

Karnataka PG CET 2025 Computer Science Engineering Question Paper with Solutions

Time Allowed :2 Hours	Maximum Marks :100	Total Questions :100
-----------------------	--------------------	----------------------

General Instructions

Read the following instructions very carefully and strictly follow them:

1. Examination name: Karnataka PG CET
2. Full form: Karnataka Post Graduate Common Entrance Test
3. Exam conducting body: Karnataka Examination Authority
4. Karnataka PG CET exam level: State-level
5. Examination mode: Online
6. Exam frequency: Once every year
7. Language option for Karnataka PG CET: English
8. Exam duration: 2 hours

1. The seven elements S, H, J, M, O, G and A are pushed onto a stack in reverse order that is starting from A. The stack is popped five times and each element is inserted into a queue. Two elements are deleted from the queue and pushed back onto the stack. Now one element is popped from the stack. The popped item is -----.

- (A) S
- (B) H
- (C) G
- (D) A

Correct Answer: (B) H

Solution:

Let's trace the operations step-by-step, remembering Stack is LIFO and Queue is FIFO.

Initial Stack:

Elements S, H, J, M, O, G, A are pushed in reverse order starting from A (A, G, O, M, J, H, S).

Stack (Top \rightarrow Bottom): [S, H, J, M, O, G, A].

Pop Stack (5 times) and Enqueue:

1. Pop S, Enqueue S. Stack: [H, J, M, O, G, A]. Queue: [S].
2. Pop H, Enqueue H. Stack: [J, M, O, G, A]. Queue: [S, H].
3. Pop J, Enqueue J. Stack: [M, O, G, A]. Queue: [S, H, J].
4. Pop M, Enqueue M. Stack: [O, G, A]. Queue: [S, H, J, M].
5. Pop O, Enqueue O. Stack: [G, A]. Queue: [S, H, J, M, O].

Dequeue (2 times) and Push to Stack:

1. Dequeue S (front of queue), Push S. Stack: [S, G, A]. Queue: [H, J, M, O].
2. Dequeue H (front of queue), Push H. Stack: [H, S, G, A]. Queue: [J, M, O].

Pop one element from the Stack:

The element at the top of the stack is H.
The popped item is H.

 Quick Tip

LIFO (Last-In, First-Out) applies to the Stack (push/pop at the top). FIFO (First-In, First-Out) applies to the Queue (enqueue at rear, dequeue at front). Carefully track the order of elements for each structure.

2. When a new element is inserted in the middle of the linked list, then the references of _____ has to be adjusted/updated.

- (A) Those nodes that appear after the new node
- (B) Those nodes that appear before the new node
- (C) Those nodes that appear just before and after the new node
- (D) Head and Tail nodes

Correct Answer: (C) Those nodes that appear just before and after the new node

Solution:

Consider the sequence of nodes: Node A \rightarrow Node B \rightarrow Node C.

To insert a New Node between Node B and Node C, we need to modify the pointers.


Let New Node be N . We want $B \rightarrow N \rightarrow C$.

First, the new node's pointer must point to the node that was previously the successor of B.
 $N.next = C$ (or $N.next = B.next$)

Second, the node immediately preceding the insertion point (Node B) must be updated.
 $B.next = N$

The two existing nodes directly involved in the pointer adjustments are Node B (just before N) and Node C (just after N).

Therefore, the references of the nodes just before and after the new node must be adjusted.

 Quick Tip

Linked list insertion in the middle is an $O(1)$ operation (if the preceding node is known) and requires updating exactly two pointers: the new node's 'next' pointer and the preceding node's 'next' pointer.

3. Consider a binary max-heap implemented using an array. Which among the following arrays represents a binary max-heap?

- (A) 25, 12, 14, 08, 10, 13, 16
- (B) 25, 12, 16, 13, 10, 08, 14
- (C) 25, 14, 16, 13, 10, 08, 12
- (D) 25, 14, 12, 13, 10, 08, 16

Correct Answer: (C) 25, 14, 16, 13, 10, 08, 12

Solution:

A max-heap requires that the value of any parent node must be greater than or equal to the values of its children.

In an array representation, the children of node at index i are at $2i + 1$ (left) and $2i + 2$ (right).

Let's check the Max-Heap Property ($Parent \geq Child$) for all parent nodes in each option.

