AP PGECET 2025 Computer Science Engineering Question Paper with Solutions

Time Allowed :2 Hour	Maximum Marks :120	Total Questions :120

General Instructions

Read the following instructions very carefully and strictly follow them:

1. The test is of 2 hour duration.
2. The question paper consists of 120 questions. The maximum marks are 120.
3. The questions are related to Computer Science Engineering.

1. In how many ways can a chairperson, a secretary, and a treasurer be chosen from a group of 10 people, if each person can hold only one position?

(A) 840
(B) 5040
(C) 1240
(D) 720

Correct Answer: (D) 720

Solution:

Step 1: Understanding the Question:
The problem asks for the number of ways to select and arrange 3 people for 3 distinct positions (chairperson, secretary, treasurer) from a group of 10 people. Since the positions are distinct, the order in which the people are chosen matters. For example, selecting person A as chairperson, B as secretary, and C as treasurer is different from selecting B as chairperson, A as secretary, and C as treasurer.

Step 2: Key Formula or Approach:
This is a permutation problem because the order of selection matters. The number of permutations of selecting r items from a set of n distinct items is given by the formula:

$$P(n, r) = \frac{n!}{(n-r)!}$$

In this problem, we are choosing 3 people for 3 positions from a group of 10. So, $n = 10$ and $r = 3$.

Step 3: Detailed Explanation:
We need to calculate the number of permutations of 10 people taken 3 at a time.

Using the permutation formula:

$$P(10,3) = \frac{10!}{(10-3)!} = \frac{10!}{7!}$$

Now, we expand the factorials:

$$P(10,3) = \frac{10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}$$

We can cancel out the 7! from the numerator and the denominator:

$$P(10,3) = 10 \times 9 \times 8$$

$$P(10,3) = 90 \times 8 = 720$$

Alternatively, we can think of filling the positions one by one:
- The position of chairperson can be filled by any of the 10 people.
- Once the chairperson is chosen, the position of secretary can be filled by any of the remaining 9 people.
- Once the chairperson and secretary are chosen, the position of treasurer can be filled by any of the remaining 8 people.

The total number of ways is the product of the number of choices for each position:

$$\text{Total ways} = 10 \times 9 \times 8 = 720$$

Step 4: Final Answer:
There are 720 ways to choose a chairperson, a secretary, and a treasurer from a group of 10 people.

> Quick Tip
>
> When a problem involves selecting items for distinct roles or positions, it's a permutation problem because the order matters. If the roles were identical (e.g., choosing a committee of 3), it would be a combination problem.

---

2. A sub graph H of a graph G is called a clique if _____.

(A) V(G) = V(H)
(B) $V(H) \subseteq V(G)$
(C) H contain all edges of G
(D) H is also a complete graph

Correct Answer: (A) V(G) = V(H)

Solution:

Step 1: Understanding the Question:
The question asks for the definition of a clique in the context of a subgraph H of a graph G. A clique is a concept in graph theory representing a subset of vertices where every two distinct vertices are adjacent.

Step 2: Key Formula or Approach:
The standard definition of a clique is a subset of vertices of an undirected graph G such that every two distinct vertices in the subset are adjacent. A subgraph H induced by a clique vertex set is a complete graph. Therefore, a subgraph H is a clique if H is a complete graph.

Step 3: Detailed Explanation:
Let's analyze the given options based on the standard definition:
- (A) V(G) = V(H): This condition defines H as a **spanning subgraph** of G. A spanning subgraph is not necessarily a clique. For H to be a clique, it must also be a complete graph. If a spanning subgraph H is a clique, it implies that the original graph G must also be a complete graph.
- (B) $V(H) \subseteq V(G)$: This is true for any subgraph H of G by definition, so it's not a distinguishing feature of a clique.
- (C) H contain all edges of G: This, combined with $V(H) \subseteq V(G)$, implies that H=G. The graph G itself may or may not be a clique.
- (D) H is also a complete graph: This is the correct standard definition of a subgraph H being a clique. In a complete graph, every pair of distinct vertices is connected by an edge.

Note on the provided answer key: The answer key marks option (A) as correct. This is non-standard and likely an error in the question paper or the key. Standard graph theory defines a subgraph H being a clique if "H is also a complete graph" (Option D). To justify the given answer (A), one might have to assume a highly specific and unusual context where the term "clique" is used to refer to a "complete spanning subgraph". In such a case, the condition V(G)=V(H) would define it as spanning, with the "complete" property being implicit. However, this is not a standard definition. Based on universal definitions, Option (D) is the correct choice.

Step 4: Final Answer:
According to the provided answer key, the correct option is (A). However, based on the standard mathematical definition of a clique, a subgraph H is a clique if it is a complete graph (Option D). The key is likely incorrect.

Quick Tip

Always rely on standard definitions for core concepts like cliques, paths, and cycles. A clique is a complete subgraph. A maximal clique is a clique that cannot be extended by adding another vertex. Be aware that exam keys can sometimes contain errors.

3. For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?

(A) v = e
(B) v = e + 1
(C) v + 1 = e
(D) v = e - 1

Correct Answer: (B) v = e + 1

Solution:

Step 1: Understanding the Question:
The question describes a graph G with specific properties: it is connected and has no cycles. We need to identify the correct relationship between the number of its vertices (v) and edges (e).

Step 2: Key Formula or Approach:
A connected graph with no cycles is the definition of a **tree**. A fundamental property of any tree is the relationship between its number of vertices and edges. For any tree with v vertices and e edges, the formula $v = e + 1$ always holds.

Step 3: Detailed Explanation:
Let's verify this property with an example.
- A tree with 1 vertex (a single node) has 0 edges. Here, $v = 1, e = 0$. So, $1 = 0 + 1$, which is true.
- A tree with 2 vertices must have 1 edge connecting them. Here, $v = 2, e = 1$. So, $2 = 1 + 1$, which is true.
- A tree with 3 vertices can be a path graph (V-V-V). It has 2 edges. Here, $v = 3, e = 2$. So, $3 = 2 + 1$, which is true.

This property can be proven by induction. The base case is a single vertex ($v = 1, e = 0$). The inductive step involves adding a new vertex and an edge to an existing tree, which preserves the property ($v \to v + 1, e \to e + 1$).
Given that G is connected and acyclic, it is a tree. Therefore, the relationship $v = e + 1$ must be true.

Step 4: Final Answer:
The correct statement for a connected graph with no cycles is $v = e + 1$.

Quick Tip

Memorize the key properties of trees: 1. A connected graph with no cycles. 2. For v vertices, there are always $v - 1$ edges. 3. There is exactly one unique simple path between any two vertices. 4. Adding any edge creates exactly one cycle. 5. Removing any edge disconnects the graph.

4. Assume G is a simple undirected graph and some vertices of G are of odd degree. Add a node v to G and make it adjacent to each odd degree vertex of G, then the resultant graph is sure to be _____.

(A) Euler
(B) complete
(C) Hamiltonian
(D) clique

Correct Answer: (A) Euler

Solution:

Step 1: Understanding the Question:
We start with a simple undirected graph G that has some vertices with an odd degree. We create a new graph by adding a new vertex 'v' and connecting it to every vertex in G that originally had an odd degree. The question asks for a property that the resulting graph is guaranteed to have.

Step 2: Key Formula or Approach:
The key concepts are vertex degree and Eulerian graphs.
1. **Handshaking Lemma:** In any graph, the number of vertices with odd degree is always even.
2. **Eulerian Graph:** A connected graph is called an Eulerian graph if it has an Eulerian circuit (a path that visits every edge exactly once and ends at the starting vertex). A connected graph has an Eulerian circuit if and only if every vertex in the graph has an even degree.

Step 3: Detailed Explanation:
Let the set of odd-degree vertices in G be V_{odd}. By the Handshaking Lemma, the number of vertices in V_{odd}, let's say $k = |V_{odd}|$, must be an even number.
We create a new graph G' by adding a new vertex 'v' and edges from 'v' to every vertex in V_{odd}. Let's analyze the degrees of vertices in G':
- **Vertices originally with even degree in G:** Their degrees remain unchanged in G', so they are still even.
- **Vertices originally with odd degree in G (vertices in V_{odd}):** Each of these vertices gets one new edge connected to 'v'. Their new degree will be (original odd degree) + 1, which is an even number.
- **The new vertex 'v':** Its degree is the number of edges connected to it, which is the number of vertices in V_{odd}. Since $|V_{odd}| = k$ is an even number, the degree of 'v' is even.
Thus, every vertex in the new graph G' has an even degree. If the original graph G was connected, the new graph G' will also be connected. A connected graph where all vertices have even degrees is an Eulerian graph. The option "Euler" refers to an Eulerian graph.

The resultant graph is not necessarily complete, Hamiltonian, or a clique.

Step 4: Final Answer:
The resultant graph is guaranteed to have all vertices of even degree, and thus will have an Euler circuit (assuming connectivity is maintained). Therefore, it is an Eulerian graph.

> Quick Tip
>
> Remember the core condition for Eulerian paths and circuits. An Eulerian circuit exists if and only if all vertices have an even degree (in a connected graph). An Eulerian path exists if and only if there are exactly two vertices of odd degree.

---

5. Dijkstra's algorithm fails for graphs with _____.

(A) Negative weights
(B) Disconnected components
(C) Directed edges
(D) Self-loops

Correct Answer: (A) Negative weights

Solution:

Step 1: Understanding the Question:
The question asks for the condition under which Dijkstra's algorithm for finding the shortest path between nodes in a graph does not work correctly.

Step 2: Key Formula or Approach:
Dijkstra's algorithm is a greedy algorithm. It works by maintaining a set of visited vertices and, at each step, it selects the unvisited vertex with the smallest known distance from the source and adds it to the visited set. It assumes that once a vertex is marked as "visited," the calculated shortest path to it is final and will not be improved upon later.

Step 3: Detailed Explanation:
Let's analyze why the presence of negative weights causes Dijkstra's algorithm to fail.
The greedy approach of Dijkstra's algorithm relies on the assumption that path lengths are always non-decreasing as we add more edges (since edge weights are non-negative). If there is a negative edge weight, this assumption is violated.
Consider a path from source S to vertex A with distance 5. Dijkstra might finalize this path. However, there could be another path S -> B -> A where the edge (B, A) has a weight of -4. If the path to B is, say, 7, the total path length to A via B would be $7 - 4 = 3$. If B is processed after A, Dijkstra's algorithm will have already finalized the

path to A with distance 5 and will not update it to the shorter path of 3. This leads to an incorrect result.
Let's review the other options:
- **Disconnected components:** Dijkstra's algorithm will correctly find the shortest paths to all reachable vertices from the source. It will not find paths to vertices in other components, which is the expected behavior.
- **Directed edges:** Dijkstra's algorithm works perfectly fine on directed graphs (digraphs) as long as the edge weights are non-negative.
- **Self-loops:** If a self-loop has a non-negative weight, it will never be part of a shortest path (as it would only increase the path length). If it has a weight of 0, it has no effect. Dijkstra's handles this correctly.

Step 4: Final Answer:
Dijkstra's algorithm fails to produce the correct shortest path in graphs containing negative weight edges. For such graphs, algorithms like Bellman-Ford or Floyd-Warshall should be used.

> Quick Tip
>
> For shortest path problems, remember the right algorithm for the job: - **Dijkstra:** Single source, non-negative edge weights. - **Bellman-Ford:** Single source, handles negative edge weights, can detect negative cycles. - **Floyd-Warshall:** All-pairs shortest path, handles negative edge weights.

---

6. In propositional logic $P \leftrightarrow Q$ is equivalent to ($\sim$ denotes NOT)_____.

(A) $\sim (P \vee Q) \wedge \sim (Q \vee P)$
(B) $(\sim P \vee Q) \wedge (\sim Q \vee P)$
(C) $(P \vee Q) \wedge (Q \vee P)$
(D) $\sim (P \vee Q) \rightarrow \sim (Q \vee P)$

Correct Answer: (B) $(\sim P \vee Q) \wedge (\sim Q \vee P)$

Solution:

Step 1: Understanding the Question:
The question asks for the logical expression that is equivalent to the biconditional statement $P \leftrightarrow Q$ (P if and only if Q).

Step 2: Key Formula or Approach:
We will use the definitions of the biconditional and conditional operators in propositional logic.
1. Biconditional Equivalence: $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$

2. Conditional Equivalence (Implication): $A \to B \equiv \neg A \lor B$

Step 3: Detailed Explanation:
Starting with the definition of the biconditional operator:

$$P \leftrightarrow Q \equiv (P \to Q) \land (Q \to P)$$

Now, we apply the conditional equivalence to each part of the expression.
For the first part, $P \to Q$, we replace it with its equivalent form:

$$P \to Q \equiv \neg P \lor Q$$

For the second part, $Q \to P$, we replace it with its equivalent form:

$$Q \to P \equiv \neg Q \lor P$$

Now, we substitute these back into the biconditional expression:

$$(P \to Q) \land (Q \to P) \equiv (\neg P \lor Q) \land (\neg Q \lor P)$$

Using the provided notation where $\sim$ denotes NOT, $\lor$ denotes OR, and $\land$ denotes AND, the expression is:

$$(\sim P \lor Q) \land (\sim Q \lor P)$$

This matches option (B).

Step 4: Final Answer:
The expression $P \leftrightarrow Q$ is equivalent to $(\sim P \lor Q) \land (\sim Q \lor P)$.

Quick Tip

It's essential to memorize the key logical equivalences: - $p \to q \equiv \neg p \lor q$ - $p \leftrightarrow q \equiv (p \to q) \land (q \to p) \equiv (\neg p \lor q) \land (\neg q \lor p)$ - De Morgan's Laws: $\neg(p \land q) \equiv \neg p \lor \neg q$ and $\neg(p \lor q) \equiv \neg p \land \neg q$

---

7. The Boolean function $\sim ((\sim P \land Q) \land \sim (\sim P \land \sim Q)) \lor (P \lor R)$ is equal to the Boolean function:

(A) Q
(B) R
(C) $P \lor Q$
(D) P
(E)

Correct Answer: (D) P

Solution:

Step 1: Understanding the Question:
We must simplify the Boolean expression
$\neg\Big((\neg P \wedge Q) \wedge \neg(\neg P \wedge \neg Q)\Big) \vee (P \vee R)$
to an equivalent, simpler Boolean function.

Step 2: Key Formula or Approach:
We will use De Morgan's laws and basic distributive/absorption identities:
$\neg(A \wedge B) = \neg A \vee \neg B$, $\neg(A \vee B) = \neg A \wedge \neg B$,
and $\neg(\neg X) = X$.

Step 3: Detailed Explanation:
Start with the inner negation:
$\neg(\neg P \wedge \neg Q) = P \vee Q$.
So the subexpression becomes
$(\neg P \wedge Q) \wedge (P \vee Q)$.
But $(\neg P \wedge Q)$ already implies Q is true, so $P \vee Q$ is true whenever $\neg P \wedge Q$ is true.
Hence $(\neg P \wedge Q) \wedge (P \vee Q)$ simplifies to $\neg P \wedge Q$.
Thus the whole expression reduces to
$\neg(\neg P \wedge Q) \vee (P \vee R)$.
Apply De Morgan: $\neg(\neg P \wedge Q) = P \vee \neg Q$.
So we get

$$(P \vee \neg Q) \vee (P \vee R).$$

Associativity and idempotence of $\vee$ gives

$$P \vee \neg Q \vee R.$$

So the simplified form (without any extra assumptions) is

$$\boxed{P \vee \neg Q \vee R}.$$

Step 4: Final Answer:
By direct Boolean simplification the expression is $P \vee \neg Q \vee R$.
However, the provided answer key states the correct option is (D) P.
To reconcile this with the key: for the expression to be **P** exactly, an additional assumption must hold (for example, that $\neg Q \vee R$ is logically implied by P, or that $\neg Q \vee R$ is always false under the context of the problem).
Without such an extra context/assumption the expression is not logically equivalent to P in general.

Quick Tip

Always simplify inner negations first using De Morgan's laws.
After simplification, check whether the result can be further reduced by absorption (e.g., $P \vee (P \wedge X) = P$).
Also, if an answer key disagrees with your algebra, look for implicit context or assumptions in the question (sometimes exam keys assume certain variables are fixed).

---

8. 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

Correct Answer: (D) congruent to 0 mod 4 or 1 mod 4

Solution:

Step 1: Understanding the Question:
The question asks for a necessary condition on the number of vertices, n, for a graph to be self-complementary. A self-complementary graph is a graph that is isomorphic to its own complement.

Step 2: Key Formula or Approach:
Let G be a graph with n vertices and m edges. Its complement, $\bar{G}$, will also have n vertices. The number of edges in $\bar{G}$ is the number of edges in a complete graph K_n minus the number of edges in G.
Number of edges in $K_n = \binom{n}{2} = \frac{n(n-1)}{2}$.
Number of edges in $\bar{G} = \frac{n(n-1)}{2} - m$.
If G is self-complementary, it must be isomorphic to $\bar{G}$. Isomorphic graphs must have the same number of vertices and edges. Therefore, the number of edges in G must equal the number of edges in $\bar{G}$.

$$m = \frac{n(n-1)}{2} - m$$

Step 3: Detailed Explanation:
From the equation derived in Step 2, we can solve for m:

$$2m = \frac{n(n-1)}{2}$$
$$m = \frac{n(n-1)}{4}$$

Since the number of edges, m, must be an integer, the term $n(n-1)$ must be divisible by 4.
Let's check the possibilities for n modulo 4:
- **Case 1: n is congruent to 0 mod 4.**
Let $n = 4k$ for some integer k. Then $n(n-1) = 4k(4k-1)$. This is clearly divisible by 4.
- **Case 2: n is congruent to 1 mod 4.**
Let $n = 4k+1$. Then $n(n-1) = (4k+1)(4k)$. This is also clearly divisible by 4.
- **Case 3: n is congruent to 2 mod 4.**
Let $n = 4k+2$. Then $n(n-1) = (4k+2)(4k+1) = 2(2k+1)(4k+1)$. Since $(2k+1)$ and $(4k+1)$ are both odd, their product is odd. So, the expression is 2 times an odd number, which is not divisible by 4.
- **Case 4: n is congruent to 3 mod 4.**
Let $n = 4k+3$. Then $n(n-1) = (4k+3)(4k+2) = (4k+3)2(2k+1)$. Again, this is 2 times the product of two odd numbers, which is not divisible by 4.
Therefore, for the number of edges to be an integer, n must be congruent to 0 mod 4 or 1 mod 4.

Step 4: Final Answer:
The number of vertices n in a self-complementary graph must be congruent to 0 mod 4 or 1 mod 4.

Quick Tip

For questions about graph properties like being self-complementary, start with the most basic invariants like the number of vertices and edges. The relationship between the edges of a graph and its complement is a powerful tool for solving such problems.

---

9. Suppose R_1 and R_2 are reflexive relations on a set A. Which of the following statements is correct?

(A) $R_1 \cap R_2$ is reflexive and $R_1 \cup R_2$ is irreflexive
(B) $R_1 \cap R_2$ is irreflexive and $R_1 \cup R_2$ is reflexive
(C) Both $R_1 \cap R_2$ and $R_1 \cup R_2$ are irreflexive
(D) Both $R_1 \cap R_2$ and $R_1 \cup R_2$ are reflexive

Correct Answer: (D) Both $R_1 \cap R_2$ and $R_1 \cup R_2$ are reflexive

Solution:

Step 1: Understanding the Question:
The question asks about the properties of the intersection ($\cap$) and union ($\cup$) of two

reflexive relations, R_1 and R_2, defined on a set A.

Step 2: Key Formula or Approach:
The definition of a reflexive relation is key here. A relation R on a set A is **reflexive** if for every element $a \in A$, the pair (a, a) is in R. This can be written as: $\forall a \in A, (a, a) \in R$. The identity relation on A is $I_A = \{(a, a) | a \in A\}$. A relation R is reflexive if and only if $I_A \subseteq R$.

Step 3: Detailed Explanation:
We are given that R_1 and R_2 are reflexive relations on set A. This means:
1. For all $a \in A$, $(a, a) \in R_1$.
2. For all $a \in A$, $(a, a) \in R_2$.

Checking the Intersection ($R_1 \cap R_2$):
A pair (x, y) is in $R_1 \cap R_2$ if and only if $(x, y) \in R_1$ AND $(x, y) \in R_2$.
To check if $R_1 \cap R_2$ is reflexive, we need to see if $(a, a) \in R_1 \cap R_2$ for all $a \in A$.
Since R_1 is reflexive, we know $(a, a) \in R_1$.
Since R_2 is reflexive, we know $(a, a) \in R_2$.
Because (a, a) is in both R_1 and R_2, by the definition of intersection, $(a, a) \in R_1 \cap R_2$.
This holds for all $a \in A$, so $R_1 \cap R_2$ is reflexive.

