, option C is tautology.
Tautologies are a key concept in propositional logic, where a tautology is defined as a propositional formula that is true under any possible Boolean valuation of its propositional variables. A key property of tautologies in propositional logic is that an effective method exists for testing whether a given formula is always satisfied (or, equivalently, whether its negation is unsatisfiable). The definition of tautology can be extended to sentences in predicate logic, which may contain quantifiers, unlike sentences of propositional logic. In propositional logic, there is no distinction between a tautology and a logically valid formula. In the context of predicate logic, many authors define a tautology to be a sentence that can be obtained by taking a tautology of propositional logic and uniformly replacing each propositional variable by a first-order formula (one formula per propositional variable). The set of such formulas is a proper subset of the set of logically valid sentences of predicate logic (which are the sentences that are true in every model).

54. Which one of the following assertions concerning code inspection and code walkthrough is true?
A. Code inspection is carried out once the code has been unit tested
B. Code inspection and code walkthrough are synonyms
C. Adherence to coding standards is checked during code inspection
D. Code walkthrough is usually carried out by an independent test team
Answer ||| C
Solution |||
A is False: It is not necessary that code inspection is carried out once the code has been unit tested. **B and D are false:** Code walkthrough or walk-through is a form of software peer review in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems" Inspection in software engineering, refers to peer review of any work product by trained individuals who look for defects using a well-defined process.

55. A half adder is implemented with XOR and AND gates. A full adder is implemented with two half adders and one OR gate. The propagation delay of an XOR gate is twice that of an AND/OR gate. The propagation delay of an AND/OR gate is 1.2 microseconds. A 4-bit ripple-carry binary adder is implemented by using four full adders. The total propagation time of this 4-bit binary adder in microseconds is_____.
A. 19.2 ms B. 20 ms
C. 15 ms D. 45 ms
Answer ||| A
Solution |||
A Ripple Carry Adder allows adding two n-bit numbers. It uses half and full adders. Following diagram shows a ripple adder using full adders.



Let us first calculate propagation delay of a single 1 bit full adder.
Propagation Delay by n bit full adder is (2n + 2) Gate delays.
[See this for formula].
Here n = 1, so total delay of a 1 bit full adder is (2 + 2)*1.2 = 4.8 ms
Delay of 4 full adders is = 4 * 4.8 = 19.2 ms