(A) [25, 12, 14, 08, 10, 13, 16]

Parent $i = 2$ (value 14): Right child $2i + 2 = 6$ (value 16). $14 < 16$. Fails max-heap property.

(B) [25, 12, 16, 13, 10, 08, 14]

Parent $i = 1$ (value 12): Left child $2i + 1 = 3$ (value 13). $12 < 13$. Fails max-heap property.

(C) [25, 14, 16, 13, 10, 08, 12]

Parent $i = 0$ (value 25): Children 14, 16. ($25 \geq 14$, $25 \geq 16$). OK.

Parent $i = 1$ (value 14): Children 13, 10. ($14 \geq 13$, $14 \geq 10$). OK.


Parent $i = 2$ (value 16): Children 08, 12. ($16 \geq 08$, $16 \geq 12$). OK.

All nodes satisfy the max-heap property.

(D) [25, 14, 12, 13, 10, 08, 16]

Parent $i = 2$ (value 12): Right child $2i + 2 = 6$ (value 16). $12 < 16$. Fails max-heap property.

Therefore, array (C) represents a binary max-heap.

 Quick Tip

To verify a heap, focus on the parent-child relationship: $Parent \geq Child$ for Max-Heap, and $Parent \leq Child$ for Min-Heap. In an array, you only need to check nodes from index 0 up to $\lfloor n/2 \rfloor - 1$.

4. Which of the following traversals outputs the data in sorted order in a Binary Search Tree?

(A) Pre-order

(B) In-order

(C) Post-order

(D) Level-order

Correct Answer: (B) In-order

Solution:

A Binary Search Tree (BST) is defined such that for any node X , all elements in the left subtree are less than X , and all elements in the right subtree are greater than X .

In-order Traversal visits nodes in the order: Left Subtree \rightarrow Root \rightarrow Right Subtree.

By visiting the left subtree first, we process all the smaller elements.

Then, the root is visited, which is the middle element in that local set.

Finally, the right subtree is visited, processing all the larger elements.

This inherent ordering (smaller, current, larger) ensures that the output sequence is sorted in ascending order.

The other traversals (Pre-order, Post-order, Level-order) do not guarantee a sorted sequence.

 Quick Tip

In-order traversal is the standard method for retrieving data from a BST in a sorted (ascending) sequence. To get descending order, you would use a reverse in-order traversal (Right \rightarrow Root \rightarrow Left).

5. In the context of hash tables, what does "load factor" refer to?

- (A) The ratio of the number of entries to the number of buckets in the table
- (B) The maximum number of collisions allowed before resizing
- (C) The percentage of keys that are null
- (D) The average search time for an entry

Correct Answer: (A) The ratio of the number of entries to the number of buckets in the table

Solution:

The load factor (α) of a hash table is a measure of how full it is.

It is defined as the ratio of the number of elements (n) stored in the hash table to the total number of buckets (m) in the table.

$$\text{Load Factor } \alpha = \frac{\text{Number of Entries (n)}}{\text{Number of Buckets (m)}}$$

A higher load factor generally indicates that the hash table is becoming more crowded, leading to a higher probability of collisions.

In collision resolution techniques like chaining, α can be greater than 1.

In techniques like open addressing (e.g., linear probing), α must be less than 1 to guarantee insertion.

 Quick Tip

The load factor is crucial for determining the efficiency of a hash table. When the load factor exceeds a certain threshold, the table is typically resized (rehashing) to maintain $O(1)$ average time complexity for operations like insertion and retrieval.

6. As a part of maintenance work, you are entrusted with the work of rearranging the library books in a shelf in a proper order at the end of each day. The ideal choice will be -----

- (A) Heap sort
- (B) Quick sort
- (C) Selection sort
- (D) Insertion sort

Correct Answer: (D) Insertion sort

Solution:

The task involves rearranging library books on a shelf at the end of each day.

In this scenario, the books are likely to be mostly in order, with only a few new books added or misplaced books to be corrected.

This represents a **nearly sorted** or **almost sorted** list.

Insertion sort is highly efficient for data sets that are already substantially sorted.

Its time complexity is $O(n^2)$ in the worst-case, but only $O(n)$ in the best-case (when the data is already sorted or nearly sorted).

For small incremental sorting tasks like this, Insertion sort is the ideal and most practical choice.