Checking the Union ($R_1 \cup R_2$):
A pair (x, y) is in $R_1 \cup R_2$ if and only if $(x, y) \in R_1$ OR $(x, y) \in R_2$.
To check if $R_1 \cup R_2$ is reflexive, we need to see if $(a, a) \in R_1 \cup R_2$ for all $a \in A$.
Since R_1 is reflexive, we know $(a, a) \in R_1$.
By the definition of union, if an element is in R_1, it must also be in $R_1 \cup R_2$.
Therefore, $(a, a) \in R_1 \cup R_2$ for all $a \in A$.
This means $R_1 \cup R_2$ is also reflexive.

Step 4: Final Answer:
Both the intersection and the union of two reflexive relations are reflexive.

> Quick Tip
>
> When checking properties of combined relations (union, intersection), always go back to the fundamental definition of the property (reflexive, symmetric, transitive). For reflexivity, you only need to check that all pairs (a, a) are present.

---

10. There are three cards in a box, Both sides of one card are black, both sides of one card are red, and the third card has one black side and one red side. We pick a card at random and observe only one side. What is the probability that the opposite side is the same colour as the one side we observed?

(A) 3/4
(B) 2/3
(C) 1/2
(D) 1/3

Correct Answer: (B) 2/3

Solution:

Step 1: Understanding the Question:
We have three cards: Black-Black (BB), Red-Red (RR), and Black-Red (BR). We randomly pick one card and look at one of its sides. We want to find the probability that the other side of this card is the same color as the side we are looking at.

Step 2: Key Formula or Approach:
This is a conditional probability problem. Let's use a sample space approach for clarity. There are 3 cards, and each has 2 sides. We can think of the total number of faces we could possibly observe as our sample space.
Total faces: 2 from BB card, 2 from RR card, 2 from BR card. Total = 6 faces. Let's name the faces: - BB card: B1, B2 - RR card: R1, R2 - BR card: B3, R3 Each of these 6 faces has an equal chance (1/6) of being the one we observe.

Step 3: Detailed Explanation:
Let's condition on the color of the side we observe.
Case 1: The side we observe is Black. The black faces we could have observed are B1, B2, or B3. So there are 3 equally likely possibilities for the observed face. - If we observed face B1 (from the BB card), the other side is B2 (same color). - If we observed face B2 (from the BB card), the other side is B1 (same color). - If we observed face B3 (from the BR card), the other side is R3 (different color). Out of the 3 scenarios where we see a black face, 2 of them result in the other face being the same color (black). So, the probability is 2/3.

Case 2: The side we observe is Red. The red faces we could have observed are R1, R2, or R3. There are 3 equally likely possibilities. - If we observed face R1 (from the RR card), the other side is R2 (same color). - If we observed face R2 (from the RR card), the other side is R1 (same color). - If we observed face R3 (from the BR card), the other side is B3 (different color). Out of the 3 scenarios where we see a red face, 2 of them result in the other face being the same color (red). So, the probability is 2/3.

Since the probability is 2/3 regardless of whether we see black or red, the overall probability is 2/3.

Alternative Method (Bayes' Theorem): Let S be the event that the opposite side is the same color. This is equivalent to picking the BB or RR card. Let O_B be the event that we observe a black side. We want to find $P(S|O_B)$.

$P(S|O_B) = P(\text{picked BB card}|O_B)$

Using Bayes' theorem: $P(BB|O_B) = \frac{P(O_B|BB) \times P(BB)}{P(O_B)}$

- $P(BB) = 1/3$ (Probability of picking the BB card).
- $P(O_B|BB) = 1$ (If we have the BB card, we will observe a black side).

- $P(O_B) = P(O_B|BB)P(BB) + P(O_B|RR)P(RR) + P(O_B|BR)P(BR)$

$P(O_B) = (1 \times 1/3) + (0 \times 1/3) + (1/2 \times 1/3) = 1/3 + 1/6 = 1/2.$

Substituting these values: $P(BB|O_B) = \frac{1 \times (1/3)}{1/2} = \frac{1/3}{1/2} = 2/3.$

Step 4: Final Answer:
The probability that the opposite side is the same colour as the one side we observed is 2/3.

Quick Tip

This is a classic probability puzzle similar to the Bertrand's Box Paradox. The common mistake is to think the answer is 1/2. The key is to correctly define the sample space. Instead of thinking about the cards, think about the faces. There are three black faces and three red faces in total. Two of the three black faces have a black face on the other side. Two of the three red faces have a red face on the other side.

---

11. An 8-bit number X is 00001100, then -X in 1's complement is _____.

(A) 12 in binary
(B) 243 binary code in 8 bit
(C) 111011100
(D) 11110101

Correct Answer: (B) 243 binary code in 8 bit

Solution:

Step 1: Understanding the Question:
We must find the 1's complement representation of the negative of an 8-bit binary number X. Given $X = 00001100$.

Step 2: Key Formula or Approach:
The 1's complement of a binary number is obtained by inverting all bits ($0 \leftrightarrow 1$). This new bit pattern represents the negation of the original number in the 1's complement

system.

Step 3: Detailed Explanation:
The given number is:

$$X = 00001100$$

Flip each bit:

$$-X = 11110011$$

Now check options:
(A) 12 in binary = 1100, incorrect.
(B) 243 in 8-bit binary = 11110011_2, matches perfectly.
(C) 111011100 is 9-bit, invalid.
(D) 11110101 is incorrect.

Step 4: Final Answer:
The 1's complement of 00001100 is 11110011, which equals 243 in unsigned decimal form.

> Quick Tip
>
> To find the 1's complement, invert all bits. To find the 2's complement (commonly used for signed numbers), invert all bits and add 1.

---

12. Assuming 5-bit addition is done on registers, the sum 12+7 leads to _____.

(A) Overflow
(B) -12
(C) 19
(D) -14

Correct Answer: (A) Overflow

Solution:

Step 1: Understanding the Question:
We add 12 and 7 in 5-bit signed representation. The range for 5-bit signed numbers is from -16 to $+15$.

Step 2: Key Formula or Approach:
Range for signed n-bit number: -2^{n-1} to $2^{n-1} - 1$. For $n = 5$: -16 to $+15$.

Step 3: Detailed Explanation:

$$12 + 7 = 19$$

Since $19 > 15$, overflow occurs.
Binary demonstration:

$$\begin{array}{r} 01100 \\ +00111 \\ \hline 10011 \end{array}$$

Here MSB = 1 indicates a negative number, even though both operands were positive. Thus overflow occurred.

Step 4: Final Answer:
Overflow occurs because 19 exceeds the range of representable 5-bit signed numbers.

Quick Tip

Overflow rules: 1. Two positives → negative result → overflow.
2. Two negatives → positive result → overflow.
3. One positive, one negative → no overflow.

---

13. In a 4-bit ripple carry adder, the worst-case delay occurs when _____.

(A) all input bits are 0
(B) a carry propagates through all full adders
(C) no carry is generated
(D) the sum exceeds 15

Correct Answer: (B) a carry propagates through all full adders

Solution:

Step 1: Understanding the Question:
The longest delay in a ripple-carry adder occurs when the carry must propagate through every stage.

Step 2: Key Formula or Approach:
Delay $\approx n \times t_{carry}$, where n = number of bits.

Step 3: Detailed Explanation:
For 4-bit adder, worst case: each stage must wait for previous carry. Example: adding 1111 and 0001. The carry ripples from LSB to MSB.

Step 4: Final Answer:
Worst-case delay occurs when a carry propagates through all full adders.

> Quick Tip
>
> Ripple-carry adders are slow for wide operands. Carry-lookahead adders overcome this by computing carries in parallel.

---

14. The output of a JK flip-flop for inputs J=1, K=1 and clock transition from 1 to 0 is _____.

(A) No change
(B) Reset
(C) Toggle
(D) Set

Correct Answer: (C) Toggle

Solution:

Step 1: Understanding the Question:
JK flip-flop behavior for inputs J=1, K=1 at the active clock edge ($1 \rightarrow 0$).

Step 2: Key Formula or Approach:
JK truth table:

J	K	Action
0	0	Hold
0	1	Reset (Q=0)
1	0	Set (Q=1)
1	1	Toggle ($Q_{next} = \neg Q$)

Step 3: Detailed Explanation:
J=1, K=1 causes the flip-flop to toggle at the active (falling) clock edge.

Step 4: Final Answer:
Output toggles.

> Quick Tip
>
> JK flip-flops are versatile; J=K=1 causes toggling and avoids race-around problems in edge-triggered designs.

---

15. The IEEE 754 single-precision floating-point representation of 0.375 is _____.

(A) 0 01111100 10000000000000000000000
(B) 0 01111101 10000000000000000000000
(C) 0 01111101 11000000000000000000000
(D) 0 01111100 11000000000000000000000

Correct Answer: (B) 0 01111101 10000000000000000000000

Solution:

Step 1: Understanding IEEE 754:
32-bit single precision: 1 sign, 8 exponent (bias 127), 23 fraction bits.

Step 2: Derivation:
$0.375_{10} = 0.011_2 = 1.1 \times 2^{-2}$.
Sign = 0, exponent = 125 (01111101), mantissa = 10000000000000000000000.

Step 3: Assemble:
0 01111101 10000000000000000000000.

Step 4: Final Answer:
Option (B) is correct; 0.375 → IEEE 754 representation: 0 01111101 10000000000000000000000.

> Quick Tip
>
> IEEE 754 single precision: Value $= (-1)^S \times (1.M) \times 2^{(E-127)}$

---

16. In a hardwired control unit, the sequence of control signals is generated by _____.

(A) Microprogrammed ROM
(B) State machine/circuitry
(C) Operating system
(D) Interrupt handler

Correct Answer: (B) State machine/circuitry

Solution:

Explanation:
Hardwired control units use fixed combinational and sequential circuits (state machines)

to generate control signals directly. Microprogrammed control units, in contrast, use microinstructions stored in ROM.

Final Answer:
State machine/circuitry.

> Quick Tip
>
> Hardwired = fast but inflexible. Microprogrammed = slower but flexible.

---

17. In DMA mode, the CPU is bypassed for data transfer between _____.

(A) CPU and I/O devices
(B) Registers and ALU
(C) Memory and I/O devices
(D) Cache and secondary storage

Correct Answer: (C) Memory and I/O devices

Solution:
DMA allows direct data transfer between memory and I/O devices without CPU intervention. CPU only initiates and finalizes the transfer.

Final Answer: Memory and I/O devices.

> Quick Tip
>
> DMA = Direct Memory Access → High-speed block transfers without CPU load.

---

18. A 4-way set-associative cache with 256 blocks divides memory addresses into _____.

(A) 256 sets
(B) 64 sets
(C) 4 sets
(D) 16 sets

Correct Answer: (B) 64 sets

Solution:
Number of sets = Total blocks / Associativity = 256 / 4 = 64 sets.

Final Answer: 64 sets.

> Quick Tip
>
> Formula: Sets = Blocks / Associativity.

---

19. $AB \oplus (A + B)$ **is equivalent to _____.**

(A) $A'B + AB'$
(B) $AB + A'B'$
(C) $A'B + A'B$
(D) $A + B$

Correct Answer: (A) $A'B + AB'$

Solution:

$$F = (AB) \oplus (A + B) = (AB)'(A + B) + (AB)(A + B)'$$

Simplify using De Morgan's law: $(AB)' = A' + B'$ and $(A + B)' = A'B'$.

$$F = (A' + B')(A + B) + AB(A'B')$$

Simplify first term: $A'B + AB' = A'B + AB'$. Second term = 0.

$$F = A'B + AB'$$

Final Answer: $A'B + AB'$.

> Quick Tip
>
> Identity: $(A \wedge B) \oplus (A \vee B) = A \oplus B$

---

20. Which of these flip-flops cannot be used to construct a serial shift register?

(A) D flip-flop
(B) SR flip-flop
(C) T flip-flop
(D) JK flip-flop

Correct Answer: (C) T flip-flop

Solution:
D, SR, and JK flip-flops can be configured to pass data serially (next state = previous output). T flip-flop only toggles; cannot directly load serial data, hence unsuitable.

Final Answer: T flip-flop.

> Quick Tip
>
> Shift registers need flip-flops that can "store input value." T flip-flops toggle instead, so they can't shift serial data.

---

21. How many AND gates are required to realize $Y = CD + EF + G$?

(A) 4
(B) 5
(C) 3
(D) 2

Correct Answer: (D) 2

Solution:

Step 1: Understanding the Question:
We need to count the number of AND gates required to implement the given Boolean expression in sum-of-products form.

Step 2: Key Formula or Approach:
In Boolean algebra, juxtaposition (e.g., CD) means AND, and the plus sign (+) means OR. Each product term with two or more variables requires one AND gate.

Step 3: Detailed Explanation:
The expression is:

$$Y = CD + EF + G$$

Breaking down the terms:

- $CD \rightarrow$ requires one 2-input AND gate.
- $EF \rightarrow$ requires one 2-input AND gate.
- $G \rightarrow$ single variable, no AND gate required.

Hence, two AND gates are required. The OR operation combines all terms using a 3-input OR gate.

Step 4: Final Answer:
Number of AND gates required = 2.

> Quick Tip
>
> Each product term with two or more literals needs one AND gate. Single-variable terms connect directly to the OR gate.

---

22. The purpose of an interrupt vector table is to _____.

(A) Store CPU registers during context switch
(B) Map interrupt requests to service routine addresses
(C) Hold page tables for virtual memory
(D) Cache frequently used instructions

Correct Answer: (B) Map interrupt requests to service routine addresses

Solution:

Step 1: Understanding the Question:
An interrupt vector table (IVT) helps the CPU determine which service routine to execute when an interrupt occurs.

Step 2: Key Formula or Approach:
Each interrupt has an associated unique interrupt number. The IVT stores the starting addresses of the corresponding interrupt service routines (ISRs).

Step 3: Detailed Explanation:
When an interrupt occurs:

1. The hardware generates an interrupt request with an interrupt number.
2. The CPU uses that number as an index into the IVT.
3. The IVT entry contains the address of the ISR.
4. The CPU jumps to that address and executes the ISR.

Thus, the IVT acts as a mapping between interrupt numbers and ISR addresses.

Step 4: Final Answer:
The IVT maps interrupt requests to their service routine addresses.

Quick Tip

The IVT is like a "directory" of ISRs — it tells the CPU where to find the right code for each interrupt.

---

23. What is the output of the following code fragment?

```
int x = 24;
printf("%d\n", printf("%3d%3d\n", x, x));
```

(A)

```
 24 24
10
```

(B)

```
 24 24
9
```

(C)

```
 24 24
8
```

(D) Compilation Error

Correct Answer: (C)

```
 24 24
8
```

Solution:

Step 1: Understanding the Question:
We must evaluate nested `printf` calls in C. Remember, the inner `printf` executes first, and `printf` returns the number of characters printed.

Step 2: Detailed Explanation:
1. Inner `printf`: `printf("%3d%3d\n", x, x);`
Each %3d prints 24 right-aligned in 3 spaces ⇒ " 24 24\n".
Characters printed = 3 + 3 + 1 (for newline) = 7.
So the inner `printf` returns 7.
2. Outer `printf`: becomes `printf("%d\n", 7);` which prints "7\n".
The total output is:

```
 24 24
7
```

The number of characters printed by the second ‘printf‘ is 2 (‘7‘ and “). This is a common tricky question; the return value of the *outer* printf is what some might consider. Let’s re-read the question carefully. It asks for the output of the code. The final output printed to the console is " 24 24" followed by a newline, and then "7" followed by a newline. The question key says "8", which implies the inner ‘printf‘ returns 8. Let’s count again: ‘$_2 4_2 4$‘.$Space = 1, 24 = 2, Space = 1, 24 = 2, newline = 1. Total = 7. The answer key (C) indicates” 8”, this happens if the output format is slightly different, e.g. ‘”%3d%$

Step 4: Final Answer:

The code as written prints " 24 24" and then "7" on the next line. The provided answer key indicating 8 is likely based on a slightly different format string.

> Quick Tip
>
> In C, `printf()` returns the number of characters successfully printed. Be careful to count spaces and escape sequences like `\n` correctly.

---

24. How many times will the message "Hi" be displayed when the following code is executed?

```
int i;
for(i=0; i%2?0:1; i++)
    printf("Hi\n");
```

(A) 3
(B) 2
(C) 1
(D) 0

Correct Answer: (C) 1

Solution:

Step 1: Understanding the Question:

The loop condition is the ternary operator expression `i%2 ? 0 : 1`. It returns 1 when `i` is even, 0 when `i` is odd.

Step 2: Execution Trace:

- Iteration 1: i = 0, i%2 = 0 $\Rightarrow$ condition = 1 (true) $\rightarrow$ prints "Hi".
- Increment i = 1.

- Iteration 2: i = 1, i%2 = 1 ⇒ condition = 0 (false) → loop ends.

Step 3: Final Answer:
"Hi" is printed once.

> Quick Tip
>
> In C, 0 is false and any nonzero value is true. Always test ternary logic with both even and odd cases.

---

25. If one wants to add and delete elements quickly without reshuffling the rest, which data structure suits best?

(A) Linked list
(B) Tree
(C) B-Tree
(D) Array

Correct Answer: (A) Linked list

Solution:

Explanation:
In arrays, insertion/deletion requires shifting elements, which is $O(n)$.
In linked lists, adding or removing a node involves only pointer changes ($O(1)$ if position known).
Therefore, linked lists allow quick modifications without reshuffling.

Final Answer: Linked list.

> Quick Tip
>
> Arrays → fast random access; slow insertion/deletion.
> Linked lists → fast insertion/deletion; slower access by index.

---

26. The height of a binary tree with 31 nodes (no single-child nodes) is _____.

(A) 4
(B) 5
(C) 6

(D) 7

Correct Answer: (A) 4

Solution:
For a perfect binary tree, number of nodes $n = 2^{h+1} - 1$.
Given $n = 31$, solve:

$$31 = 2^{h+1} - 1 \implies 2^{h+1} = 32 \implies h + 1 = 5 \implies h = 4.$$

Thus, height = 4.

Final Answer: Height = 4.

> Quick Tip
>
> Perfect tree node formula: $n = 2^{h+1} - 1$. For 31 nodes, h = 4.

---

27. In a max-heap, the largest element is always located at the _____.

(A) Leftmost leaf
(B) Rightmost leaf
(C) Root node
(D) Middle of the heap

Correct Answer: (C) Root node

Solution:
By definition, a max-heap maintains the property: every parent $\geq$ its children. Repeated application of this rule ensures the maximum value is at the root.

Final Answer: Root node.

> Quick Tip
>
> Max-heap $\rightarrow$ largest at root.
> Min-heap $\rightarrow$ smallest at root.

---

28. The recurrence $T(n) = 2T(n/2) + n$ has a solution of _____.

(A) $O(n)$
(B) $O(n \log n)$
(C) $O(n^2)$
(D) $O(\log n)$

Correct Answer: (B) $O(n \log n)$

Solution:
Using Master Theorem: $a = 2, b = 2, f(n) = n$.
Compute $n^{\log_b a} = n^{\log_2 2} = n$.
Since $f(n) = \Theta(n^{\log_b a})$, Case 2 applies:

$$T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n \log n).$$

Final Answer: $O(n \log n)$.

> Quick Tip
>
> Recurrence $T(n) = aT(n/b) + f(n) \rightarrow$ use Master Theorem to quickly get asymptotic behavior.

---

29. What is the time complexity to insert an element at the beginning of a dynamic array?

(A) $O(1)$
(B) $O(n)$
(C) $O(n \log n)$
(D) $O(\log n)$

Correct Answer: (B) $O(n)$

Solution:
In a dynamic array, inserting at the front requires shifting all existing n elements one position right.
Thus, the time complexity = $O(n)$.

Final Answer: $O(n)$.

> Quick Tip
>
> Insert at end $\rightarrow O(1)$ (amortized).
> Insert at front $\rightarrow O(n)$ due to element shifts.

30. The inorder traversal of a BST gives elements in _____.

(A) Level order
(B) Ascending order
(C) Descending order
(D) Random order

Correct Answer: (B) Ascending order

Solution:
Inorder traversal = Left → Root → Right.
For BST: left subtree < root < right subtree.
Thus, visiting nodes in this order yields ascending values.

Final Answer: Ascending order.

> Quick Tip
>
> BST traversal orders: Inorder → ascending, Preorder → root-first, Postorder → root-last.

---

31. A threaded binary tree is a binary tree in which every node that does not have right child has a thread to its _____.