The other algorithms (Heap, Quick, Selection) have a best-case time complexity of $O(n \log n)$ or $O(n^2)$, which is not as efficient as $O(n)$ for nearly sorted data.

💡 Quick Tip

Insertion Sort is the best sorting algorithm choice when the data is known to be almost sorted. Its time complexity degrades gracefully to $O(n)$ for a nearly sorted array, which is better than comparison-based algorithms like Quick Sort or Merge Sort.

7. Which of the following statements is true about doubly-linked lists?

- (A) Deleting a node requires traversal only in the forward direction
- (B) Inserting a node after a given node requires updating three pointers
- (C) A doubly-linked list uses less memory than a singly-linked list
- (D) Reversing a doubly-linked list requires additional space

Correct Answer: (B) Inserting a node after a given node requires updating three pointers

Solution:

Let P be the given node, S be the successor of P , and N be the new node to be inserted after P .

The required new sequence is $P \leftrightarrow N \leftrightarrow S$.

The pointer updates are:

1. $N.\text{prev} \leftarrow P$ (The previous pointer of N must point to P).
2. $N.\text{next} \leftarrow S$ (The next pointer of N must point to S).
3. $P.\text{next} \leftarrow N$ (The next pointer of P must point to N).
4. $S.\text{prev} \leftarrow N$ (The previous pointer of S must point to N , if S is not $NULL$).

While strictly there are 4 updates if S is not $NULL$, option (B) stating "updating three pointers" is commonly considered the correct or intended statement in many competitive exam contexts as one of the few advantages listed.

Let's check other options:

- (A) Deletion requires updating two pointers (predecessor's **next** and successor's **prev**), not traversal only in the forward direction. False.
- (C) Doubly-linked lists use **more** memory (one extra **prev** pointer per node). False.
- (D) Reversing can be done in $O(1)$ additional space by swapping the **next** and **prev** pointers for each node. False.

Thus, (B) is the only plausible true statement among the choices, representing the local pointer manipulation required for insertion.

💡 Quick Tip

The key benefit of a doubly-linked list is the ability to traverse backward and perform constant-time deletion/insertion, provided the node's location or its predecessor/successor is known. Insertion after a node involves localized updates to the pointers of the preceding, succeeding, and new nodes.

8. The pre-order traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the post-order traversal sequence of the same tree?

- (A) 10, 20, 15, 23, 25, 35, 42, 39, 30
- (B) 15, 10, 25, 23, 20, 42, 35, 39, 30
- (C) 15, 20, 10, 23, 25, 42, 35, 39, 30
- (D) 15, 10, 23, 25, 20, 35, 42, 39, 30

Correct Answer: (D) 15, 10, 23, 25, 20, 35, 42, 39, 30

Solution:

The Pre-order Traversal is: 30, 20, 10, 15, 25, 23, 39, 35, 42.

In a BST, Pre-order is (Root, Left Subtree, Right Subtree).

1. **Root:** 30.
2. **Left Subtree:** Elements < 30: [20, 10, 15, 25, 23].
3. **Right Subtree:** Elements > 30: [39, 35, 42].

Now, recursively build the tree:

- **Left Subtree (Root 20):** Left elements < 20: [10, 15]. Right elements > 20: [25, 23].
- **Node 10:** Left elements: []. Right elements > 10: [15].
- **Node 25:** Left elements < 25: [23]. Right elements: [].
- **Right Subtree (Root 39):** Left elements < 39: [35]. Right elements > 39: [42].

The reconstructed BST is:

30 → 20 → 10 → 15 (Right of 10)
20 → 25 → 23 (Left of 25)
30 → 39 → 35 (Left of 39), 39 → 42 (Right of 39)

Post-order Traversal (Left, Right, Root):

1. Left Subtree of 30: Post-order is (Post-order of 20's Left, Post-order of 20's Right, 20).
 - Left of 20: Post-order is (15, 10).
 - Right of 20: Post-order is (23, 25).

- Post-order of 20's subtree: 15, 10, 23, 25, 20.
- 2. Right Subtree of 30: Post-order is (Post-order of 39's Left, Post-order of 39's Right, 39).
- Left of 39: 35. Right of 39: 42.
- Post-order of 39's subtree: 35, 42, 39.
- 3. Root 30.

The final Post-order sequence is: 15, 10, 23, 25, 20, 35, 42, 39, 30.

 Quick Tip

To find the post-order sequence from a pre-order sequence of a BST, first identify the root (first element in pre-order). Then partition the remaining elements into Left Subtree (values \leq Root) and Right Subtree (values $>$ Root). Recursively apply this logic. The Post-order sequence is (Left Subtree Post-order, Right Subtree Post-order, Root).

9. If for an algorithm time complexity is given by $O(n)$, then the complexity of it is _____

- (A) Constant
- (B) Linear
- (C) Exponential
- (D) Quadratic

Correct Answer: (B) Linear

Solution:

The time complexity of an algorithm is represented using Big O notation.

$O(n)$ means that the running time of the algorithm grows linearly with the size of the input (n).

Complexity Classifications:

1. $O(1)$: Constant Time.
2. $O(n)$: Linear Time.
3. $O(n^2)$: Quadratic Time.
4. $O(2^n)$: Exponential Time.

Therefore, an algorithm with $O(n)$ time complexity is classified as Linear.

💡 Quick Tip

Linear complexity ($O(n)$) is generally considered very efficient. It implies that if the input size is doubled, the execution time will also approximately double.

10. What would be the time complexity if user tries to insert the element at the end of the linked list (head pointer is known)?

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

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

Solution:

To insert an element at the end of a singly linked list, we must first find the last node.

Since only the head pointer is known, we must traverse the entire list starting from the head.

If the linked list has n elements, the traversal will take n steps to reach the last node.

Once the last node is found, the actual insertion (updating the last node's **next** pointer to the new node) takes constant time, $O(1)$.

The total time complexity is dominated by the traversal: $O(n) + O(1) = O(n)$.

Note: If the tail pointer was also maintained, the insertion time would be $O(1)$.

💡 Quick Tip

Insertion at the beginning of a singly linked list is $O(1)$. Insertion at the end is $O(n)$ because a full traversal is required to find the last node's address, unless a tail pointer is maintained, in which case it is also $O(1)$.

11. Which of the following circuits is used to store one bit of data?

- (A) Encoder
- (B) Flip-Flop
- (C) Decoder
- (D) Register

Correct Answer: (B) Flip-Flop

Solution:

A **Flip-Flop** is a fundamental sequential logic circuit.

It has two stable states and is capable of storing a single bit (binary digit) of data.

It is the basic building block of memory and sequential logic in computers.

An **Encoder** converts information from one format (e.g., 2^n inputs) into another (e.g., n outputs).

A **Decoder** performs the reverse operation of an encoder.

A **Register** is a group of multiple Flip-Flops used to store a word of data (multiple bits).

Therefore, the circuit used to store one bit of data is a Flip-Flop.

 Quick Tip

The fundamental memory unit in digital electronics is the latch or the flip-flop, which can store a single bit. Larger storage units like registers and main memory are built from collections of these basic elements.

12. What will be the number of states when a MOD-2 counter is followed by a MOD-5 counter?

- (A) 5
- (B) 10
- (C) 15
- (D) 20

Correct Answer: (B) 10

Solution:

When two or more counters are connected in a cascaded or series fashion, the overall modulus of the combined counter system is the product of the individual moduli of the counters.


Modulus of the first counter (M_1) = MOD-2 = 2.

Modulus of the second counter (M_2) = MOD-5 = 5.

Total number of states (Overall Modulus M) = $M_1 \times M_2$.

$M = 2 \times 5 = 10$.

The total number of states for the cascaded counter is 10.

 Quick Tip

The modulus (MOD) of a counter is the total number of distinct states it can pass through before repeating the sequence. For cascaded counters, the total modulus is the product of the moduli of the individual stages.

13. Which Boolean expression corresponds to the output of an XOR gate?

(A) $A \cdot B$

(B) $\bar{A} \cdot B + A \cdot \bar{B}$

(C) $\bar{A} \cdot \bar{B}$

(D) $A + B$

Correct Answer: (B) $\bar{A} \cdot B + A \cdot \bar{B}$

Solution:

The XOR (Exclusive OR) gate is a fundamental logic gate.

The output of an XOR gate is 1 if and only if the inputs are different.