(A) Pre-order successor
(B) In-order successor
(C) In-order predecessor
(D) Post-order successor

Correct Answer: (B) In-order successor

Solution:

Step 1: Understanding the Question:
This is a definition-based question about threaded binary trees. It asks where the "thread" of a node with a NULL right child pointer points.

Step 2: Key Formula or Approach:
A threaded binary tree is a modification of a normal binary tree designed to make traversals, particularly inorder traversal, faster and more memory-efficient (by avoiding recursion or a stack). In a standard threaded binary tree, NULL pointers are utilized to

store useful information.

Step 3: Detailed Explanation:
The standard convention for threading is as follows:
- If a node's **right child pointer** is NULL, it is replaced with a special pointer, called a "thread," that points to the node's **in-order successor** (the next node that would be visited in an inorder traversal).
- If a node's **left child pointer** is NULL, it is replaced with a thread that points to the node's **in-order predecessor**.
The question specifically asks about a node that does not have a right child. According to the definition, its thread points to its in-order successor.

Step 4: Final Answer:
The thread of a node with no right child points to its in-order successor.

Quick Tip

Threading is all about making inorder traversal easier. The right thread points "forward" to the in-order successor, and the left thread points "backward" to the in-order predecessor.

---

32. Which notation represents the tightest upper bound of an algorithm's running time?

(A) $O(n)$
(B) $\Omega(n)$
(C) $\Theta(n)$
(D) $\omega(n)$

Correct Answer: (C) $\Theta(n)$

Solution:

Step 1: Understanding the Question:
The question asks to identify the asymptotic notation that represents a "tightest bound."

Step 2: Key Formula or Approach:
Definitions:
- O **(Big O):** Asymptotic upper bound.
- Ω **(Big Omega):** Asymptotic lower bound.
- Θ **(Big Theta):** Asymptotic tight bound (both upper and lower).
- ω **(Little Omega):** Lower bound that is not tight.

Step 3: Detailed Explanation:
The tightest bound is the one that captures both upper and lower limits simultaneously, which is $\Theta(n)$. It accurately describes how the runtime grows asymptotically.

Step 4: Final Answer:
The Θ notation represents the tightest bound on an algorithm's running time.

> Quick Tip
>
> Think of Θ as "exact growth rate." O = at most, Ω = at least, Θ = exactly.

---

33. The minimum possible time complexity to sort n integers in the range [1,n^2] is _____.

(A) $O(n)$
(B) $O(n \log n)$
(C) $O(n^2)$
(D) $O(\log n)$

Correct Answer: (A) $O(n)$

Solution:

We can use **Radix Sort** with base n for numbers in the range $[1, n^2]$. Each number has at most 2 digits in base n, so $d = 2$ and $k = n$. Thus:

$$O(d(n + k)) = O(2(n + n)) = O(n)$$

Hence, the minimum possible time complexity is linear.

> Quick Tip
>
> When the range of numbers is polynomial in n, non-comparison sorts (Counting or Radix) achieve linear time.

---

34. The greedy approach is optimal for _____.

(A) 0/1 Knapsack
(B) Fractional Knapsack
(C) Longest Increasing Subsequence

(D) Matrix Chain Multiplication

Correct Answer: (B) Fractional Knapsack

Solution:

Greedy choice works for the Fractional Knapsack because we can take fractions of items, ensuring optimal use of remaining capacity. In 0/1 Knapsack, items are indivisible, so greedy fails — DP is required.

> Quick Tip
>
> Fractional → Greedy works. 0/1 → Use Dynamic Programming.

---

35. Best case complexity of insertion sorting is _____.

(A) $O(n)$
(B) $O(n \log n)$
(C) $O(n^2)$
(D) $O(\log n)$

Correct Answer: (A) $O(n)$

Solution:

In the best case (already sorted array), no element needs to be shifted. The inner loop runs once per iteration, giving linear time:

$$T(n) = O(n)$$

> Quick Tip
>
> Insertion sort is adaptive: Best → $O(n)$, Average/Worst → $O(n^2)$.

---

36. A problem is NP-complete if it is _____.

(A) Solvable in polynomial time
(B) In NP and is NP-hard

(C) Verifiable in polynomial time
(D) NP-hard

Correct Answer: (B) In NP and is NP-hard

Solution:

NP-complete problems satisfy both: 1. They are in NP (solutions verifiable in polynomial time). 2. They are NP-hard (every NP problem reduces to them).

> Quick Tip
>
> NP-complete = hardest problems in NP = (NP NP-hard).

---

37. The time complexity to compute the 15th Fibonacci number using dynamic programming is _____.

(A) $O(1)$
(B) $O(n)$
(C) $O(n \log n)$
(D) $O(\log n)$

Correct Answer: (B) $O(n)$

Solution:

In bottom-up DP, we compute Fibonacci numbers iteratively:

$$fib[i] = fib[i-1] + fib[i-2]$$

Each iteration is constant time, repeated n times, so total complexity $O(n)$.

> Quick Tip
>
> Naive recursion $\rightarrow O(2^n)$; Dynamic Programming $\rightarrow O(n)$.

---

38. One way to build a heap is to start at the end of the array (the leaves) and push each new value up to the root. Its time complexity is _____.

(A) $O(n)$
(B) $O(\log n)$

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

Correct Answer: (C) $O(n \log n)$

Solution:

This method inserts each element one by one into the heap. Each insertion takes $O(\log n)$ time, repeated n times.

$$T(n) = n \times O(\log n) = O(n \log n)$$

Quick Tip

Repeated insertions $\rightarrow O(n \log n)$, Heapify (bottom-up) $\rightarrow O(n)$.

---

39. If the recursive call keeps calculating the same things repeatedly, we can use _____ which stores partial results already calculated and to be used again.

(A) Divide and conquer algorithm
(B) Recursive
(C) Dynamic Programming
(D) Greedy method

Correct Answer: (C) Dynamic Programming

Solution:

Dynamic Programming avoids redundant computation by storing intermediate results (memoization/tabulation) and reusing them.

Quick Tip

DP = recursion + memory. Divide & Conquer solves disjoint subproblems; DP solves overlapping ones.

---

40. The _____ process updates the costs of all vertices V connected to a vertex U, if we could improve the best estimate of the shortest path to

V **by including** (U, V) **in the path to** V**.**

(A) Relaxation
(B) Improvement
(C) Shortening
(D) Costing

Correct Answer: (A) Relaxation

Solution:

In Dijkstra's and Bellman-Ford algorithms, relaxation means checking whether

$$dist(U) + w(U, V) < dist(V)$$

If yes, we update $dist(V) = dist(U) + w(U, V)$.

> Quick Tip
>
> Relaxation is the heart of shortest path algorithms — it "loosens" distance estimates toward the correct value.

---

41. The minimum number of states in a DFA accepting $L = \{w \mid w$ **ends with** $01\}$ **over** $\Sigma = \{0, 1\}$ **is:**

(A) 2
(B) 3
(C) 4
(D) 5

Correct Answer: (B) 3

Solution:

Step 1: Understanding the Question:
Design a minimal DFA that accepts all binary strings ending with the substring "01".

Step 2: Key Formula or Approach:
Keep track of how much of the pattern "01" has been matched so far.

Step 3: Detailed Explanation:
Use three states:
- q_0: start state — nothing useful matched yet (also corresponds to "not ended with 0").
- q_1: last symbol seen is '0' (potential start of "01").

- q_2: last two symbols are "01" (accepting).
Transitions: from q_0, $'0' \to q_1$, $'1' \to q_0$; from q_1, $'1' \to q_2$, $'0' \to q_1$; from q_2, $'0' \to q_1$, $'1' \to q_0$.
All three states are necessary to distinguish remainders of the pattern, so minimal number $= 3$.

Step 4: Final Answer:
3.

> Quick Tip
>
> For a DFA that accepts strings ending with a pattern of length m, you typically need $m + 1$ states (one for each prefix length of the pattern).

---

42. The language $L = \{\langle M \rangle \mid M$ halts on all inputs$\}$ is _____.

(A) decidable
(B) undecidable but recognizable
(C) not recognizable
(D) regular

Correct Answer: (C) not recognizable

Solution:

Step 1: Understanding the Question:
This is the Totality Problem: determine whether a Turing machine halts on every input.

Step 2: Key Formula or Approach:
Use classic computability results (Rice's theorem and known undecidability classifications).

Step 3: Detailed Explanation:
The Totality Problem is undecidable and in fact not even recursively enumerable (not recognizable). A recognizer would have to confirm that M halts on infinitely many or all possible inputs, which cannot be done algorithmically. The complement (machines that do not halt on some input) is recognizable, but the language itself is not.

Step 4: Final Answer:
Not recognizable.

Quick Tip

Remember: HALT (does M halt on input w?) is recognizable but undecidable. Totality (halts on all inputs) is harder — not recognizable.

---

43. Inherited attributes in SDTs are used for _____.

(A) passing information up the parse tree
(B) passing information down the parse tree
(C) storing symbol table entries
(D) generating target code

Correct Answer: (B) passing information down the parse tree

Solution:

Step 1: Understanding the Question:
Distinguish inherited vs. synthesized attributes in syntax-directed translation.

Step 2: Key Formula or Approach:
Synthesized attributes pass information up (from children to parent). Inherited attributes pass information from parent/siblings down to children.

Step 3: Detailed Explanation:
Inherited attributes are computed from parent or sibling attributes and are used to convey context (e.g., type information, inherited scope) downwards. Example: passing a type from a declaration node to identifier children.

Step 4: Final Answer:
Passing information down the parse tree.

Quick Tip

Mnemonic: you **inherit** characteristics **down** from parents; you **synthesize** results **up** from children.

---

44. The output of a syntax-directed translation scheme is _____.

(A) Abstract Syntax Tree
(B) Three Address Code
(C) Postfix Notation

(D) Depends on the SDT rules

Correct Answer: (D) Depends on the SDT rules

Solution:

Step 1: Understanding the Question:
SDT is a framework that attaches semantic actions to grammar productions.

Step 2: Key Formula or Approach:
The semantic actions determine what is produced during parsing.

Step 3: Detailed Explanation:
Depending on the embedded actions, an SDT can produce an AST, TAC, postfix, generate code, evaluate expressions, or do other transformations. There is no fixed single output — it depends on the rules.

Step 4: Final Answer:
Depends on the SDT rules.

Quick Tip

Think of SDT as "grammar + recipe"; the "recipe" (semantic actions) decides the final product.

---

45. Subset Construction method refers to _____.

(A) Conversion of NFA to DFA
(B) DFA minimization
(C) Eliminating Null references
(D) DFA to NFA

Correct Answer: (A) Conversion of NFA to DFA

Solution:

Step 1: Understanding the Question:
Identify what the Subset Construction algorithm does.

Step 2: Key Formula or Approach:
Each DFA state corresponds to a subset of NFA states.

Step 3: Detailed Explanation:
The algorithm constructs an equivalent DFA by treating sets of NFA states (including ϵ-closures) as DFA states, thereby determinizing the nondeterministic automaton.

Step 4: Final Answer:
Conversion of NFA to DFA.

> Quick Tip
>
> "Subset" = each DFA state = subset of NFA states. That hint usually gives the correct mapping immediately.

---

46. The maximum number of transitions which can be performed over a state in a DFA? $\Sigma = \{a, b, c\}$

(A) 8
(B) 2
(C) 3
(D) 7

Correct Answer: (C) 3

Solution:

Step 1: Understanding the Question:
For each symbol in the alphabet, a DFA state must have exactly one outgoing transition.

Step 2: Key Formula or Approach:
Number of transitions = size of alphabet $|\Sigma|$.

Step 3: Detailed Explanation:
Given $\Sigma = \{a, b, c\}$, each state has one transition for 'a', one for 'b', and one for 'c', so exactly 3 transitions per state.

Step 4: Final Answer:
3.

> Quick Tip
>
> In a DFA, outgoing transitions per state = $|\Sigma|$. In an NFA it can be 0..many.

---

47. Which of the following is a regular language?

(A) A string whose length is a sequence of prime numbers
(B) A string with substring ww'
(C) A palindrome
(D) A string with even number of zero's

Correct Answer: (D) A string with even number of zero's

Solution:

Step 1: Understanding the Question:
Decide which language can be recognized by a finite automaton.

Step 2: Key Formula or Approach:
Finite automata can track a finite amount of information (e.g., parity), but cannot test arbitrary primes, equality of arbitrary-length substrings, or general palindromes.

Step 3: Detailed Explanation:
- (A) length is prime $\rightarrow$ not regular (primality requires unbounded counting).
- (B) substring of form ww' (matching two arbitrary-length substrings) $\rightarrow$ not regular.
- (C) palindrome $\rightarrow$ not regular (requires unbounded memory to compare halves).
- (D) even number of 0's $\rightarrow$ regular: 2-state DFA toggles parity on each '0'.
Thus (D) is regular.

Step 4: Final Answer:
A string with even number of zeros is regular.

> Quick Tip
>
> Parity properties (even/odd) are classic regular-language examples solvable with tiny DFAs.

---

48. The minimum number of nodes in a DFA that recognizes strings over $\{a, b\}$ with length $\mod 3 = 0$ are _____.

(A) 4
(B) 2
(C) 3
(D) 1

Correct Answer: (C) 3

Solution:

Step 1: Understanding the Question:
We must accept strings whose length is a multiple of 3. Track length modulo 3.

Step 2: Key Formula or Approach:
States correspond to remainders $\{0, 1, 2\}$ modulo 3.

Step 3: Detailed Explanation:
Use three states: q_0 (length $\equiv 0 \bmod 3$, accepting), q_1 (length $\equiv 1$), q_2 (length $\equiv 2$). Reading any symbol advances remainder by 1 (mod 3). Minimum states required = 3.

Step 4: Final Answer:
3.

> Quick Tip
>
> To recognize length mod $k = r$, you need exactly k states.

---

49. If the production rules are given as:
$S \to XY \mid W$
$X \to aXb \mid \epsilon$
$Y \to cY \mid \epsilon$
$W \to aWc\ Z$
$Z \to bZ \mid \epsilon$
Then the language generated by these rules is

(A) $\{a^i b^j c^k \mid i, j, k \geq 0\}$
(B) $\{a^i b^j c^k \mid i, j, k \geq 1\}$
(C) $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = j \text{ or } i = k\}$
(D) $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = j \text{ or } i \neq k\}$

Correct Answer: (C) $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = j \text{ or } i = k\}$

Solution:

Step 1: Understanding the Question:
Analyze languages from each top-level alternative XY and W, then union them.

Step 2: Key Formula or Approach:
X generates $a^i b^i$ $(i \geq 0)$, Y generates c^k $(k \geq 0)$. So XY yields $a^i b^i c^k$.
Z generates b^j $(j \geq 0)$. The W production nests a and c equally around recursion,

yielding strings with equal numbers of a and c: $a^i c^i$, followed (or interleaved according to production order) by b^j, so effectively $a^i b^j c^i$.

Step 3: Detailed Explanation:
Union of the two forms gives $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = j\}$ (from XY) and $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = k\}$ (from W). Hence the language is $\{a^i b^j c^k \mid i, j, k \geq 0,\ i = j \text{ or } i = k\}$.

Step 4: Final Answer:
Option (C).

Quick Tip

When the start symbol has alternatives, parse each alternative to determine its pattern and then union the results. Productions of the form $aXb \mid \epsilon$ are typical generators of equal-count patterns $a^n b^n$.

---

50. Let P be a regular language and Q be a context free language such that $Q \subseteq P$. Then which of the following is ALWAYS regular?

(A) $P \cap Q$
(B) $P - Q$
(C) $\Sigma^* - P$
(D) $\Sigma^* - Q$

Correct Answer: (C) $\Sigma^* - P$

Solution:

Step 1: Understanding the Question:
Given P is regular, Q is CFL with $Q \subseteq P$. Which of the listed languages must be regular?

Step 2: Key Formula or Approach:
Regular languages are closed under complement.

Step 3: Detailed Explanation:
Since P is regular, its complement $\Sigma^* \setminus P$ is regular. Other options need not be regular: $P \cap Q$ is context-free in general; $P \setminus Q$ need not be regular; complement of Q need not be regular.

Step 4: Final Answer:
$\Sigma^* - P$ is always regular.

Quick Tip

Closure facts: Regular languages are closed under complement; CFLs are not closed under complement in general.

---

51. Which one of the following is FALSE?

(A) Every NFA can be converted to an equivalent PDA
(B) Complement of every context free language is recursive
(C) There is a unique minimal DFA for every regular language
(D) Every nondeterministic PDA can be converted to equivalent deterministic PDA

Correct Answer: (D) Every nondeterministic PDA can be converted to equivalent deterministic PDA

Solution:

Step 1: Understanding the Question:
Identify the false statement among standard automata-theory facts.

Step 2: Key Formula or Approach:
Recall expressiveness relationships: NFA $\subseteq$ PDA (PDA can simulate NFA), CFLs are decidable, minimal DFA uniqueness, and DPDA vs NPDA power.

Step 3: Detailed Explanation:
(A) True: a PDA can simulate an NFA by ignoring its stack.
(B) True: Every CFL is decidable (recursive), so its complement is also recursive.
(C) True: Minimal DFA (with unreachable states removed) is unique up to isomorphism.
(D) False: Nondeterministic PDAs are strictly more powerful than deterministic PDAs; not every NPDA has an equivalent DPDA.

Step 4: Final Answer:
(D) is false.

Quick Tip

Remember hierarchies: DFA/NFA (same power) $\subset$ DPDA $\subset$ NPDA (non-deterministic PDAs can accept more CFLs).

---

52. Which of the following is not a three-address code statement?

(A) $x = y + z$
(B) `if x goto L`
(C) $x = *y$
(D) $x = y^z$

Correct Answer: (D) $x = y^z$

Solution:

Step 1: Understanding the Question:
Identify which statement is not a typical primitive in three-address code (TAC).

Step 2: Key Formula or Approach:
TAC primitives are simple operations like `x = y op z`, unary ops, conditional jumps, loads/stores.

Step 3: Detailed Explanation:
(A) and (B) are standard TAC forms. (C) represents an indirect load/dereference, common in TAC. (D) exponentiation is not a canonical TAC primitive; it's usually lowered into calls or sequences of multiplications. Thus (D) is not a standard TAC primitive.

Step 4: Final Answer:
(D).

Quick Tip

TAC aims to keep operations simple (at most three addresses). Complex operators are decomposed.

---

53. Which of the following is not shared among threads in the same process?

(A) Heap memory
(B) Stack
(C) File Descriptors
(D) Code section

Correct Answer: (B) Stack

Solution:

Step 1: Understanding the Question:
Which resource is private per thread?

Step 2: Key Formula or Approach:
Threads share code, global/static data, heap, file descriptors; each thread has its own stack and registers.

Step 3: Detailed Explanation:
Stack is per-thread for call frames and local (auto) variables, therefore not shared.

Step 4: Final Answer:
Stack.

Quick Tip

Short mnemonic: "HeAP and globals are sHAred; Stacks and registers are pRivate."

---

54. The primary advantage of user-level threads over kernel-level threads is

(A) lower context-switch overhead
(B) better parallelism on multicore CPUs
(C) no need for synchronization
(D) higher priority scheduling

Correct Answer: (A) lower context-switch overhead

Solution:

Step 1: Understanding the Question:
Compare ULT and KLT advantages.

Step 2: Key Formula or Approach:
ULT context switches happen in user space without kernel traps, making them cheaper.

Step 3: Detailed Explanation:
ULTs are fast to create and switch because no kernel involvement is required; however, they cannot utilize multiple cores unless mapped to kernel threads. Synchronization is still needed.

Step 4: Final Answer:
Lower context-switch overhead.

Quick Tip

Trade-off: user-level = fast switches, kernel-level = true parallelism and kernel support.

---

55. A program uses environment variables to

(A) store data between login sessions
(B) pass configuration settings to other programs
(C) store data persistently
(D) pass data to the operating system

Correct Answer: (B) pass configuration settings to other programs

Solution:

Step 1: Understanding the Question:
What are environment variables primarily used for?

Step 2: Key Formula or Approach:
Environment variables are inherited by child processes to convey configuration.

Step 3: Detailed Explanation:
Common uses: PATH, HOME, LANG — configuration values passed to child processes. They are not inherently persistent across sessions unless stored in initialization scripts.

Step 4: Final Answer:
Pass configuration settings to other programs.

Quick Tip

Environment variables are process-level configuration inherited by child processes.

---

56. Which is not a CPU scheduling criterion?

(A) CPU utilisation
(B) Reliability
(C) Response time
(D) Waiting time

Correct Answer: (B) Reliability

Solution:

Step 1: Understanding the Question:
Pick the item not usually listed as a scheduling criterion.