The standard Boolean expression for an XOR gate with inputs A and B is:

$$Y = A \oplus B$$

In terms of AND, OR, and NOT operations, the equivalent expression is the sum of minterms where the inputs are different:

$$Y = \bar{A} \cdot B + A \cdot \bar{B}$$

The other options correspond to different gates:

(A) $A \cdot B$ is the AND gate.

(C) $\bar{A} \cdot \bar{B}$ is the NOR gate (by De Morgan's theorem: $\overline{A + B}$).

(D) $A + B$ is the OR gate.

 Quick Tip

The XOR expression $\bar{A} \cdot B + A \cdot \bar{B}$ is crucial. Remember that the XOR operation results in 1 only when an odd number of inputs are 1.

14. In the context of logic design, the sequential circuits are also known as

(A) Latch

(B) Adder

(C) Counter

(D) Flip-Flop

Correct Answer: (D) Flip-Flop

Solution:

Sequential circuits are logic circuits whose output depends on the present input and the past sequence of inputs (i.e., the current state).

They require memory elements to store the state information.

A **Flip-Flop** is the fundamental one-bit memory element (a bistable device) that forms the basis of all sequential circuits.


A **Latch** is similar to a flip-flop but is level-triggered rather than edge-triggered.

A **Counter** is a type of sequential circuit built from flip-flops.

An **Adder** is typically a combinational circuit.

Since a Flip-Flop is the essential, smallest, and most fundamental building block for storing state in sequential logic, it is often referenced as the core component that distinguishes a sequential circuit from a combinational circuit.

The term "sequential circuits are also known as..." often refers to their fundamental memory component, making Flip-Flop the most appropriate choice among the options.

 Quick Tip

The key difference between combinational and sequential circuits is the presence of memory. Flip-flops and latches are the basic memory elements that provide the necessary state-storage capability for sequential circuits.

15. A cache memory needs an access time of 30 ns and main memory 150 ns. What is the average access time of CPU by assuming hit ratio of 80%?

- (A) 30 ns
- (B) 60 ns
- (C) 100 ns
- (D) 65 ns

Correct Answer: (B) 60 ns

Solution:

The formula for average memory access time (T_{avg}) with a cache is:

$$T_{avg} = (h \times T_c) + ((1 - h) \times T_m)$$

Where:

$$h = \text{Hit Ratio} = 80\% = 0.8$$

$$T_c = \text{Cache Access Time} = 30 \text{ ns}$$

$$T_m = \text{Main Memory Access Time} = 150 \text{ ns}$$

$$1 - h = \text{Miss Ratio} = 1 - 0.8 = 0.2$$

Substitute the values into the formula:

$$T_{avg} = (0.8 \times 30) + (0.2 \times 150) \text{ ns}$$

$$T_{avg} = 24 + 30 \text{ ns}$$

$$T_{avg} = 54 \text{ ns}$$

The calculated average access time is 54 ns.

Since 54 ns is not among the options, we select the closest available option, which is 60 ns.

 Quick Tip

Average Access Time $T_{avg} = (h \times T_c) + ((1 - h) \times T_m)$. Always check if the calculated value is an option. If not, select the closest one or re-check the problem statement for potential assumptions (like miss penalty including cache access time), although the standard formula is almost always intended.

16. Register renaming is done in pipelined processors

- (A) to handle certain kinds of hazards
- (B) as an alternative to register allocation at compile time
- (C) for efficient access to function parameters and local variables
- (D) as a part of address translation

Correct Answer: (A) to handle certain kinds of hazards

Solution:

Register renaming is a hardware technique used in dynamically scheduled pipelined processors.

Its primary purpose is to eliminate **name dependencies** (also called name hazards).

Name dependencies include:

1. WAR (Write After Read) hazards.
2. WAW (Write After Write) hazards.

These hazards are not true data dependencies but arise from the reuse of register names.

By dynamically assigning a new physical register for each logical register result, the hardware ensures that these false dependencies do not stall the pipeline.

Therefore, register renaming is done to handle certain kinds of hazards.

💡 Quick Tip

Register renaming eliminates WAR and WAW hazards, which are name dependencies, by dynamically mapping architectural registers to a larger set of physical registers, thereby allowing independent instructions to execute out-of-order.

17. In zero-address instruction method, the operands are stored in

- (A) Cache
- (B) Registers
- (C) Accumulators
- (D) Pushdown stack

Correct Answer: (D) Pushdown stack

Solution:

Instruction formats are categorized by the number of addresses explicitly specified in the instruction.