Step 2: Key Formula or Approach:
Common criteria: CPU utilization, throughput, turnaround, waiting time, response time, fairness.

Step 3: Detailed Explanation:
Reliability is a system-level property, not a standard metric used to evaluate scheduling policies.

Step 4: Final Answer:
Reliability.

Quick Tip

Remember canonical scheduler metrics: utilisation, throughput, turnaround, waiting, response, fairness.

---

57. Among priority inversion between two processes, one with high priority and other with low priority that share a critical section will cause which of the following problem?

(A) High priority process executes before low priority process and finishes faster than it should.
(B) Low priority process executes before high priority process and finishes faster than it should.
(C) High priority process waits for low priority process to finish, but the low priority process never gets scheduled.
(D) Low priority process changes priority temporary to the priority of the high priority process.

Correct Answer: (C) High priority process waits for low priority process to finish, but the low priority process never gets scheduled.

Solution:

Step 1: Understanding the Question:
Describe the classical priority inversion scenario.

Step 2: Key Formula or Approach:
Priority inversion occurs when a low-priority task holds a resource needed by a high-priority task and the low-priority task is preempted by medium-priority tasks.

Step 3: Detailed Explanation:
The high-priority process becomes blocked waiting for the low-priority process to release the resource; meanwhile, medium-priority processes preempt the low-priority process, preventing it from running and releasing the resource — causing the high-priority process to wait (inversion).

Step 4: Final Answer:
Option (C).

Quick Tip

Mitigations: priority inheritance or priority ceiling protocols.

---

58. A currently running process can be put on a ready queue or one of the I/O queues by each of the following except

(A) the process issued an i/o request
(B) the process did an illegal memory access
(C) there was an interrupt
(D) the process issued a system call

Correct Answer: (B) the process did an illegal memory access

Solution:

Step 1: Understanding the Question:
Which event does not simply move the process to ready/I/O queues?

Step 2: Key Formula or Approach:
Normal blocking events: I/O requests, blocking syscalls; interrupts cause preemption. Illegal memory access causes traps/signals/termination.

Step 3: Detailed Explanation:
Illegal memory access typically raises a fault (e.g., segmentation fault) that leads to signal delivery or termination, not enqueuing on normal ready/I/O queues.

Step 4: Final Answer:
(B).

> Quick Tip
>
> Distinguish normal blocking (I/O, waiting) from exceptional faults (illegal memory access) which often result in signals or process termination.

---

59. Which of the following component of program state is shared across threads in a multithreaded process?

(A) Registers
(B) Auto variables
(C) Heap memory
(D) Stack memory

Correct Answer: (C) Heap memory

Solution:

Step 1: Understanding the Question:
Identify which component is shared among threads.

Step 2: Key Formula or Approach:
Threads in a process share code, global/static data, heap, and file descriptors; each thread has its own stack and registers.

Step 3: Detailed Explanation:
Heap memory is allocated from the process heap and is visible to all threads. Registers, auto (local) variables, and the stack are per-thread.

Step 4: Final Answer:
Heap memory.

> Quick Tip
>
> One-liner: "Heap and globals are shared; stack and registers are private to each thread."

---

60. Which of the following frees the CPU from having to deal with transfer of memory to or from an I/O device?

(A) Interrupts
(B) Direct Memory Access
(C) Buffer cache
(D) Device Driver

Correct Answer: (B) Direct Memory Access

Solution:

Step 1: Understanding CPU's Role in I/O Transfer
In simple I/O methods like Programmed I/O or Interrupt-driven I/O, the CPU is actively involved in the data transfer process. It must execute instructions to move data byte by byte or word by word between the I/O device controller and main memory, which is inefficient for large data transfers.

Step 2: Analyzing the Options
1. **Interrupts:** An interrupt is a signal to the CPU from a device, but in interrupt-driven I/O, the CPU is still responsible for executing the code (the interrupt service routine) that performs the data transfer.

2. **Direct Memory Access (DMA):** This is a specific hardware mechanism designed to solve this problem. A special controller, the DMA controller, is given permission to take control of the memory bus. It can then transfer blocks of data directly between an I/O device and main memory without any CPU intervention. The CPU is only involved at the beginning (to set up the transfer) and at the end (when the DMA controller sends an interrupt to signal completion).

3. **Buffer cache:** This is an area in memory used to temporarily store data being transferred. The CPU still manages the movement of data into and out of this cache.

4. **Device Driver:** This is software that the CPU runs to control an I/O device. It is the opposite of freeing the CPU; it is the code the CPU executes to manage the device.

Step 3: Final Answer:
Direct Memory Access (DMA) is the hardware feature that frees the CPU from having to deal with the actual data transfer between memory and I/O devices.

> Quick Tip
>
> Think of DMA as delegating a task. The CPU tells the DMA controller, "Move this block of data from the disk to this memory location and let me know when you're done." The CPU is then free to do other work until the task is complete.

61. In FCFS, I/O bound processes may have to wait a long time in the ready queue waiting for a CPU bound job to finish. This is known as _____.

(A) Aging
(B) Convoy effect
(C) Priority inversion
(D) Belady's anomaly

Correct Answer: (B) Convoy effect

Solution:

Step 1: Understanding FCFS Scheduling
First-Come, First-Served (FCFS) is a non-preemptive CPU scheduling algorithm where processes are allocated the CPU in the order they arrive in the ready queue. Once a process gets the CPU, it runs until it completes its CPU burst or blocks for I/O.

Step 2: CPU-bound vs. I/O-bound Processes
1. **CPU-bound process:** A process that spends most of its time doing computations and has long CPU bursts.
2. **I/O-bound process:** A process that spends most of its time waiting for I/O operations and has very short CPU bursts.

Step 3: Explaining the Convoy Effect
1. Consider a situation in an FCFS queue where a long CPU-bound process arrives just before several short I/O-bound processes.
2. The CPU-bound process gets the CPU and runs for its entire long burst.
3. All the I/O-bound processes, which only need the CPU for a very short time, are forced to wait in the ready queue behind the long process.
4. This leads to poor utilization of both the CPU (as it's tied up by one process) and the I/O devices (which are idle because all the processes that would use them are stuck waiting for the CPU).
5. This specific problem, where short processes get stuck waiting behind a long one, is known as the **convoy effect**.

Step 4: Analyzing Other Options
- **Aging:** A technique to prevent starvation in priority-based scheduling by gradually increasing the priority of processes that wait for a long time.
- **Priority inversion:** A problem in priority scheduling where a high-priority task is indirectly blocked by a lower-priority task.
- **Belady's anomaly:** A phenomenon in some page replacement algorithms where increasing the number of page frames results in an increase in the number of page faults.

Step 5: Final Answer:
The phenomenon described is known as the convoy effect.

Quick Tip

Imagine a narrow road (the CPU) where a very slow, long truck (the CPU-bound process) is in front of a line of fast sports cars (the I/O-bound processes). The entire line of cars is forced to move at the truck's slow speed. This is the convoy effect.

---

62. In general, there will be _____ inodes than directories in Unix File system.

(A) zero
(B) less
(C) same
(D) more

Correct Answer: (D) more

Solution:

Step 1: Understanding the Question:
The question asks to compare the number of inodes and the number of directories in a typical Unix-like file system.

Step 2: Key Formula or Approach:
We need to understand the relationship between files, directories, and inodes.
- An **inode** (index node) is a data structure that stores metadata about a file system object (like a file or directory). This includes permissions, owner, size, timestamps, and pointers to the actual data blocks. Every file and directory has exactly one inode.
- A **directory** is a special type of file whose content is a list of other files and directories and their corresponding inode numbers.

Step 3: Detailed Explanation:
In a Unix file system, every object, whether it's a regular file (text file, executable), a directory, a symbolic link, etc., requires an inode to store its information.

Since directories are themselves a type of file, every directory has an inode. However, a file system also contains many other regular files that are not directories. Each of these regular files also has its own unique inode.

Therefore, the total number of inodes will be the sum of (inodes for directories) + (inodes for regular files) + (inodes for other objects). This sum will always be greater than the number of inodes for directories alone.

Step 4: Final Answer:
In general, there will be more inodes than directories in a Unix file system.

> Quick Tip
>
> Think of it this way: the set of all directories is a subset of the set of all files in the file system. Since every file (including directories) has an inode, the total number of inodes must be greater than just the number of directories.

---

63. To ensure that the _____ condition never occurs in the system, we must guarantee that, whenever a process requests a resource, it does not have any other resource.

(A) mutual exclusion
(B) no-preemption
(C) circular wait
(D) hold and wait

Correct Answer: (D) hold and wait

Solution:

Step 1: Understanding the Question:
The question describes a strategy for deadlock prevention and asks which of the four necessary conditions for deadlock this strategy aims to eliminate.

Step 2: Key Formula or Approach:
Deadlock can only occur if four conditions hold simultaneously:
1. **Mutual Exclusion:** At least one resource must be held in a non-sharable mode.
2. **Hold and Wait:** A process is holding at least one resource and is waiting to acquire additional resources that are currently being held by other processes.
3. **No Preemption:** Resources cannot be forcibly taken away from a process; they must be released voluntarily.
4. **Circular Wait:** A set of waiting processes P0, P1, ..., Pn must exist such that P0 is waiting for a resource held by P1, P1 is waiting for a resource held by P2, ..., and Pn is waiting for a resource held by P0.

Step 3: Detailed Explanation:
The strategy described is: "whenever a process requests a resource, it does not have any other resource."

This directly negates the **Hold and Wait** condition. The "Hold and Wait" condition requires a process to be **holding** a resource while **waiting** for another. By guaranteeing

that a process cannot hold any resources when it makes a new request, we ensure this condition can never be met. One way to implement this is to require processes to request all their needed resources at once before starting execution.

Step 4: Final Answer:
The described guarantee is a method to prevent the "hold and wait" condition for deadlock.

> Quick Tip
>
> To prevent deadlock, you only need to break one of the four necessary conditions. The "hold and wait" condition is often targeted by protocols that require a process to request all resources upfront or release all current resources before requesting new ones.

---

64. A _____ architecture assigns only a few essential functions to the kernel, including address spaces, Interprocess communication (IPC) and basic scheduling.

(A) monolithic kernel
(B) micro kernel
(C) macro kernel
(D) mini kernel

Correct Answer: (B) micro kernel

Solution:

Step 1: Understanding the Question:
This is a definition-based question about different types of operating system kernel architectures. It describes a minimalist approach to kernel design.

Step 2: Key Formula or Approach:
There are two primary kernel architectures:
- **Monolithic Kernel:** The entire operating system (including scheduling, file systems, device drivers, memory management, etc.) runs as a single large program in kernel space. Examples include Linux and traditional Unix.
- **Microkernel:** The kernel is kept as small as possible, providing only the most fundamental services like memory management (address spaces), inter-process communication (IPC), and basic process scheduling. Other services like device drivers and file systems run as separate processes in user space.

Step 3: Detailed Explanation:
The question describes an architecture where "only a few essential functions" like address

spaces, IPC, and scheduling are assigned to the kernel. This perfectly matches the definition of a **microkernel**. The goal of this design is to improve reliability and security by moving non-essential components out of the privileged kernel mode.

The terms "macro kernel" and "mini kernel" are not standard terminology for mainstream OS architectures in this context.

Step 4: Final Answer:
The described architecture is that of a microkernel.

> Quick Tip
>
> Remember the trade-off: Microkernels are generally more modular and secure, but the frequent IPC required between user-space services and the kernel can lead to higher overhead and potentially lower performance compared to a monolithic kernel.

---

65. Among the four criteria for synchronization mechanism, which of the two criteria are mandatory?

(A) Mutual Exclusion and Progress
(B) Progress and Architectural neural
(C) Bounded wait and mutual exclusion
(D) Bounded wait and progress

Correct Answer: (A) Mutual Exclusion and Progress

Solution:

Step 1: Understanding the Question:
The question asks to identify the two most fundamental, or mandatory, criteria that any valid solution to the critical-section problem must satisfy.

Step 2: Key Formula or Approach:
A correct solution to the critical-section problem must satisfy three main requirements:
1. **Mutual Exclusion:** If a process P is executing in its critical section, then no other processes can be executing in their critical sections. This is the core purpose of synchronization.
2. **Progress:** If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely. This ensures the system doesn't grind to a halt.
3. **Bounded Waiting:** There must be a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request

to enter its critical section and before that request is granted. This prevents starvation of any single process.

Step 3: Detailed Explanation:
Of these three, **Mutual Exclusion** and **Progress** are considered the absolute mandatory requirements. A solution is fundamentally broken if it violates either of these.

Bounded Waiting is a fairness condition that is highly desirable but is sometimes considered a secondary requirement to the core functionality provided by the first two. A system could technically function without Bounded Waiting, but some processes might starve. It cannot function at all without Mutual Exclusion and Progress.

Step 4: Final Answer:
The two mandatory criteria are Mutual Exclusion and Progress.

> Quick Tip
>
> Think of it like traffic control: Mutual Exclusion is "only one car in the intersection at a time." Progress is "if the light is green and cars are waiting, a car must be allowed to go." Bounded Waiting is "you won't have to wait forever at the red light." The first two are essential for the system to work; the third is for fairness.

---

66. Which SQL operation is not part of the ACID properties?

(A) COMMIT
(B) ROLLBACK
(C) CREATE INDEX
(D) SAVEPOINT

Correct Answer: (C) CREATE INDEX

Solution:

Step 1: Understanding the Question:
The question asks to identify which SQL command is not directly related to the ACID properties of database transactions.

Step 2: Key Formula or Approach:
ACID is an acronym for the four properties that guarantee reliable transaction processing: - **Atomicity:** A transaction is an "all or nothing" operation. It either completes fully (commits) or has no effect at all (rolls back). - **Consistency:** A transaction brings the database from one valid state to another. - **Isolation:** Concurrent transactions do not interfere with each other. The result is the same as if the transactions were executed

serially. - **Durability:** Once a transaction has been committed, its effects are permanent, even in the event of a system failure.

Step 3: Detailed Explanation:
Let's analyze the given SQL operations:
- **COMMIT:** This command makes the changes of a transaction permanent, directly relating to Atomicity and Durability.

- **ROLLBACK:** This command undoes the changes of a transaction, which is the other half of ensuring Atomicity.

- **SAVEPOINT:** This command sets a point within a transaction to which you can later roll back, providing finer control over Atomicity.

- **CREATE INDEX:** This is a Data Definition Language (DDL) command. It creates a data structure (an index) to improve the speed of data retrieval operations. While the creation of an index itself might be done within a transaction, the command itself is about database schema and performance optimization, not a fundamental part of the transactional control flow that defines the ACID properties.

Step 4: Final Answer:
CREATE INDEX is a DDL operation for performance tuning and is not a core part of the transaction management commands that implement ACID properties.

> Quick Tip
>
> ACID properties are managed by Transaction Control Language (TCL) commands like COMMIT, ROLLBACK, and SAVEPOINT. Commands that define or modify the database structure, like CREATE, ALTER, and DROP, are part of the Data Definition Language (DDL).

---

67. A B+ tree is preferred over a B-tree for databases because _____.

(A) it supports range queries efficiently
(B) it has a smaller node size
(C) it allows faster insertion/deletion
(D) it uses less disk space

Correct Answer: (A) it supports range queries efficiently

Solution:

Step 1: Understanding the Question:
The question asks for the primary advantage of using a B+ tree over a standard B-tree,

particularly in the context of database indexing.

Step 2: Key Formula or Approach:
The key difference between B-trees and B+ trees lies in their structure:
- **B-tree:** Stores keys and data pointers in both internal nodes and leaf nodes. - **B+ tree:** Stores all data pointers exclusively in the leaf nodes. Internal nodes only store keys to guide the search. Additionally, all leaf nodes are linked together in a sequential linked list.

Step 3: Detailed Explanation:
This structural difference leads to a major performance advantage for B+ trees in certain types of queries:
- **Range Queries:** A query like 'SELECT * FROM users WHERE age BETWEEN 30 AND 40' is a range query. In a B+ tree, the database can perform a search to find the first leaf node in the range (age 30). Then, it can simply traverse the linked list of leaf nodes sequentially to find all other records in the range. This is extremely efficient.
- In a B-tree, a range query would require traversing up and down the tree multiple times to find all the relevant records, as data pointers are scattered throughout the internal and leaf nodes. This is much less efficient.
The other options are generally incorrect. For a given order, B+ trees can often store more keys in their internal nodes (since they don't store data pointers), which can sometimes lead to a shorter, bushier tree, potentially improving performance, but the primary, defining advantage is for range queries.

Step 4: Final Answer:
The linked list structure of the leaf nodes in a B+ tree allows it to support range queries and sequential access far more efficiently than a B-tree.

Quick Tip

When you see B+ Tree, think "efficient range scans". This is the main reason they are the de facto standard for database indexes.

---

68. A relation R(A,B,C,D) with functional dependencies A→B, B→C is in _____.

(A) 1NF but not 2NF
(B) 2NF but not 3NF
(C) 3NF but not BCNF
(D) BCNF

Correct Answer: (B) 2NF but not 3NF

Solution:

Note: This question is slightly ambiguous as the primary key is not explicitly stated. A standard interpretation is required to arrive at the intended answer. Let's find the candidate key first.

Step 1: Find the Candidate Key
- We find the closure of each attribute: A+ = A,B,C. This does not determine D. - This means any candidate key must include the attribute D, as it is not determined by any other attributes. - Let's test AD: (AD)+ = A,D,B,C. This determines all attributes in the relation. - Therefore, the only candidate key for R is AD.

Step 2: Check for Second Normal Form (2NF)
A relation is in 2NF if it is in 1NF and contains no partial dependencies. A partial dependency exists when a non-prime attribute is functionally dependent on a proper subset of a candidate key.
- Candidate Key: AD - Prime attributes: A, D - Non-prime attributes: B, C - We are given the dependency A $\rightarrow$ B. - Here, a non-prime attribute (B) is dependent on a proper subset of the candidate key (A is a subset of AD). - This is a partial dependency. Therefore, the relation R is not in 2NF. It is in 1NF but not 2NF.

Step 3: Re-evaluating based on the Answer Key
The provided answer is "2NF but not 3NF". This contradicts the standard analysis above. This situation typically arises in exams if the question has a typo or makes an unstated assumption. The only way for the relation to be in 2NF is if there are no partial dependencies. This would happen if the key was just a single attribute, like A. Let's assume the question intended for R(A,B,C) with key A, and D was an extraneous attribute. Or perhaps R(A,B,C,D) with key A was intended, which is not possible as A does not determine D. A more likely scenario is that the key was intended to be A,D but the FD's were meant to avoid partial dependency, e.g. AD $\rightarrow$ B.

However, to justify the given answer, we must assume the intended primary key was A, which would make the relation in 2NF but not 3NF due to the transitive dependency A$\rightarrow$B$\rightarrow$C.

Step 4: Final Answer:
Based on the standard analysis, the relation is in 1NF but not 2NF. However, to align with the provided answer key, one must assume the intended primary key was A, which would make the relation in 2NF but not 3NF due to the transitive dependency A$\rightarrow$B$\rightarrow$C.

Quick Tip

Always start normalization problems by finding all candidate keys. Then, check for 2NF (no partial dependencies), 3NF (no transitive dependencies), and BCNF (every determinant is a superkey) in that order. Be aware that exam questions can sometimes be flawed.

69. In two-phase locking (2PL), a transaction must be _____.

(A) release all locks before acquiring new ones
(B) acquire all locks before releasing any
(C) use only shared locks
(D) avoid deadlocks by timeouts

Correct Answer: (B) acquire all locks before releasing any

Solution:

Step 1: Understanding the Question:
This is a definition question about the Two-Phase Locking (2PL) protocol used for concurrency control in databases.

Step 2: Key Formula or Approach:
Two-Phase Locking is a protocol that ensures serializability. It divides the execution of a transaction into two distinct phases:
1. **Growing Phase (or Expansion Phase):** In this phase, a transaction may obtain locks on data items, but may not release any lock.
2. **Shrinking Phase (or Contraction Phase):** In this phase, a transaction may release locks, but may not obtain any new lock.

Step 3: Detailed Explanation:
The fundamental rule of 2PL is that a transaction cannot acquire any new locks after it has released its first lock. This means all lock acquisition must happen before any lock releasing begins. Option (B), "acquire all locks before releasing any," is a concise way of stating this core principle. It ensures that the "growing phase" is completed before the "shrinking phase" starts.

The other options are incorrect. Option (A) is the opposite of the rule. 2PL can use both shared and exclusive locks. While 2PL ensures serializability, it does not prevent deadlocks; in fact, deadlocks are a common issue with 2PL that must be handled separately (e.g., by timeouts or deadlock detection).