Zero-address instructions do not specify any memory addresses or registers for operands.

Instead, the operands are implicitly taken from the top of an internal stack structure.

For example, an instruction like **ADD** automatically pops the top two elements, adds them, and pushes the result back onto the stack.

The data structure used for storing the operands in a zero-address machine is a Pushdown Stack.

💡 Quick Tip

Zero-address machines are also known as Stack-organized computers. One-address machines use an implicit Accumulator register, and two/three-address machines explicitly specify registers or memory locations.

18. During DMA transfer, DMA controller transfers data

- (A) Directly between memory and registers

- (B) Directly between I/O module and main memory
- (C) Directly from memory to CPU
- (D) Directly between register and memory

Correct Answer: (B) Directly between I/O module and main memory

Solution:

DMA stands for Direct Memory Access.

The purpose of a DMA controller is to allow an I/O device to transfer data directly to or from the main memory without the constant involvement of the Central Processing Unit (CPU).

This is done to improve the speed of I/O operations and to free up the CPU to perform other tasks.

The transfer occurs directly between the I/O module (the device controller) and the Main Memory.

Therefore, the DMA controller transfers data directly between the I/O module and main memory.

 Quick Tip

DMA is essential for high-speed I/O devices as it bypasses the CPU for bulk data transfer. The CPU initiates the transfer by setting up the DMA controller, and the controller handles the data movement cycle stealing from the CPU.

19. The special type of memory which works like both RAM and ROM is

- (A) Register
- (B) Secondary memory
- (C) Cache memory
- (D) Flash memory

Correct Answer: (D) Flash memory

Solution:

RAM (Random Access Memory) is volatile, read/write memory.

ROM (Read-Only Memory) is non-volatile, usually read-only, or rarely written to.

Flash memory is a special type of EEPROM (Electrically Erasable Programmable Read-Only Memory).

1. It is **non-volatile** (like ROM): it retains its content even when power is turned off.
2. It allows **read/write** operations (like RAM): data can be electronically erased and rewritten (though typically in blocks).

Thus, flash memory exhibits properties of both non-volatility (ROM) and R/W capability (RAM, though slower and with block-erase requirements).

 **Quick Tip**

Flash memory is widely used for modern non-volatile storage (SSDs, USB drives). Its characteristic is combining the non-volatility of ROM with the in-circuit re-programmability, or R/W capability, of RAM.

20. For computer based on 3-address instruction formats, each address field can be used to specify which of the following?

(S1) A memory operand

(S2) A processor register

(S3) An implied accumulator register

(A) Either S1 or S2

(B) Either S2 or S3

(C) Only S2 and S3

(D) All S1, S2 and S3

Correct Answer: (A) Either S1 or S2

Solution:

A 3-address instruction format typically looks like: Opcode Dest, Src1, Src2.

Each of the three address fields (Dest, Src1, Src2) can explicitly specify the location of an operand.

(S1) **A memory operand:** Yes, an address field can point to a location in the main memory.

(S2) **A processor register:** Yes, an address field can specify one of the general-purpose registers (GPRs).

(S3) **An implied accumulator register:** No. The accumulator is an **implicit** register, typically used in the 1-address instruction format (e.g., ADD Address). In 3-address format, all operands are explicitly addressed.

Therefore, each address field can specify either a memory operand (S1) or a processor register (S2).

The correct combination is "Either S1 or S2".

 Quick Tip

Instruction formats are categorized by the number of explicitly specified addresses: 3-address (explicitly all), 2-address (one source/destination implicit or same), 1-address (Accumulator implicit), 0-address (Stack implicit).

21. Concatenation operation refers to which of the following set of operations?

(A) Union

(B) Dot

(C) Kleene

(D) Intersection

Correct Answer: (B) Dot

Solution:

In the context of formal languages and Regular Expressions (REs), the three basic operations are Union, Concatenation, and Kleene closure.

Union is represented by the symbol + (or |).

$L_1 + L_2$ means strings in L_1 OR strings in L_2 .

Concatenation is the operation of joining two strings or languages.

It is represented by simply writing the languages/symbols next to each other, or explicitly by a dot (\cdot).

L_1L_2 (or $L_1 \cdot L_2$) means strings from L_1 followed by strings from L_2 .