Step 4: Final Answer:
The rule for two-phase locking is that a transaction must acquire all the locks it needs before it begins releasing any of them.

Quick Tip

Remember the two phases: Phase 1 is for "growing" your set of locks (acquisitions only). Phase 2 is for "shrinking" it (releases only). You can never go back to the growing phase after you start shrinking.

70. The isolation level that prevents dirty reads but allows non-repeatable reads is _____.

(A) read uncommitted
(B) read committed
(C) repeatable read
(D) serializable

Correct Answer: (B) read committed

Solution:

Step 1: Understanding the Question:
The question asks to identify the specific SQL transaction isolation level that has a certain set of properties regarding concurrency phenomena.

Step 2: Key Formula or Approach:
Let's define the phenomena and isolation levels:
- **Dirty Read:** A transaction reads data that has been modified by another transaction that has not yet committed.

- **Non-Repeatable Read:** A transaction reads the same row twice but gets different values because another transaction has modified and committed the data in between the reads.

- **Phantom Read:** A transaction re-executes a query returning a set of rows that satisfy a search condition and finds that additional rows have been inserted by another committed transaction.

The standard isolation levels offer increasing protection:
- **Read Uncommitted:** Allows dirty reads, non-repeatable reads, and phantom reads. - **Read Committed:** Prevents dirty reads. Allows non-repeatable reads and phantom reads. - **Repeatable Read:** Prevents dirty reads and non-repeatable reads. Allows phantom reads. - **Serializable:** Prevents all three phenomena.

Step 3: Detailed Explanation:
We are looking for the isolation level that "prevents dirty reads" but "allows non-repeatable reads". According to the definitions above, this is exactly the guarantee provided by the **Read Committed** isolation level. It ensures a transaction only reads data that has been committed, but it does not guarantee that the data will remain unchanged for the duration of the transaction.

Step 4: Final Answer:
The isolation level is Read Committed.

Quick Tip

This hierarchy of isolation levels and the phenomena they prevent is a crucial concept in database transaction management. A good way to remember is that each level adds one more guarantee than the level below it.

---

71. The maximum number of keys in a B-tree of order 5 with height 3 is _____.

(A) 31
(B) 156
(C) 624
(D) 3125

Correct Answer: (B) 156

Solution:

Note: This question is highly ambiguous due to differing definitions of "height" and whether it asks for nodes or keys. The provided answer (156) corresponds to the maximum number of *nodes*, not keys, under a specific definition of height. We will proceed to justify this answer.

Step 1: Define Parameters
- Order ‘m = 5‘. - In a B-tree of order ‘m‘, a node can have at most ‘m‘ children and ‘m-1‘ keys. So, max keys/node = 4, max children/node = 5. - Height ‘h = 3‘. Let’s assume height is defined as the number of edges on the longest path from the root to a leaf. A root-only tree has h=0. A tree with height 3 therefore has 4 levels (level 0 to level 3).

Step 2: Calculate Maximum Number of Nodes
To get the maximum number of nodes, we assume every node has the maximum number of children (‘m‘).
- Level 0 (root): 1 node - Level 1: $m^1 = 5$ nodes
- Level 2: $m^2 = 25$ nodes
- Level 3: $m^3 = 125$ nodes
The total number of nodes is the sum of nodes at all levels:

$$\text{Total Nodes} = 1 + 5 + 25 + 125 = 156$$

This can also be calculated using the geometric series formula:

$$\text{Total Nodes} = \sum_{i=0}^{h} m^i = \frac{m^{h+1} - 1}{m - 1} = \frac{5^{3+1} - 1}{5 - 1} = \frac{5^4 - 1}{4} = \frac{624}{4} = 156$$

Step 3: Justification of the Answer
The number 156 matches the calculated maximum number of *nodes*. The question

explicitly asks for the maximum number of *keys*. The maximum number of keys would be the (max number of nodes) $\times$ (max keys per node) $= 156 \times 4 = 624$.

Given the options, it is overwhelmingly likely that the question intended to ask for the maximum number of nodes but mistakenly used the word keys. Under this assumption, 156 is the correct answer.

Step 4: Final Answer:
Assuming the question intended to ask for the maximum number of *nodes* in a B-tree of order 5 and height 3 (4 levels), the answer is 156.

Quick Tip

Be wary of B-tree questions involving height, as definitions can vary. "Height" can mean number of levels or number of edges. If your calculation doesn't match an option, try the alternate definition. Also, double-check if the question might be confusing "nodes" and "keys".

---

72. In _____ relation, each tuple has relation R within it.

(A) primary
(B) prime
(C) nested
(D) atomic

Correct Answer: (C) nested

Solution:

Step 1: Understanding the Question:
The question describes a type of relation where an attribute of a tuple can itself be another relation. This violates the principles of the basic relational model.

Step 2: Key Formula or Approach:
The standard relational model is based on the First Normal Form (1NF), which requires that all attribute values be **atomic**. This means that each cell in a table must hold a single, indivisible value.

Step 3: Detailed Explanation:
The scenario described—"each tuple has relation R within it"—means that an attribute's value is not atomic but is instead a complex, multi-valued object (a set of tuples, i.e., a relation). This is the defining characteristic of a **nested relation**.

Nested relational models are an extension of the basic relational model and are sometimes referred to as Non-First Normal Form (NF^2 or N1NF) models. They allow for more complex, hierarchical data to be stored more naturally within a single relation.

Step 4: Final Answer:
In a nested relation, an attribute of a tuple can be a relation itself.

> Quick Tip
>
> Remember that the foundation of classical relational databases is 1NF, which requires atomic values. Any model that allows an attribute to be a list, a set, or another relation is moving beyond 1NF and can be described as having nested relations.

---

73. The rule which states that addition of same attributes to the right side and left side will result in other valid dependency is classified as _____.

(A) Augmentation rule
(B) Inference rule
(C) Referential rule
(D) Reflexive rule

Correct Answer: (A) Augmentation rule

Solution:

Step 1: Understanding the Question:
The question asks for the name of a specific inference rule for functional dependencies in database theory.

Step 2: Key Formula or Approach:
The formal rules for reasoning about functional dependencies are known as Armstrong's axioms. The three basic axioms are:
1. **Reflexivity:** If Y is a subset of X, then $X \rightarrow Y$.

2. **Augmentation:** If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any set of attributes Z.

3. **Transitivity:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

Step 3: Detailed Explanation:
The rule described in the question is: "addition of same attributes to the right side and left side will result in other valid dependency".

Let's say we start with a valid dependency X $\rightarrow$ Y. Let the "same attributes" we add be the set Z. - Adding Z to the left side gives XZ. - Adding Z to the right side gives YZ. The rule states that the new dependency XZ $\rightarrow$ YZ will also be valid.

This is precisely the definition of the **Augmentation rule**.

Step 4: Final Answer:
The described rule is the Augmentation rule.

> Quick Tip
>
> Armstrong's axioms (Reflexivity, Augmentation, Transitivity) are the foundation for all functional dependency reasoning. "Augment" means to make larger by adding to it, which is exactly what this rule does to both sides of a dependency.

---

74. The separation of the data definition from program is known as _____.

(A) Data dictionary
(B) Data independence
(C) Data integrity
(D) Referential integrity

Correct Answer: (B) Data independence

Solution:

Step 1: Understanding the Question:
The question asks for the term that describes the principle of separating how data is defined (its schema) from the application programs that use the data.

Step 2: Key Formula or Approach:
This is a core concept in database management systems (DBMS). The primary goal of a DBMS is to provide an abstract view of the data to the users and application developers. This abstraction is achieved through different levels of schema (external, conceptual, physical).

Step 3: Detailed Explanation:
Data Independence is the ability to modify a schema definition at one level without affecting the schema definition at the next higher level. There are two types:
1. Physical Data Independence: The ability to change the physical schema (how data is stored on disk, e.g., changing file organization or indexing) without requiring changes to the conceptual schema or application programs.
2. Logical Data Independence: The ability to change the conceptual schema (the logical

structure of the entire database, e.g., adding a new attribute or table) without requiring changes to the external schema or application programs.

Both of these are forms of separating the data definition from the programs that use it. Therefore, "Data Independence" is the correct term.

Step 4: Final Answer:
The separation of the data definition from the program is known as data independence.

> Quick Tip
>
> Data independence is a key benefit of using a DBMS over traditional file systems. It allows the database structure and storage to evolve without breaking all the applications that depend on it.

---

75. Which is the candidate key for the relation R(ABCDE) with functional dependencies F=A→BC, B→D, CD→ E, E→ A?

(A) A
(B) CD
(C) B
(D) E

Correct Answer: (A) A

Solution:

Step 1: Understanding the Question:
We need to find a candidate key for the relation R with attributes A, B, C, D, E and a given set of functional dependencies (FDs). A candidate key is a minimal set of attributes that can uniquely determine all other attributes in the relation.

Step 2: Key Formula or Approach:
To find a candidate key, we can compute the attribute closure for each of the potential keys given in the options. The attribute closure of a set of attributes X, denoted as X+, is the set of all attributes that are functionally determined by X. If the closure contains all attributes of the relation R, then X is a superkey. If X is also minimal (no proper subset of X is a superkey), then X is a candidate key.

Step 3: Detailed Explanation:
Let's compute the closure for each option:
- **Test A:** - A+ = A (start with A) - Since A → BC, A+ = A, B, C - Since B → D, A+ = A, B, C, D - Since CD → E, A+ = A, B, C, D, E - The closure of A, A+, contains all

attributes of R. Since A is a single attribute, it is minimal. Therefore, **A is a candidate key**.

- **Test CD:** - (CD)+ = C, D - Since CD → E, (CD)+ = C, D, E - Since E → A, (CD)+ = C, D, E, A - Since A → BC, (CD)+ = C, D, E, A, B - The closure of CD also contains all attributes. So, CD is also a candidate key.

- **Test B:** - B+ = B - Since B → D, B+ = B, D - This does not determine A, C, or E. So, B is not a candidate key.

- **Test E:** - E+ = E - Since E → A, E+ = E, A - Since A → BC, E+ = E, A, B, C - Since B → D, E+ = E, A, B, C, D - The closure of E also contains all attributes. So, E is also a candidate key.

The relation has multiple candidate keys: A, CD, and E. Since A is listed as option 1 and is a valid candidate key, it is the correct choice.

Step 4: Final Answer:
A is a candidate key for the relation R.

> Quick Tip
>
> To quickly find candidate keys, first identify attributes that are not on the right side of any FD. These must be part of any candidate key. In this case, no attribute fits this description. Then, test the closures of the left sides of the FDs.

---

76. Natural Join can also be termed as _____.

(A) combination of Union and Cartesian Product
(B) combination of Selection and Cartesian Product
(C) combination of Projection and Cartesian Product
(D) combination of Projection and Union

Correct Answer: (C) combination of Projection and Cartesian Product

Solution:
Note: The most accurate answer would be "a combination of Selection, Cartesian Product, and Projection". However, given the options, we must choose the best fit.

Step 1: Understanding the Question:
The question asks for an equivalent definition of the Natural Join operation using more fundamental relational algebra operations.

Step 2: Key Formula or Approach:
A Natural Join ($\bowtie$) between two relations, R and S, is an operation that combines their tuples based on matching values in their common attributes.

Step 3: Detailed Explanation:
The process of a Natural Join can be broken down:
1. **Cartesian Product ($\mathbf{R \times S}$):** First, a Cartesian product of the two relations is performed, creating a new relation with all attributes from both R and S, and all possible combinations of their tuples.
2. **Selection (σ):** From the result of the Cartesian product, we select only those tuples where the values of the common attributes are equal. For every attribute 'A' common to both R and S, we apply the condition 'R.A = S.A'.
3. **Projection (π):** Finally, since the common attributes are now duplicated, we project the result to eliminate the duplicate columns, keeping only one instance of each common attribute.

So, Natural Join $\equiv \pi(\sigma(R \times S))$. It is a combination of Cartesian Product, Selection, and Projection.

Let's re-examine the options. Option (B) includes Selection and Cartesian Product, but misses Projection. Option (C) includes Projection and Cartesian Product but misses Selection. Both are incomplete. However, often in textbook simplifications, the Selection part is considered the core of the "join" logic, and Projection is the final cleanup. Some might argue that the 'EQUI JOIN' (Selection + Cartesian Product) followed by Projection is the best description. There might be an error in the question or options. If we must choose, both (B) and (C) are partially correct. However, let's consider a Theta Join as 'Selection(Cartesian Product)'. A Natural join is a specific type of Theta join (an equijoin) followed by a projection to remove duplicate columns. The answer key points to (C), which is unusual. Let's try to justify it. One could argue that the essence is taking a product and then projecting out the desired columns, with the equality selection being an implicit part of the join condition itself. This is not a standard view. Given the likely error, and the fact that a join is fundamentally a product followed by filtering and column selection, (C) is a plausible, though imperfect, choice presented.

Step 4: Final Answer:
Following the provided key, the answer is a combination of Projection and Cartesian Product. A more complete answer would also include Selection.

> Quick Tip
>
> The most precise definition of Natural Join is 'Projection(Selection(Cartesian Product))'. Remember that it joins on all common columns, selects rows where they are equal, and removes the duplicate columns.

77. If A → B has trivial functional dependency, then _____.

(A) B is a subset of A
(B) A is a subset of B
(C) A is a subset of A'
(D) B is a subset of B'

Correct Answer: (A) B is a subset of A

Solution:

Step 1: Understanding the Question:
The question asks for the definition of a trivial functional dependency.

Step 2: Key Formula or Approach:
A functional dependency (FD) X → Y is said to be **trivial** if the set of attributes on the right-hand side (Y) is a subset of the set of attributes on the left-hand side (X).

Step 3: Detailed Explanation:
The given functional dependency is A → B.

For this dependency to be trivial, the attributes in B must be a subset of the attributes in A.

For example, if A = StudentID, Name and B = Name, then the dependency StudentID, Name → Name is trivial. We already know the name if we know the student ID and the name. Trivial dependencies are always true by definition and do not represent any real constraint on the data.

Therefore, the condition for A → B to be trivial is $B \subseteq A$.

Step 4: Final Answer:
If A → B is a trivial functional dependency, then B is a subset of A.

Quick Tip

A trivial dependency is "obvious" information. If you know a set of attributes, you certainly know any subset of those attributes. For a dependency X → Y, if Y is part of X, it's trivial. Otherwise, it's non-trivial.

---

78. Which of the following commands is used to delete all rows and free up space from a table?

(A) Drop
(B) Delete
(C) Truncate
(D) Alter

Correct Answer: (C) Truncate

Solution:

Step 1: Understanding the Question:
The question asks for the SQL command that removes all rows from a table and also deallocates the space used by those rows efficiently.

Step 2: Key Formula or Approach:
Let's compare the relevant SQL commands:
- **DELETE FROM tablename;** This is a Data Manipulation Language (DML) command. It removes all rows one by one and logs each deletion. Because it's a logged operation, it is generally slower and can be rolled back. It does not typically reset the high-water mark or deallocate space immediately.
- **TRUNCATE TABLE tablename;** This is a Data Definition Language (DDL) command. It removes all rows from a table by deallocating the data pages. It is not logged in the same way as DELETE, making it much faster. It cannot be rolled back in most database systems and resets any identity columns. It frees up the storage space used by the table's data.
- **DROP TABLE tablename;** This DDL command removes the entire table structure and its data completely from the database. This is more than just deleting the rows.

Step 3: Detailed Explanation:
The question's requirements are "delete all rows" and "free up space".

'DELETE' meets the first requirement but is inefficient and may not meet the second requirement immediately.
'DROP' is too destructive as it removes the table itself.
'TRUNCATE' is the command designed specifically for this purpose: a high-performance, non-logged operation to remove all data from a table and reclaim its storage space.

Step 4: Final Answer:
The 'TRUNCATE' command is used to delete all rows and free up space from a table.

Quick Tip

- 'DELETE' = Remove some/all rows (slow, can be rolled back). - 'TRUNCATE' = Remove all rows (fast, cannot be rolled back, frees space). - 'DROP' = Remove the entire table.

79. _____ uses Quantifiers that can be either Existential (∃) or Universal (∀).

(A) Domain Relational Calculus
(B) Tuple Relational Calculus
(C) Distributed Relational Calculus
(D) NoSQL

Correct Answer: (B) Tuple Relational Calculus

Solution:

Step 1: Understanding the Question:
The question asks to identify the type of relational calculus that uses quantifiers over variables. The symbols ∃ (Existential, "there exists") and ∀ (Universal, "for all") are mentioned.

Step 2: Key Formula or Approach:
Relational Calculus is a non-procedural query language. There are two main forms:
- **Tuple Relational Calculus (TRC):** The variables in TRC range over tuples of relations. A typical query is of the form ‘ t | P(t) ‘, which means "find all tuples ‘t‘ such that predicate ‘P‘ is true for ‘t‘". The predicate P can use quantifiers like ∃ and ∀ to bind these tuple variables.
- **Domain Relational Calculus (DRC):** The variables in DRC range over the domains (data types like integer, string) of the attributes, not entire tuples. A query is of the form ‘ <x, y, z> | P(x, y, z) ‘. While it also uses logic, the primary variables are domain variables.

Step 3: Detailed Explanation:
Both TRC and DRC use logical quantifiers. Typically, Tuple Relational Calculus is the formalism more directly associated with queries that look like ‘ t | ... $\exists s \in R$...‘. $For instance, to find all courses taught in the Fall semester, a TRC query would be : ‘t|\exists s \in sectio$

Given the options, Tuple Relational Calculus is the most direct and common answer associated with using these quantifiers over relations.

Step 4: Final Answer:
Tuple Relational Calculus uses existential (∃) and universal (∀) quantifiers over tuple variables.

Quick Tip

- Tuple Relational Calculus: Variables represent tuples. ‘t | ...‘. - Domain Relational Calculus: Variables represent attribute domain values. ‘<x, y> | ...‘. Both use logical quantifiers, but TRC is the classic example taught with tuple variables.

---

80. The database design that consists of multiple tables that are linked together through matching data stored in each table is called a _____.

(A) Relational database
(B) Network database
(C) Object oriented database
(D) Hierarchical database

Correct Answer: (A) Relational database

Solution:

Step 1: Understanding the Question:
The question asks for the name of the database model characterized by multiple tables connected by matching data values.

Step 2: Key Formula or Approach:
We need to know the defining characteristics of the major database models.
- **Hierarchical Model:** Data is organized in a tree-like structure. Records are linked in parent-child relationships.

- **Network Model:** A more flexible version of the hierarchical model, allowing many-to-many relationships in a graph-like structure.

- **Relational Model:** Data is organized into tables (relations) consisting of rows (tuples) and columns (attributes). Relationships between tables are established by matching values in common columns, typically using primary keys and foreign keys.

- **Object-Oriented Model:** Data is stored as objects, similar to object-oriented programming.

Step 3: Detailed Explanation:
The description "multiple tables that are linked together through matching data" is the exact definition of the **relational database model**. For example, an ‘Orders‘ table might have a ‘CustomerID‘ column that "matches" the ‘CustomerID‘ primary key in a ‘Customers‘ table, thereby linking an order to the customer who placed it. This linking through shared values is the cornerstone of the relational model.

Step 4: Final Answer:
The described design is that of a relational database.

> Quick Tip
>
> When you think of modern databases with tables, rows, columns, and keys (like MySQL, PostgreSQL, SQL Server), you are thinking of the relational database model.

---

81. The incremental model of software development is _____.

(A) a reasonable approach when requirements are well defined
(B) a good approach when a working core product is required quickly
(C) the best approach to use for projects with large development teams
(D) a revolutionary model that is not used for commercial products

Correct Answer: (B) a good approach when a working core product is required quickly

Solution:

Step 1: Understanding the Question:
The question asks to identify the most accurate description or best-use case for the incremental model of software development.

Step 2: Key Formula or Approach:
The **incremental model** is a software development process where requirements are broken down into multiple standalone modules of the software development cycle. Each increment builds on the previous one, adding new functionality. The first increment is often a core product with basic functionality. Subsequent increments add features until the full system is complete.

Step 3: Detailed Explanation:
Let's analyze the options:
- (A) When requirements are well defined and unlikely to change, a sequential model like the Waterfall model is often considered reasonable. Incremental models are better when requirements can be chunked or are expected to evolve.

- (B) The key advantage of the incremental model is that it delivers a working version of the software early in the development lifecycle. This "working core product" can be delivered to the customer for feedback or early use, which is a significant benefit. This statement accurately describes a primary strength of the model.

- (C) For large teams, models that support parallel development and integration, like the spiral model or agile methodologies, are often preferred. While incremental can be used,

it's not inherently the "best" for large teams.

- (D) The incremental model is a well-established, practical model used widely in commercial product development. It is not revolutionary and is definitely used.

The most fitting description is that it allows a core product to be developed and deployed quickly.

Step 4: Final Answer:
The incremental model is a good approach when a working core product is required quickly.

> Quick Tip
>
> Think of the incremental model as building a house. The first increment is the foundation and frame (a core, usable structure). The next increment adds walls and a roof. The next adds plumbing and electricity. You get a usable piece of the final product early and add to it over time.

---

82. The cyclomatic complexity metric provides the designer with information regarding number of _____.

(A) cycles in the program
(B) errors in the program
(C) independent logic paths in the program
(D) statement in the program

Correct Answer: (C) independent logic paths in the program

Solution:

Step 1: Understanding the Question:
This question asks for the definition of McCabe's Cyclomatic Complexity, a software metric.

Step 2: Key Formula or Approach:
Cyclomatic complexity is a quantitative measure of the complexity of a program. It is computed using the control flow graph of the program. The formula is:

$$V(G) = E - N + 2P$$

where 'E' is the number of edges, 'N' is the number of nodes, and 'P' is the number of connected components (usually 1 for a single program/function).

Step 3: Detailed Explanation:
The value 'V(G)' represents the number of linearly **independent paths** through the program's source code. An independent path is one that introduces at least one new edge that has not been traversed before in other independent paths.

Essentially, it measures the amount of decision logic ('if', 'while', 'for', 'case' statements) in a code block. A higher cyclomatic complexity indicates more complex code with more branching, which can be harder to test and maintain. The metric provides a lower bound for the number of test cases required to achieve full branch coverage.

Step 4: Final Answer:
Cyclomatic complexity measures the number of independent logic paths in a program.

> Quick Tip
>
> A simple way to calculate cyclomatic complexity is to count the number of decision points (like 'if', 'while', 'for') in the code and add 1. This gives you a quick estimate of the program's complexity and the minimum number of tests needed.

---

83. In the spiral model of software development, the primary determinant in selecting activities in each iteration is _____.

(A) iteration size
(B) cost
(C) risk
(D) adopted process such as rational unified process or extreme programming

Correct Answer: (C) risk

Solution:

Step 1: Understanding the Question:
The question asks for the main factor that drives the planning and activities within each cycle of the Spiral Model.

Step 2: Key Formula or Approach:
The Spiral Model is an evolutionary software process model that combines elements of both prototyping and the waterfall model. It is known for its emphasis on risk management. The model proceeds in a series of iterations, with each loop of the spiral representing a phase of the software process.

Step 3: Detailed Explanation:
Each iteration in the Spiral Model begins with the identification of objectives, alternatives, and constraints for that phase. The next and most crucial step is the evaluation of

these alternatives, with a primary focus on identifying and resolving **risks**.

The activities selected for the iteration (e.g., building a prototype, running simulations, using formal methods) are chosen specifically to mitigate the highest-priority risks identified. If the primary risk is related to the user interface, the iteration will focus on UI prototyping. If it's related to performance, the iteration will focus on benchmarking and analysis. Therefore, risk analysis is the primary determinant of what happens in each cycle.

Step 4: Final Answer:
In the spiral model, risk is the primary determinant in selecting activities in each iteration.

> Quick Tip
>
> When you see "Spiral Model", the key word to associate with it is "Risk". The model is fundamentally a risk-driven approach to software development, making it well-suited for large, complex, and high-risk projects.

---

84. The optimizer that explores the space of all query-evaluation plans is called _____.

(A) cost-based
(B) plan-based
(C) estimate-based
(D) count-based

Correct Answer: (A) cost-based

Solution:

Step 1: Understanding the Question:
The question asks for the name of a database query optimizer that evaluates multiple possible execution plans to find the most efficient one.

Step 2: Key Formula or Approach:
A query optimizer is a component of a database management system that attempts to determine the most efficient way to execute a given query. There are two main types:

- **Heuristic/Rule-Based Optimizer:** Uses a set of predefined rules to transform the query into a reasonably good execution plan. It doesn't typically compare multiple plans based on their estimated execution cost.

- **Cost-Based Optimizer (CBO):** Generates multiple possible execution plans for a query. It then estimates the "cost" (in terms of I/O operations, CPU time, etc.) of each plan, using statistics about the data stored in the database. Finally, it selects the plan with the lowest estimated cost for execution.

Step 3: Detailed Explanation:
The description "explores the space of all query-evaluation plans" precisely defines the operation of a **cost-based** optimizer. It systematically generates different ways to execute the query (e.g., different join orders, different access methods like index scans vs. table scans) and uses a cost model to choose the best one. The other terms are not standard classifications for query optimizers. "Estimate-based" is part of what a cost-based optimizer does, but "cost-based" is the proper name for the entire strategy.

Step 4: Final Answer:
The optimizer that explores the space of all query-evaluation plans and chooses the one with the lowest estimated cost is called a cost-based optimizer.

> Quick Tip
>
> Modern sophisticated database systems almost universally use cost-based optimizers because they can produce far better execution plans for complex queries than simpler rule-based optimizers.

---

85. The term that optimizes subexpressions shared by different expressions in a program is called _____.

(A) multiple subexpression elimination
(B) common subexpression elimination
(C) parametric subexpression elimination
(D) shared subexpression elimination

Correct Answer: (B) common subexpression elimination

Solution:

Step 1: Understanding the Question:
This is a terminology question from compiler design, asking for the name of a specific optimization technique.

Step 2: Key Formula or Approach:
The technique described involves identifying instances of identical subexpressions that appear in a program and ensuring they are evaluated only once.

Step 3: Detailed Explanation:
Common Subexpression Elimination (CSE) is a compiler optimization that searches for instances of identical expressions (i.e., they evaluate to the same value) and analyzes whether it is worthwhile to replace them with a single variable holding the computed value.

For example, in the code:

```
a = (b * c) + g;
d = (b * c) * e;
```

The subexpression ‘(b * c)‘ is common to both statements. A CSE optimization would transform this to:

```
temp = b * c;
a = temp + g;
d = temp * e;
```

This avoids re-computing the multiplication, potentially making the code faster. The other options are not standard terms for this widely-known optimization.

Step 4: Final Answer:
The optimization is called common subexpression elimination.

> Quick Tip
>
> Common Subexpression Elimination is one of the most important and fundamental "peephole" and global optimizations performed by modern compilers to improve code efficiency.

---

86. Frequency of failure and network recovery time after a failure are measures of the _____ of a network.

(A) performance
(B) reliability
(C) security
(D) feasibility

Correct Answer: (B) reliability

Solution:

Step 1: Understanding the Question:
The question asks which quality attribute of a network is measured by metrics like "frequency of failure" and "recovery time".

Step 2: Key Formula or Approach:
Let's define the network attributes given in the options:
- **Performance:** Typically measured by metrics like throughput (data rate), latency (delay), and jitter. It's about how fast the network is. - **Reliability:** The ability of a system or component to function under stated conditions for a specified period. It is often measured by metrics related to failures, such as Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR). - **Security:** The protection of the network and its data from unauthorized access, use, disclosure, disruption, modification, or destruction. - **Feasibility:** The likelihood that a proposed network design can be successfully created and implemented.

Step 3: Detailed Explanation:
"Frequency of failure" is directly related to MTBF. A low frequency of failure means a high MTBF.
"Network recovery time after a failure" is the same as MTTR.

These two metrics—how often something fails and how quickly it recovers—are the standard ways to quantify **reliability**. A reliable network is one that fails infrequently and recovers quickly when it does.

Step 4: Final Answer:
Frequency of failure and recovery time are measures of the reliability of a network.

Quick Tip

- Performance = Speed (Throughput, Latency) - Reliability = Uptime (Failure Rate, Recovery Time) - Security = Safety (Protection against attacks)

---

87. A Gantt chart is least useful for tracking _____.

(A) task dependencies
(B) resource allocation
(C) critical path
(D) code complexity

Correct Answer: (D) code complexity

Solution:

Step 1: Understanding the Question:
The question asks what a Gantt chart is not well-suited for tracking, among the given options.

Step 2: Key Formula or Approach:
A Gantt chart is a project management tool. It is a type of bar chart that illustrates a project schedule over time. We need to evaluate its capabilities against the options.

Step 3: Detailed Explanation:
- **Task Dependencies:** Gantt charts can show dependencies between tasks, often with lines or arrows linking the end of one task to the beginning of another.

- **Resource Allocation:** Gantt charts are frequently used to show which resources (e.g., people, equipment) are assigned to which tasks over time.

- **Critical Path:** While a PERT chart is better for explicitly calculating and displaying the critical path, modern Gantt chart software can often highlight the sequence of critical tasks.

- **Code Complexity:** This is a software metric (like Cyclomatic Complexity) that measures the internal quality and intricacy of the source code. Gantt charts are project management tools concerned with schedules and resources; they have no capability to analyze or track the quality or complexity of the actual code being written.

Therefore, a Gantt chart is least useful for tracking code complexity.

Step 4: Final Answer:
A Gantt chart is a project scheduling tool and is not used for tracking internal software quality metrics like code complexity.

Quick Tip

Associate project management tools with project-level concerns (schedule, resources, dependencies, progress). Associate software metrics with code-level concerns (complexity, bugs, maintainability). A Gantt chart is a project management tool.

---

88. Which phase of the SDLC typically consumes the most resources?

(A) Requirements gathering
(B) Testing
(C) Maintenance
(D) Coding

Correct Answer: (C) Maintenance

Solution:

Step 1: Understanding the Question:
The question asks which phase of the Software Development Life Cycle (SDLC) is generally the most resource-intensive, which includes factors like cost, time, and effort.

Step 2: Key Formula or Approach:
The SDLC consists of several phases, including planning, requirements, design, coding, testing, deployment, and maintenance. We need to consider the entire lifespan of a typical software product.

Step 3: Detailed Explanation:
While phases like coding and testing are intensive during the initial development, the **Maintenance** phase encompasses the entire period after the software is deployed until it is retired.

This phase can last for many years and includes:
- **Corrective maintenance:** Fixing bugs discovered by users. - **Adaptive maintenance:** Modifying the software to work in a new environment (e.g., new OS, new hardware). - **Perfective maintenance:** Adding new features and enhancements requested by users. - **Preventive maintenance:** Restructuring code to improve maintainability and prevent future problems.

Studies have consistently shown that for successful, long-lived software, the total cost and effort spent on maintenance far exceeds the initial development cost, often accounting for 60% to 80% of the total lifetime cost.

Step 4: Final Answer:
Over the entire life of a software system, the maintenance phase typically consumes the most resources.

> Quick Tip
>
> Don't just think about the initial creation of software. The real cost of software is often in keeping it running, relevant, and bug-free for years after its first release. This is why the maintenance phase is the most expensive.

---

89. Regression testing is performed to ensure that _____.

(A) new code doesn't break existing functionality
(B) all paths are covered
(C) users accept the system
(D) compliance with laws

Correct Answer: (A) new code doesn't break existing functionality

Solution:

Step 1: Understanding the Question:
The question asks for the primary purpose of regression testing.

Step 2: Key Formula or Approach:
Regression testing is a type of software testing that focuses on the impact of change. "Regression" refers to the software "regressing" or going backward into a worse state (i.e., a previously working feature is now broken).

Step 3: Detailed Explanation:
Whenever a change is made to a software system—such as fixing a bug or adding a new feature—there is a risk that this change could have unintended side effects, breaking other parts of the application that were previously working correctly.

Regression testing involves re-running a suite of existing tests to verify that the recent code changes have not adversely affected existing functionalities. Its goal is to catch these regressions.

The other options represent different types of testing:
- (B) all paths are covered: This is the goal of path coverage testing, a white-box technique. - (C) users accept the system: This is User Acceptance Testing (UAT). - (D) compliance with laws: This is compliance testing.

Step 4: Final Answer:
The purpose of regression testing is to ensure that new code changes do not break existing functionality.

> Quick Tip
>
> Think of the word "regress," which means to return to a former, less developed state. Regression testing checks if your software has regressed after you've made a change.

---

90. The COCOMO model estimates _____.

(A) project risk
(B) software cost
(C) team productivity
(D) user satisfaction

Correct Answer: (B) software cost

Solution:

Step 1: Understanding the Question:
The question asks what the COCOMO (Constructive Cost Model) is used to estimate.

Step 2: Key Formula or Approach:
COCOMO is an algorithmic software cost estimation model. It uses formulas, primarily based on the estimated size of the software project (in lines of code or function points), to predict project metrics.

Step 3: Detailed Explanation:
The primary outputs of the COCOMO model are:
1. **Effort:** The amount of labor required to complete the project, usually expressed in person-months.
2. **Schedule:** The estimated time required to complete the project, in months.

The effort estimate (person-months) is then used to calculate the overall **software cost** by multiplying it by the average cost per person-month. While factors related to team productivity are used as inputs (cost drivers) to the model, the main output is cost and schedule, not productivity itself. Project risk and user satisfaction are not directly estimated by the model.

Step 4: Final Answer:
The COCOMO model is used to estimate software cost.

Quick Tip

The name itself is a clue: COnstructive COst MOdel. Its primary purpose is to estimate the cost (and effort/schedule) of a software project.

---

91. At which OSI layer does MAC address filtering occur?

(A) Network
(B) Data Link
(C) Transport
(D) Physical

Correct Answer: (B) Data Link

Solution:

Step 1: Understanding the Question:
The question asks to identify the layer in the OSI (Open Systems Interconnection) model

where operations related to MAC addresses, such as filtering, take place.

Step 2: Key Formula or Approach:
We need to know the primary responsibilities and addressing schemes of the OSI layers. - Layer 1 (Physical): Deals with bits and the physical medium. No concept of addressing. - Layer 2 (**Data Link**): Deals with frames and provides reliable transit of data across a physical link. It uses MAC (Media Access Control) addresses for local network addressing. - Layer 3 (Network): Deals with packets and provides routing between different networks. It uses logical addresses like IP addresses. - Layer 4 (Transport): Deals with segments/datagrams and provides end-to-end communication services. It uses port numbers.

Step 3: Detailed Explanation:
The MAC address is a unique hardware identifier assigned to a network interface controller (NIC). It is used for communication within a local network segment (like an Ethernet LAN).

Since MAC addressing is a fundamental part of the Data Link Layer, any operation that involves inspecting or making decisions based on MAC addresses, such as MAC address filtering in a switch or wireless access point, occurs at **Layer 2, the Data Link Layer**.

Step 4: Final Answer:
MAC address filtering occurs at the Data Link layer.

Quick Tip

Remember the address types for each layer: - Layer 2 (Data Link): MAC Addresses (Physical) - Layer 3 (Network): IP Addresses (Logical) - Layer 4 (Transport): Port Numbers (Service)

---

92. Token Ring networks avoid collisions by _____.

(A) CSMA/CD
(B) Token passing
(C) Exponential backoff
(D) Priority queuing

Correct Answer: (B) Token passing

Solution:

Step 1: Understanding the Question:
The question asks for the specific media access control method used by Token Ring networks to prevent data collisions.

Step 2: Key Formula or Approach:
Network access methods can be broadly categorized:
- **Contention-based:** All nodes compete for access to the medium. Collisions can occur and must be handled. Example: CSMA/CD in Ethernet. - **Controlled access:** A mechanism ensures that only one node can transmit at any given time. Collisions are inherently avoided.

Step 3: Detailed Explanation:
Token Ring is a controlled access method. It uses a special three-byte frame called a **token** that is passed from node to node around the logical ring.

A node that wishes to transmit data must wait until it receives a free token. It then captures the token, changes it to a data frame, appends its data, and sends it to the destination. Since only the node that holds the token is allowed to transmit, two nodes can never transmit at the same time, and thus **collisions are avoided**. This method is called **token passing**.

CSMA/CD is the collision detection method used in classic Ethernet. Exponential backoff is part of the CSMA/CD collision resolution algorithm.

Step 4: Final Answer:
Token Ring networks avoid collisions by using the token passing media access method.

Quick Tip

- Ethernet = Contention-based (listens for silence, sends, detects collisions). - Token Ring = Controlled access (waits for permission slip/token, then sends).

---

93. Which routing algorithm guarantees shortest paths even with negative edge weights?

(A) Dijkstra's
(B) Bellman-Ford
(C) Link-State
(D) OSPF

Correct Answer: (B) Bellman-Ford

Solution:

Step 1: Understanding the Question:
The question asks to identify a shortest-path routing algorithm that works correctly even when the graph contains edges with negative weights (as long as there are no

negative-weight cycles).

Step 2: Key Formula or Approach:
We must compare the capabilities of different shortest-path algorithms.
- **Dijkstra's Algorithm:** A greedy algorithm that works by iteratively selecting the unvisited node with the smallest known distance. Its greedy assumption only holds true if edge weights are non-negative. It fails if there are negative edge weights.
- **Bellman-Ford Algorithm:** A dynamic programming-based algorithm. It works by relaxing all edges in the graph repeatedly. This methodical approach allows it to correctly calculate shortest paths in graphs with negative edge weights. It can also be used to detect the presence of negative-weight cycles.
- **Link-State:** This is a class of routing protocols (like OSPF). They typically use Dijkstra's algorithm to compute shortest paths within a routing area, and therefore do not support negative edge weights.
- **OSPF (Open Shortest Path First):** A specific link-state routing protocol that uses Dijkstra's algorithm.

Step 3: Detailed Explanation:
Dijkstra's algorithm's core principle of finalizing the shortest path to a node once it's visited is violated by negative edges. A shorter path might be discovered later through a path that includes a negative edge.

The Bellman-Ford algorithm does not make this greedy assumption. It systematically considers paths of increasing length (in terms of number of edges), which allows it to correctly propagate the effect of negative weights throughout the graph.

Step 4: Final Answer:
The Bellman-Ford algorithm guarantees shortest paths in graphs with negative edge weights (provided no negative cycles reachable from the source exist).

> Quick Tip
>
> For shortest path problems: - No negative weights → Use Dijkstra's (faster). - Negative weights allowed → Use Bellman-Ford (slower but correct).

---

94. A TCP header includes all except _____.

(A) Sequence number
(B) Checksum
(C) TTL
(D) Window size

Correct Answer: (C) TTL

Solution:

Step 1: Understanding the Question:
The question asks to identify which of the given fields is not part of the TCP (Transmission Control Protocol) header.

Step 2: Key Formula or Approach:
We need to know the structure of the TCP header and the IP header.
- **TCP (Layer 4) Header:** Contains fields for managing a reliable, end-to-end connection. Key fields include Source Port, Destination Port, **Sequence Number**, Acknowledgment Number, Flags (SYN, ACK, FIN), **Window Size**, and **Checksum**.
- **IP (Layer 3) Header:** Contains fields for routing packets across networks. Key fields include Source IP Address, Destination IP Address, and **Time to Live (TTL)**.

Step 3: Detailed Explanation:
The Sequence Number, Checksum, and Window Size are all essential fields in the TCP header used for ordering, error checking, and flow control, respectively.

The **TTL (Time to Live)** field is part of the IP header. Its purpose is to prevent packets from looping indefinitely in the network. Each router that forwards the packet decrements the TTL value, and if it reaches zero, the packet is discarded. Since this is a network-level (routing) concern, it belongs in the IP header (Layer 3), not the TCP header (Layer 4).

Step 4: Final Answer:
TTL is a field in the IP header, not the TCP header.

> Quick Tip
>
> Remember to associate protocol features with their correct layer. TTL is for preventing network loops (a routing problem, Layer 3/IP). Sequence numbers and window size are for reliable, ordered, flow-controlled streams (an end-to-end connection problem, Layer 4/TCP).

---

95. The three-way handshake in TCP ensures _____.

(A) Encryption setup
(B) Synchronized sequence numbers
(C) QoS negotiation
(D) Multicast membership

Correct Answer: (B) Synchronized sequence numbers

Solution:

Step 1: Understanding the Question:
The question asks for the primary purpose of the three-way handshake procedure used to establish a TCP connection.

Step 2: Key Formula or Approach:
The TCP three-way handshake consists of three steps:
1. **SYN:** The client sends a segment with the SYN (Synchronize) flag set, containing its Initial Sequence Number (ISN), let's say 'ISN$_c$'.2.**SYN-ACK:***Theserverrepliesswithasegmentwithb* ISN_c+1').3.**ACK:***Theclientsendsafinalsegmentwiththe*ACK*flagset*, *acknowledgingtheserver'* $ISN_s + 1$').

Step 3: Detailed Explanation:
The primary goal of this exchange is to establish a reliable, full-duplex connection. A critical part of this is for both sides to agree upon the starting sequence numbers for their respective streams of data. TCP uses sequence numbers to ensure data is received in order and to handle duplicate or lost packets.

By exchanging and acknowledging ISNs, both the client and server **synchronize their sequence numbers**, ensuring they have a common starting point for the reliable data transfer that will follow. This process confirms that both sides are ready to communicate and have received each other's initial parameters.

Step 4: Final Answer:
The three-way handshake in TCP ensures that a reliable connection is established and that the sequence numbers used for data transfer are synchronized between the client and server.

Quick Tip

Think of the three-way handshake like a phone call: 1. SYN: "Hello, this is Client, can you hear me?" 2. SYN-ACK: "Yes, I can hear you, this is Server. Can you hear me?" 3. ACK: "Yes, I can hear you too. Let's talk." The main point is to confirm two-way communication is possible and to set a starting point for the conversation (the sequence numbers).

---

96. An IPv4 datagram with header length=5 and total length=400 bytes carries _____.

(A) 380 bytes payload
(B) 400 bytes payload
(C) 20 bytes header
(D) 320 bytes payload

Correct Answer: (A) 380 bytes payload

Solution:

Step 1: Understanding the Question:
We are given the values of the 'Header Length' and 'Total Length' fields from an IPv4 datagram and asked to calculate the size of the data it carries (the payload).

Step 2: Key Formula or Approach:
In an IPv4 header:
- The **Total Length** field gives the size of the entire datagram (header + payload) in bytes. - The **Header Length** (IHL) field gives the size of the header in 4-byte words.

The formulas are:

$$\text{Header Size (in bytes)} = \text{Header Length field value} \times 4$$

$$\text{Payload Size} = \text{Total Length} - \text{Header Size}$$

Step 3: Detailed Explanation:
We are given:
- Header Length field value = 5 - Total Length = 400 bytes

First, calculate the actual size of the header in bytes:

$$\text{Header Size} = 5 \times 4 = 20 \text{ bytes}$$

Next, calculate the payload size:

$$\text{Payload Size} = 400 \text{ bytes} - 20 \text{ bytes} = 380 \text{ bytes}$$

Step 4: Final Answer:
The IPv4 datagram carries a 380-byte payload.

> **Quick Tip**
>
> A very common mistake is forgetting to multiply the IPv4 Header Length field by 4. Remember, this field counts 32-bit (4-byte) words, not individual bytes.

---

97. Which protocol uses port 53 by default?

(A) HTTP
(B) FTP
(C) DNS

(D) SMTP

Correct Answer: (C) DNS

Solution:

Step 1: Understanding the Question:
The question asks to identify the application layer protocol associated with the well-known port number 53.

Step 2: Key Formula or Approach:
This requires knowledge of standard port numbers for common internet protocols.

Step 3: Detailed Explanation:
Let's review the standard ports for the protocols listed:
- **HTTP** (Hypertext Transfer Protocol): Port 80 - **FTP** (File Transfer Protocol): Port 21 (for control) and Port 20 (for data). - **DNS** (Domain Name System): Port 53. It uses UDP for standard queries and TCP for zone transfers. - **SMTP** (Simple Mail Transfer Protocol): Port 25

Therefore, port 53 is the well-known port for the Domain Name System.

Step 4: Final Answer:
DNS uses port 53 by default.

> Quick Tip
>
> Memorizing the port numbers for the most common protocols (HTTP-80, HTTPS-443, FTP-21, SSH-22, SMTP-25, DNS-53) is essential for any networking or security exam.

---

98. A switch differs from a hub because, it _____.

(A) operates at Layer 1
(B) broadcasts all frames
(C) uses MAC tables for forwarding
(D) only supports half-duplex

Correct Answer: (C) uses MAC tables for forwarding

Solution:

Step 1: Understanding the Question:
The question asks for a key feature that distinguishes a network switch from a network hub.

Step 2: Key Formula or Approach:
We need to compare the operational characteristics of hubs and switches.
- **Hub:** A Layer 1 (Physical Layer) device. It acts as a multiport repeater. Any frame it receives on one port is regenerated and broadcast out to all other ports, regardless of the destination. This creates a single, large collision domain.

- **Switch:** A Layer 2 (Data Link Layer) device. It is an intelligent device that inspects the destination MAC address of incoming frames. It maintains a MAC address table that maps MAC addresses to the switch ports. It forwards the frame only to the specific port connected to the destination device. This segments the network into separate collision domains for each port.

Step 3: Detailed Explanation:
Based on the definitions:
- (A) is incorrect. A switch operates at Layer 2, while a hub operates at Layer 1. - (B) is incorrect. This describes the behavior of a hub. A switch only broadcasts frames if the destination MAC is unknown or is a broadcast address. - (C) is correct. The use of a MAC address table to make intelligent forwarding decisions is the defining difference between a switch and a hub. - (D) is incorrect. Switches typically support full-duplex communication, whereas hubs are limited to half-duplex.

Step 4: Final Answer:
A switch differs from a hub because it uses MAC tables for intelligent forwarding of frames.

> Quick Tip
>
> - Hub = Dumb (repeats everything to everyone). - Switch = Smart (learns who is where and sends messages only to the intended recipient).

---

99. Digital signatures provide _____.

(A) confidentiality
(B) non-repudiation
(C) key exchange
(D) firewall bypass

Correct Answer: (B) non-repudiation

Solution:

Step 1: Understanding the Question:
The question asks to identify a primary security service provided by digital signatures.

Step 2: Key Formula or Approach:
A digital signature is created by taking a hash of a message and encrypting the hash with the sender's **private key**. The recipient can then decrypt the signature with the sender's **public key** and compare the result with their own hash of the message. This process provides several security guarantees.

Step 3: Detailed Explanation:
The security services provided by a digital signature are:
1. **Authentication:** Since only the sender has the private key, a successful decryption with the public key proves the message came from the sender.
2. **Integrity:** If the message was altered in transit, the hash calculated by the receiver will not match the decrypted hash from the signature, proving the data has been tampered with.
3. **Non-repudiation:** Because the signature can only be created with the sender's unique private key, the sender cannot later deny having sent the message. This is non-repudiation.

Confidentiality (keeping the message secret) is not provided by a digital signature itself. To achieve confidentiality, one must encrypt the *message* with the *recipient's* public key.

Step 4: Final Answer:
Among the options, digital signatures are a primary mechanism for providing non-repudiation.

> Quick Tip
>
> - Digital Signature = Hash + Encryption with Sender's Private Key. Provides Authentication, Integrity, and Non-Repudiation. - Confidentiality = Encryption of message with Recipient's Public Key.

---

100. In asynchronous transmission, the gap time between bytes is _____.

(A) fixed
(B) variable
(C) zero
(D) dependent on the clock speed

Correct Answer: (B) variable

Solution:

Step 1: Understanding the Question:
The question asks about the nature of the timing between data units (bytes) in asynchronous transmission.

Step 2: Key Formula or Approach:
We must differentiate between synchronous and asynchronous transmission.
- **Synchronous Transmission:** A continuous, high-speed stream of data bits is sent. A common clock signal is shared between the sender and receiver to keep them synchronized. The timing is precise and fixed.
- **Asynchronous Transmission:** Data is sent one character (or byte) at a time. Each byte is framed with a start bit and one or more stop bits. These bits tell the receiver when the byte begins and ends. There is no shared clock.

Step 3: Detailed Explanation:
Because there is no shared clock in asynchronous transmission, the sender can transmit bytes whenever it is ready. The receiver uses the start bit to synchronize its clock for the duration of that single byte. After the stop bit is received, the receiver waits for the next start bit.

This means the time interval, or gap, between the end of one byte (its stop bit) and the beginning of the next byte (its start bit) is not fixed. It can be of any length. This is why it is called "asynchronous"—the bytes are not synchronized to a common clock.

Step 4: Final Answer:
In asynchronous transmission, the gap time between bytes is variable.

> Quick Tip
>
> Think of "asynchronous" like sending text messages. You can send one message, wait five minutes, and then send another. The gap is variable. "Synchronous" is like a live phone call, where the conversation flows continuously without arbitrary gaps.

---

101. In Java, when we implement an interface method, it must be declared as _____.

(A) private
(B) protected
(C) public
(D) friend

Correct Answer: (C) public

Solution:

Step 1: Understanding the Question:
The question asks about the required visibility modifier for a method in a class that implements a Java interface.

Step 2: Key Concepts:
- Methods declared in a Java interface are implicitly `public` and `abstract`. - When a class implements an interface, it must provide a concrete implementation for the interface's abstract methods. - The visibility of the implementing method in the class cannot be more restrictive than the visibility of the method in the interface.

Step 3: Detailed Explanation:
Since all methods in an interface are implicitly `public`, any class that implements the interface must declare its implementation of those methods as `public`. Using a more restrictive modifier like `protected` or `private` would violate the contract of the interface and result in a compilation error. The `friend` keyword is not a valid access modifier in Java.

Step 4: Final Answer:
The implementing method must be declared as `public`.

Quick Tip

A simple rule in Java inheritance and implementation is that you cannot reduce the visibility of a method. Since interface methods are public, the implementing methods must also be public.

---

102. Which of the following control fields in TCP header is used to specify whether the sender has no more data to transport?

(A) FIN
(B) RST
(C) SYN
(D) PSH

Correct Answer: (A) FIN

Solution:

Step 1: Understanding the Question:
The question asks to identify the TCP header flag that signals the end of data transmission from the sender.

Step 2: Key Concepts:
The TCP header contains several control flags (bits) to manage the state of the connection: - **SYN (Synchronize):** Initiates a connection.
- **ACK (Acknowledge):** Acknowledges receipt of a segment.
- **FIN (Finish):** Gracefully terminates a connection.
- **RST (Reset):** Abruptly terminates a connection.
- **PSH (Push):** Asks the receiver to push the data to the application layer immediately.

Step 3: Detailed Explanation:
When a sender has finished sending all its data, it needs a way to inform the receiver that its side of the connection is closing. It does this by sending a TCP segment with the **FIN** flag set. This indicates that the sender has "no more data to transport." The receiver can then acknowledge the FIN and proceed to close its own side of the connection.

Step 4: Final Answer:
The FIN flag is used to indicate that the sender has finished sending data.

Quick Tip

Think of the TCP connection lifecycle: It starts with SYN, data is exchanged, and it ends gracefully with FIN.

---

103. In XML, DOCTYPE declaration specifies to include a reference to a _____ file.

(A) document type declaration
(B) document type definition
(C) document type language
(D) document transfer definition

Correct Answer: (B) document type definition

Solution:

Step 1: Understanding the Question:
The question asks what the ‘<!DOCTYPE>‘ declaration in an XML document refers to.

Step 2: Key Concepts:
XML (eXtensible Markup Language) documents can be validated against a formal

grammar to ensure they are well-formed and valid. The ‘<!DOCTYPE>‘ declaration is the mechanism used to associate an XML document with its grammar.

Step 3: Detailed Explanation:
The grammar that defines the structure, legal elements, and attributes for an XML document is called a **Document Type Definition (DTD)**. The ‘<!DOCTYPE>‘ declaration, which appears at the top of an XML file, specifies the DTD that the document conforms to. This allows a parser to validate the document's structure against the rules defined in the DTD.

Step 4: Final Answer:
The DOCTYPE declaration specifies a reference to a Document Type Definition (DTD) file.

Quick Tip

The acronym for "document type definition" is DTD. The DOCTYPE declaration points to the DTD.

---

104. The CSS used inside HTML elements alongside the style attribute is called _____.

(A) external CSS
(B) internal CSS
(C) inline CSS
(D) outline CSS

Correct Answer: (C) inline CSS

Solution:

Step 1: Understanding the Question:
The question asks for the name of the CSS application method that uses the ‘style‘ attribute directly within an HTML tag.

Step 2: Key Concepts:
There are three main ways to apply CSS styles to HTML:

1. **Inline CSS:** Styles are applied to a single element using the ‘style‘ attribute. Example: ‘<p style="color: blue;">‘.

2. **Internal CSS:** Styles are defined within a ‘<style>‘ block in the ‘<head>‘ section of the HTML document.

3. **External CSS:** Styles are defined in a separate ‘.css‘ file and linked to the HTML document using a ‘<link>‘ tag.

Step 3: Detailed Explanation:
The method described in the question, where styles are placed "inside HTML elements alongside the style attribute," is the definition of **inline CSS**.

Step 4: Final Answer:
The described method is called inline CSS.

Quick Tip

Remember the location: - Inline: Inside the HTML tag. - Internal: Inside the HTML file’s head. - External: In a completely separate file.

---

105. A web client sends a request to a web server. The web server transmits a program to that client and is executed at the client, which creates a web document. Such web documents are _____.

(A) dynamic
(B) static
(C) active
(D) passive

Correct Answer: (C) active

Solution:

Step 1: Understanding the Question:
The question describes a scenario where a program is sent from the server to be executed on the client side to generate a web page. We need to classify this type of web document.

Step 2: Key Concepts:
- **Static Web Document:** A fixed document stored on the server as a file (e.g., an ‘.html‘ file). The content is the same for every user.

- **Dynamic Web Document:** The document is created on the server-side by a program (e.g., PHP, Python) in response to a client’s request. The content can change based on user input, database information, etc.

- **Active Web Document:** A program (e.g., JavaScript, Java Applet) is sent from the server to the client. This program then executes on the client’s machine to create or modify the document.

Step 3: Detailed Explanation:
The scenario described—"transmits a program to that client and is executed at client"—is the definition of an **active** web document. JavaScript is the most common technology used for this purpose today.

Step 4: Final Answer:
Such web documents are called active documents.

> Quick Tip
>
> Remember where the processing happens: - Static: No processing needed, just sending a file. - Dynamic: Processing happens on the server. - Active: Processing happens on the client.

---

106. Cascading Style Sheets define style and appearance using _____.

(A) functions with parameters and return values
(B) rules with selectors, properties and their values
(C) techniques with block and inline elements
(D) heuristics with rules and maps

Correct Answer: (B) rules with selectors, properties and their values

Solution:

Step 1: Understanding the Question:
The question asks for the fundamental components that make up a CSS declaration.

Step 2: Key Concepts:
The basic syntax of CSS is composed of rules. Each rule has three main parts.

Step 3: Detailed Explanation:
A CSS rule consists of:
1. A **selector**, which targets the HTML element(s) to be styled.
2. A declaration block, enclosed in curly braces ‘‘, which contains one or more declarations.
3. Each declaration is a key-value pair consisting of a CSS **property** and its corresponding **value**, separated by a colon.

For example, in ‘h1 color: blue; ‘, ‘h1‘ is the selector, ‘color‘ is the property, and ‘blue‘ is the value. This entire structure is a rule. Therefore, CSS defines styles using rules with selectors, properties, and their values.

Step 4: Final Answer:
CSS defines style using rules with selectors, properties and their values.

> Quick Tip
>
> The basic syntax to remember for CSS is: ‘selector property: value; ‘.

---

107. What is the correct HTML for making a hyperlink?

(A) `<a href="http://abc.com">abc</a>`
(B) `<a name="http://abc.com">abc</a>`
(C) `<http://abc.com</a>`
(D) `url="http://abc.com">`

Correct Answer: (A) `<a href="http://abc.com">abc</a>`

Solution:

Step 1: Understanding the Question:
The question asks for the correct HTML syntax to create a clickable link.

Step 2: Key Concepts:
In HTML, hyperlinks are created using the anchor tag, ‘<a>‘. The destination of the link is specified by the Hypertext Reference attribute, ‘href‘.

Step 3: Detailed Explanation:
The correct syntax is ‘<a>‘ tag with an ‘href‘ attribute pointing to the destination URL. The content between the opening ‘<a>‘ tag and the closing ‘‘ tag becomes the clickable text or element.

- Option (A) correctly uses the ‘<a>‘ tag with the ‘href‘ attribute.

- Option (B) uses the ‘name‘ attribute, which was used for creating anchors within a page but is now deprecated.

- Options (C) and (D) are not valid HTML syntax.

Step 4: Final Answer:
The correct HTML is `<a href="http://abc.com">abc</a>`.

Quick Tip

Remember 'a' is for anchor, and 'href' is for hypertext reference.

---

108. What is the correct HTML for adding a background color?

(A) `<body color="yellow">`
(B) `<body bgcolor="yellow">`
(C) `<background>yellow</background>`
(D) `<body background="yellow">`

Correct Answer: (B) `<body bgcolor="yellow">`

Solution:

Step 1: Understanding the Question:
The question asks for the correct HTML attribute to set the background color of the entire page.

Step 2: Key Concepts:
In older versions of HTML (before the widespread adoption of CSS), presentation attributes were used directly in tags. - The 'bgcolor' attribute was used to set the background color of an element. - The 'color' attribute on the '<body>' tag is not standard; 'text' was used for text color. - The 'background' attribute on the '<body>' tag was used to specify a background image, not a color.

Step 3: Detailed Explanation:
Among the given choices, `<body bgcolor="yellow">` is the historically correct HTML syntax for setting the page's background color. While this attribute is now deprecated and CSS ('<body style="background-color:yellow;">') is the modern, preferred method, it is the correct answer in the context of HTML-only attributes.

Step 4: Final Answer:
The correct HTML is `<body bgcolor="yellow">`.

Quick Tip

Although 'bgcolor' is the correct answer here, always use CSS for styling in modern web development. Separating structure (HTML) from presentation (CSS) is a fundamental best practice.

---

109. Which is the correct syntax to link an XML file with CSS?

(A) `<?xml type="text/css" href="file.css"?>`
(B) `<?xml type="text/css" src="file.css"?>`
(C) `<?xml-stylesheet type="text/css" href="file.css"?>`
(D) `<?xml-stylesheet type="text/css" src="file.css"?>`

Correct Answer: (C) `<?xml-stylesheet type="text/css" href="file.css"?>`

Solution:

Step 1: Understanding the Question:
The question asks for the correct processing instruction to apply a CSS stylesheet to an XML document.

Step 2: Key Concepts:
Unlike HTML, which uses a ‘<link>‘ tag, XML uses a special processing instruction to associate stylesheets. A processing instruction starts with ‘<¿ and ends with ‘?>‘.

Step 3: Detailed Explanation:
The standard way to link a CSS file to an XML document is with the ‘xml-stylesheet‘ processing instruction. The correct syntax requires a ‘type‘ attribute (e.g., ‘"text/css"‘) and an ‘href‘ attribute that points to the location of the CSS file. The ‘src‘ attribute is not used for this purpose.

Therefore, the correct syntax is `<?xml-stylesheet type="text/css" href="file.css"?>`.

Step 4: Final Answer:
The correct syntax is given in option (C).

> Quick Tip
>
> Remember that XML uses processing instructions for metadata like stylesheets. The target is ‘xml-stylesheet‘ and the attributes are ‘type‘ and ‘href‘.

---

110. A Session variable is created _____.

(A) when the application is first placed on a web server
(B) when the web server is first started
(C) when the first client requests a URL resource
(D) every time a new client interacts with the web application

Correct Answer: (D) every time a new client interacts with the web application

Solution:

Step 1: Understanding the Question:
The question asks when a "session" is initiated in the context of a web application.

Step 2: Key Concepts:
HTTP is a stateless protocol. To maintain state and track a user across multiple page requests, web applications use sessions. A session provides a way to store user-specific data on the server.

Step 3: Detailed Explanation:
A new session is created on the server for a user when that user (a unique client/browser) makes their first request to the application and the application needs to store session data. The server generates a unique session ID, sends it to the client (usually in a cookie), and the client sends this ID back with every subsequent request. This allows the server to retrieve the correct session data for that user. Therefore, a session is initiated when a new client starts an interaction with the application. Options A, B, and C describe application or server-level events, not user-specific ones.

Step 4: Final Answer:
A session and its variables are created for each new client that begins interacting with the web application.

> Quick Tip
>
> A "session" is a server-side concept tied to a specific user's visit. It starts with the user's first request and ends after a period of inactivity (timeout) or when explicitly closed.

---

111. Let A, B be two events and $\bar{A}$ be the complement of A. If $P(\bar{A}) = 0.7$, $P(B) = 0.7$ and $P(B|A) = 0.5$, then $P(A \cup B) =$ _____.

(A) 0.65
(B) 0.85
(C) 0.75
(D) 0.50

Correct Answer: (B) 0.85

Solution:

Step 1: Find $P(A)$

The probability of an event and its complement sum to 1.

$$P(A) = 1 - P(\bar{A}) = 1 - 0.7 = 0.3$$

Step 2: Find $P(A \cap B)$
Use the formula for conditional probability: $P(B|A) = \frac{P(A \cap B)}{P(A)}$. Rearranging this gives:

$$P(A \cap B) = P(B|A) \times P(A)$$

$$P(A \cap B) = 0.5 \times 0.3 = 0.15$$

Step 3: Find $P(A \cup B)$
Use the addition rule for probability:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cup B) = 0.3 + 0.7 - 0.15$$

$$P(A \cup B) = 1.0 - 0.15 = 0.85$$

Step 4: Final Answer:
The value of $P(A \cup B)$ is 0.85.

> Quick Tip
>
> This problem requires chaining three fundamental probability rules: the complement rule, the conditional probability formula, and the addition rule.

---

112. Let a random variable X have a binomial distribution with mean 8 and standard deviation 2. If $P(X < 2) = \frac{k}{2^{16}}$, then the value of k is _____.

(A) 17
(B) 16
(C) 136
(D) 137

Correct Answer: (A) 17

Solution:

Step 1: Find the parameters n and p of the binomial distribution
For a binomial distribution, the mean is $\mu = np$ and the variance is $\sigma^2 = np(1-p)$. We are given: - Mean $\mu = np = 8$ - Standard deviation $\sigma = 2$, so the variance $\sigma^2 = 2^2 = 4$. Substitute the mean into the variance formula:

$$\sigma^2 = (np)(1-p) \implies 4 = 8(1-p)$$

$$\frac{4}{8} = 1 - p \implies 0.5 = 1 - p \implies p = 0.5$$

Now find n using the mean:

$$np = 8 \implies n(0.5) = 8 \implies n = 16$$

So, the random variable X follows a binomial distribution with $n = 16$ and $p = 0.5$, written as $X \sim B(16, 0.5)$.

Step 2: Calculate $P(X < 2)$
$P(X < 2)$ is the probability that X is either 0 or 1.

$$P(X < 2) = P(X = 0) + P(X = 1)$$

The probability mass function for a binomial distribution is $P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$.

$$P(X = 0) = \binom{16}{0} (0.5)^0 (0.5)^{16-0} = 1 \cdot 1 \cdot (0.5)^{16} = \frac{1}{2^{16}}$$

$$P(X = 1) = \binom{16}{1} (0.5)^1 (0.5)^{16-1} = 16 \cdot (0.5)^{16} = \frac{16}{2^{16}}$$

Summing these probabilities:

$$P(X < 2) = \frac{1}{2^{16}} + \frac{16}{2^{16}} = \frac{17}{2^{16}}$$

Step 3: Find k
We are given that $P(X < 2) = \frac{k}{2^{16}}$. Comparing our result, we have:

$$\frac{17}{2^{16}} = \frac{k}{2^{16}} \implies k = 17$$

Step 4: Final Answer:
The value of k is 17.

> Quick Tip
>
> For binomial distribution problems, first use the given mean and variance (or standard deviation) to solve for the parameters n and p. Then apply the binomial probability formula.

---

113. Let A = {1,2,3}. Which of the following is the number of distinct relations on A?

(A) 256
(B) 158
(C) 512
(D) 64

Correct Answer: (C) 512

Solution:

Step 1: Understanding the Question:
The question asks for the total number of possible relations on a set A with 3 elements.

Step 2: Key Concepts:
- A relation on a set A is defined as any subset of the Cartesian product $A \times A$. - The Cartesian product $A \times A$ is the set of all ordered pairs (a, b) where a and b are both elements of A. - If a set has m elements, the total number of distinct subsets of that set is 2^m. This is the size of its power set.

Step 3: Detailed Explanation:
First, we find the size of the set A: $|A| = 3$.
Next, we find the size of the Cartesian product $A \times A$:

$$|A \times A| = |A| \times |A| = 3 \times 3 = 9$$

Since a relation on A is any subset of $A \times A$, we need to find the total number of subsets of a set with 9 elements. The number of distinct relations is the size of the power set of $A \times A$, which is:

$$\text{Number of relations} = 2^{|A \times A|} = 2^9$$

$$2^9 = 512$$

Step 4: Final Answer:
There are 512 distinct relations on the set A.
Note: The provided image cuts off the options, but the correct answer is 512, which has been added as option C.

> Quick Tip
>
> To find the number of relations on a set with 'n' elements, the formula is always $2^{(n^2)}$. Here, n=3, so the answer is $2^{(3^2)} = 2^9 = 512$.

---

114. In any Lattice L, $(((a \wedge b) \vee a) \wedge b) =$_____**.**

(A) $a \vee b$
(B) $a \wedge b$
(C) $(a \wedge b) \vee a$
(D) $((a \vee b) \wedge a) \vee b$

Correct Answer: (B) $a \wedge b$

Solution:

Step 1: Understanding the Question:
We need to simplify the given lattice expression using the properties of lattices.

Step 2: Key Formula or Approach:
The key property needed here is the Absorption Law in lattice theory, which states:
For any elements a, b in a lattice:
1. $a \vee (a \wedge b) = a$
2. $a \wedge (a \vee b) = a$

Step 3: Detailed Explanation:
Let's simplify the expression step by step, starting from the innermost parenthesis.
The expression is: $((a \wedge b) \vee a) \wedge b$

First, consider the sub-expression: $(a \wedge b) \vee a$
This directly matches the form of the first absorption law, where the expression simplifies to 'a'.
So, $(a \wedge b) \vee a = a$.

Now, substitute this result back into the main expression:
The expression becomes: $a \wedge b$.
This cannot be simplified further.

Step 4: Final Answer:
The simplified expression is $a \wedge b$.

Quick Tip

Always look for applications of the Absorption Laws ($a \vee (a \wedge b) = a$ and $a \wedge (a \vee b) = a$) when simplifying lattice expressions. They are very common and powerful.

---

115. If the system of linear equations $x + 2y + z = 5$, $2x + \lambda y + 4z = 12$, $4x + 8y + 12z = 2\mu$ have an infinite number of solutions, then the values of λ and μ are _____.

(A) $\lambda = 4, \mu = 28$
(B) $\lambda = 4, \mu = 14$
(C) $\lambda = 8, \mu = 14$
(D) $\lambda = 8, \mu = 28$

Correct Answer: (B) $\lambda = 4, \mu = 14$

Solution:

Step 1: Condition for Infinite Solutions
For a system of linear equations to have an infinite number of solutions, two conditions must be met:
1. The determinant of the coefficient matrix must be zero.
2. The system must be consistent.

Step 2: Calculate the Determinant to find λ
The coefficient matrix is $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & \lambda & 4 \\ 4 & 8 & 12 \end{pmatrix}$.
We set its determinant to zero:
$\det(A) = 1(12\lambda - 32) - 2(24 - 16) + 1(16 - 4\lambda) = 0$
$12\lambda - 32 - 2(8) + 16 - 4\lambda = 0$
$12\lambda - 32 - 16 + 16 - 4\lambda = 0$
$8\lambda - 32 = 0$
$8\lambda = 32 \implies \lambda = 4.$

Step 3: Check for Consistency to find μ
Now substitute $\lambda = 4$ into the system and form the augmented matrix:

$$[A|B] = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 5 \\ 2 & 4 & 4 & 12 \\ 4 & 8 & 12 & 2\mu \end{array}\right)$$

Apply row operations to find the rank:
$R_2 \to R_2 - 2R_1$: $\left(\begin{array}{ccc|c} 1 & 2 & 1 & 5 \\ 0 & 0 & 2 & 2 \\ 4 & 8 & 12 & 2\mu \end{array}\right)$
$R_3 \to R_3 - 4R_1$: $\left(\begin{array}{ccc|c} 1 & 2 & 1 & 5 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 8 & 2\mu - 20 \end{array}\right)$
For the system to be consistent with infinite solutions, one row must be a multiple of another. We need R_3 to be a multiple of R_2.
Notice that the left side of R_3 is 4 times the left side of R_2 $(0, 0, 8$ vs $0, 0, 2)$.

For consistency, the right side must also have the same relationship:
$(2\mu - 20) = 4 \times (2)$
$2\mu - 20 = 8$
$2\mu = 28 \implies \mu = 14.$

Step 4: Final Answer:
The values are $\lambda = 4$ and $\mu = 14$.

Quick Tip

For infinite solutions, the rank of the coefficient matrix 'A' must equal the rank of the augmented matrix '[A|B]', and this rank must be less than the number of variables.

---

116. If the product and sum of eigenvalues of a 2 × 2 matrix $\begin{pmatrix} 3 & x \\ y & 1 \end{pmatrix}$ are -3 and 5, respectively, then $x + y =$ _____.

(A) -5
(B) $-6 \pm i3\sqrt{2}$
(C) $-2 \pm i3$
(D) -1

Correct Answer: (D) -1

Solution:
Note: There is a fundamental contradiction in the problem statement as written. The sum of the eigenvalues must equal the trace of the matrix. The trace of the given matrix is $3 + 1 = 4$, but the problem states the sum is 5. We will proceed by assuming the sum of eigenvalues being 5 is the intended condition, which implies the matrix has a typo and should have been, for example, $\begin{pmatrix} 3 & x \\ y & 2 \end{pmatrix}$ or another form where the diagonal sums to 5. Without a clear matrix, this problem is ill-posed. However, if we assume a typo and attempt to find a logical path to an answer, a common type of question would involve a different matrix structure. Let's assume the matrix was intended to be $\begin{pmatrix} x & 3 \\ 1 & y \end{pmatrix}$.

Step 1: Relate Eigenvalue Properties to Matrix Properties
For any square matrix:
- The sum of its eigenvalues is equal to the trace of the matrix (the sum of its diagonal elements).
- The product of its eigenvalues is equal to the determinant of the matrix.

Step 2: Formulate Equations based on a plausible intended matrix
Let's assume the intended matrix was $A = \begin{pmatrix} x & 3 \\ 1 & y \end{pmatrix}$.
Given: Sum of eigenvalues = 5, Product of eigenvalues = -3.
From the properties:
Trace(A) $= x + y \implies x + y = 5$.
Determinant(A) $= xy - (3)(1) \implies xy - 3 = -3 \implies xy = 0$.
This would give the answer $x + y = 5$, which is not an option.

Let's try another plausible matrix: $A = \begin{pmatrix} x & 1 \\ y & 3 \end{pmatrix}$.

Trace(A) $= x + 3 = 5 \implies x = 2$.
Determinant(A) $= 3x - y = -3 \implies 3(2) - y = -3 \implies 6 - y = -3 \implies y = 9$.
In this case, $x + y = 2 + 9 = 11$. Not an option.

Step 3: Work Backwards from the Answer
Given the ambiguity, this question is flawed. There is no clear path from the problem statement to the answer of -1. The contradiction between the stated trace (4) and the stated sum of eigenvalues (5) makes a direct solution impossible.

Step 4: Final Conclusion:
The problem statement is inconsistent. The trace of the given matrix is 4, while the sum of eigenvalues is stated to be 5. It is not possible to solve for $x + y$ with the information provided.

Quick Tip

Always check for consistency in eigenvalue problems. The sum of eigenvalues MUST equal the trace. If they don't match, the problem statement is flawed.

---

117. Given that the value of the integral $\int_1^9 (x^2 - 2)dx$ calculated using the Simpson's 1/3 rule with four uniform subintervals over the interval [1,9] is given by $f(1) + \alpha^2 + \frac{8}{3}$, then the possible value of α is _____.

(A) $\alpha = 15$
(B) $\alpha = 14$
(C) $\alpha = 13$
(D) $\alpha = 12$

Correct Answer: (A) $\alpha = 15$

Solution:

Step 1: Set up Simpson's 1/3 Rule
The interval is $[a, b] = [1, 9]$ and the number of subintervals is $n = 4$.
The step size is $h = \frac{b-a}{n} = \frac{9-1}{4} = 2$.
The x-values are $x_0 = 1, x_1 = 3, x_2 = 5, x_3 = 7, x_4 = 9$.
The function is $f(x) = x^2 - 2$.

Step 2: Calculate Function Values
$y_0 = f(1) = 1^2 - 2 = -1$
$y_1 = f(3) = 3^2 - 2 = 7$
$y_2 = f(5) = 5^2 - 2 = 23$
$y_3 = f(7) = 7^2 - 2 = 47$

$y_4 = f(9) = 9^2 - 2 = 79$

Step 3: Apply Simpson's 1/3 Rule Formula
The formula is $I \approx \frac{h}{3}[y_0 + 4(y_1 + y_3) + 2(y_2) + y_4]$.
$I \approx \frac{2}{3}[-1 + 4(7 + 47) + 2(23) + 79]$

$I \approx \frac{2}{3}[-1 + 4(54) + 46 + 79]$

$I \approx \frac{2}{3}[78 + 216 + 46]$

$I \approx \frac{2}{3}[340] = \frac{680}{3}$.

Step 4: Solve for α
We are given the value is also $f(1) + \alpha^2 + \frac{8}{3}$.

Since $f(1) = -1$, the value is $-1 + \alpha^2 + \frac{8}{3}$.

Equate the two expressions for the integral's value:

$\frac{680}{3} = -1 + \alpha^2 + \frac{8}{3}$

$\frac{680}{3} = \alpha^2 + \frac{8}{3} - \frac{3}{3}$

$\frac{680}{3} = \alpha^2 + \frac{5}{3}$

$\alpha^2 = \frac{680}{3} - \frac{5}{3} = \frac{675}{3}$

$\alpha^2 = 225$

$\alpha = \pm 15$.

Step 5: Final Answer:
A possible value for α is 15.

> Quick Tip
>
> For Simpson's 1/3 rule, remember the pattern of coefficients for the y-values: 1, 4, 2, 4, 2, ..., 4, 1.

---

118. $\lim_{x \to 1} \left(\frac{x^x - x}{x - 1 - \log x} \right) =$ _____.

(A) 0
(B) 1

(C) 2
(D) 1/2

Correct Answer: (C) 2

Solution:

Step 1: Check the Form of the Limit
Substitute $x = 1$ into the numerator and denominator.
Numerator: $1^1 - 1 = 0$.
Denominator: $1 - 1 - \log(1) = 0 - 0 = 0$.
This is the indeterminate form $\frac{0}{0}$, so we can apply L'Hôpital's Rule.

Step 2: Apply L'Hôpital's Rule (First Time)

We need the derivatives of the numerator and denominator.
Derivative of denominator: $\frac{d}{dx}(x - 1 - \log x) = 1 - \frac{1}{x}$.
Derivative of numerator: To find $\frac{d}{dx}(x^x)$, let $y = x^x$. Then $\ln y = x \ln x$. Differentiating gives $\frac{1}{y}y' = \ln x + 1$, so $y' = x^x(\ln x + 1)$.
So, $\frac{d}{dx}(x^x - x) = x^x(\ln x + 1) - 1$.
The limit becomes: $\lim_{x\to 1} \frac{x^x(\ln x+1)-1}{1-\frac{1}{x}}$.

Step 3: Check the Form Again

Substitute $x = 1$ again.
Numerator: $1^1(\ln 1 + 1) - 1 = 1(0 + 1) - 1 = 0$.
Denominator: $1 - \frac{1}{1} = 0$.
It is still the $\frac{0}{0}$ form. We apply L'Hôpital's Rule again.

Step 4: Apply L'Hôpital's Rule (Second Time)

Derivative of denominator: $\frac{d}{dx}(1 - x^{-1}) = -(-1)x^{-2} = \frac{1}{x^2}$.
Derivative of numerator: $\frac{d}{dx}(x^x(\ln x + 1) - 1)$. Using the product rule:
$\frac{d}{dx}(x^x)(\ln x + 1) + x^x \frac{d}{dx}(\ln x + 1)$
$[x^x(\ln x + 1)](\ln x + 1) + x^x(\frac{1}{x})$
$= x^x(\ln x + 1)^2 + x^{x-1}$.
The limit becomes: $\lim_{x\to 1} \frac{x^x(\ln x+1)^2+x^{x-1}}{\frac{1}{x^2}}$.

Step 5: Evaluate the Final Limit

Now substitute $x = 1$.
Numerator: $1^1(\ln 1 + 1)^2 + 1^{1-1} = 1(0 + 1)^2 + 1^0 = 1 + 1 = 2$.
Denominator: $\frac{1}{1^2} = 1$.
The limit is $\frac{2}{1} = 2$.

Quick Tip

When differentiating x^x, always use logarithmic differentiation. Don't be afraid to apply L'Hôpital's rule multiple times if you keep getting indeterminate forms.

---

119. Which of the following integrals is unbounded?

(A) $\int_0^{\pi/4} \tan(x)dx$
(B) $\int_0^{\infty} xe^{-x}dx$
(C) $\int_0^{a} \frac{1}{a-x}dx$
(D) $\int_0^{\infty} xe^{x}dx$

Correct Answer: (C) $\int_0^{a} \frac{1}{a-x}dx$

Solution:
Note: Both options (C) and (D) represent unbounded (divergent) integrals. However, option (C) is a more common example of an improper integral with a vertical asymptote, which might be the intended answer.

Step 1: Analyze Each Integral
(A) $\int_0^{\pi/4} \tan(x)dx$: The function $\tan(x)$ is continuous and finite on the closed interval $[0, \pi/4]$. This is a proper integral and has a finite value. It is bounded.

(B) $\int_0^{\infty} xe^{-x}dx$: This is an improper integral over an infinite interval. The function xe^{-x} approaches 0 as $x \to \infty$. This integral converges (its value is 1). It is bounded.

(C) $\int_0^{a} \frac{1}{a-x}dx$: This is an improper integral because the integrand has a vertical asymptote at the endpoint $x = a$. We must evaluate it as a limit:
$\lim_{t\to a^-} \int_0^{t} \frac{1}{a-x}dx$
$= \lim_{t\to a^-} [-\ln|a-x|]_0^t$
$= \lim_{t\to a^-} (-\ln(a-t) - (-\ln(a)))$
$= \lim_{t\to a^-} (\ln(a) - \ln(a-t))$.
As $t \to a^-$, $a - t \to 0^+$, so $\ln(a-t) \to -\infty$.
Therefore, the limit is $\ln(a) - (-\infty) = \infty$. The integral diverges, so it is unbounded.

(D) $\int_0^{\infty} xe^{x}dx$: This is an improper integral over an infinite interval. The integrand xe^x grows without bound as $x \to \infty$. The integral clearly diverges to infinity. It is unbounded.

Step 2: Conclusion
Both integrals (C) and (D) are unbounded. Since option (C) is marked as the correct answer, it is the intended choice, likely focusing on improper integrals of Type 2 (with a vertical asymptote).

Quick Tip

An integral is unbounded (diverges) if it is improper and its limiting value is $\pm\infty$. Improper integrals occur either when the interval of integration is infinite (Type 1) or when the function has a vertical asymptote within the interval (Type 2).

---

120. Let $f(x) = x^3 - \frac{9}{2}x^2 + 6x - 2$ be a function defined on the closed interval [0,3]. Then, the global maximum value of $f(x)$ is _____.

(A) 4.5
(B) 0.5
(C) 2.5
(D) 3.0

Correct Answer: (C) 2.5

Solution:

Step 1: Find the Critical Points
To find the global maximum of a continuous function on a closed interval, we must check the function's values at its critical points and at the endpoints of the interval. First, find the derivative, $f'(x)$.
$f'(x) = 3x^2 - 2\left(\frac{9}{2}\right)x + 6$
$f'(x) = 3x^2 - 9x + 6$
Set the derivative to zero to find the critical points:
$3x^2 - 9x + 6 = 0$
Divide by 3: $x^2 - 3x + 2 = 0$
Factor the quadratic: $(x-1)(x-2) = 0$
The critical points are $x = 1$ and $x = 2$. Both lie within the interval [0,3].

Step 2: Evaluate the Function at Critical Points and Endpoints
We evaluate $f(x)$ at the endpoints $x = 0, x = 3$ and the critical points $x = 1, x = 2$.
- $f(0) = (0)^3 - \frac{9}{2}(0)^2 + 6(0) - 2 = -2$
- $f(1) = (1)^3 - \frac{9}{2}(1)^2 + 6(1) - 2 = 1 - 4.5 + 6 - 2 = 0.5$
- $f(2) = (2)^3 - \frac{9}{2}(2)^2 + 6(2) - 2 = 8 - \frac{9}{2}(4) + 12 - 2 = 8 - 18 + 12 - 2 = 0$
- $f(3) = (3)^3 - \frac{9}{2}(3)^2 + 6(3) - 2 = 27 - \frac{9}{2}(9) + 18 - 2 = 43 - 40.5 = 2.5$

Step 3: Determine the Global Maximum
Compare the values we calculated: -2, 0.5, 0, and 2.5.
The largest value is 2.5, which occurs at $x = 3$.

Step 4: Final Answer:
The global maximum value of $f(x)$ on the interval [0,3] is 2.5.

Quick Tip

To find the absolute (global) maximum or minimum of a continuous function on a closed interval, always remember to test the function's value at both the critical points inside the interval and at the interval's endpoints.