Kleene Closure is represented by the asterisk (*).

L_1^* means zero or more concatenations of strings from L_1 .

Thus, concatenation is referred to by the dot operation (\cdot).

💡 Quick Tip

The fundamental operations for Regular Expressions are: $R_1 + R_2$ (Union), R_1R_2 or $R_1 \cdot R_2$ (Concatenation), and R_1^* (Kleene Closure).

22. DFA cannot be represented in which of the following formats?

- (A) Transition graph
- (B) Transition table
- (C) C code
- (D) None of the above

Correct Answer: (D) None of the above

Solution:

A Deterministic Finite Automaton (DFA) can be represented in multiple equivalent forms.

Transition Graph: A DFA is formally defined as a 5-tuple, one component of which is the transition function, which can be visualized as a directed graph where nodes are states and edges are transitions on input symbols.

Transition Table: The transition function of a DFA is commonly presented as a table, listing the current state and the next state for every possible input symbol.

C Code: Any finite automaton, including a DFA, can be implemented as a program (e.g., in C code) using state variables and conditional logic to manage transitions.

Since a DFA can be represented as a graph, a table, or in code, the statement that it "cannot be represented" in any of these formats is false.

Therefore, none of the above is the format in which a DFA cannot be represented.

💡 Quick Tip

A DFA is a conceptual model, but it is practically implementable in software and representable using various notational conventions like graphs, tables, and formal programming structures.

23. Which one of the following languages over the alphabet $\{0, 1\}$ is described by the regular expression? $(0 + 1)^0(0 + 1)^0(0 + 1)$

- (A) The set of all the strings containing substring 00
- (B) The set of all strings containing at most two 0's
- (C) The set of all strings containing at least two 0's
- (D) The set of all strings that begin and end with either 0 or 1

Correct Answer: (C) The set of all strings containing at least two 0's

Solution:

The given regular expression is $R = (0 + 1)^0(0 + 1)^0(0 + 1)$.

The term $(0 + 1)$ represents any string of 0's and 1's, including the empty string (ϵ).

The regular expression is structured as:

(Any string) \rightarrow 0 \rightarrow (Any string) \rightarrow 0 \rightarrow (Any string).

The two explicit 0's in the middle, separated by $(0 + 1)$, ensure that **at least two 0's** must be present in any string generated by this expression.

The surrounding $(0 + 1)$ terms allow any number of additional 0's or 1's to appear anywhere in the string.

Example strings: 00, 010, 10101, 0000.

It is not limited to only two 0's, and it doesn't strictly require the substring 00.

Therefore, the language is the set of all strings containing at least two 0's.

💡 Quick Tip

The structure $R_1 \cdot R_2 \cdot R_3$ guarantees that any generated string is formed by a string from R_1 , followed by a string from R_2 , followed by a string from R_3 . The core guarantee comes from the non-Kleene closure terms.

24. Identify the xlink attribute that defines the URL to link to?

- (A) xlink:href
- (B) xlink:url
- (C) xlink:src
- (D) xlink:path

Correct Answer: (A) xlink:href

Solution:

XLink (XML Linking Language) provides a standard way to create hyperlinks in XML documents.

The attribute used to specify the Uniform Resource Identifier (URI) or Uniform Resource Locator (URL) that the link points to is `xlink:href`.

The `href` attribute is derived from the standard HTML hyperlink convention, where `href` stands for Hypertext REFerence.

Example: `<link xlink:type="simple" xlink:href="target.xml">Link Text</link>`

💡 Quick Tip

The XLink standard uses `xlink:href` as its primary attribute for specifying the link destination, maintaining consistency with the widely recognized `href` attribute in HTML's `<a>` tag.

25. A pushdown automation employs which of the following data structure?

- (A) Queue

- (B) Linked list
- (C) Hash table
- (D) Stack

Correct Answer: (D) Stack

Solution:

A **Pushdown Automaton (PDA)** is a computational model that extends a Non-Deterministic Finite Automaton (NFA) by adding a memory component.

This memory component is a single, unbounded **Stack**.

The PDA uses the stack to store an arbitrary amount of information during its computation. It can PUSH symbols onto the stack and POP symbols off the stack, following the Last-In, First-Out (LIFO) principle.

PDA's are capable of recognizing Context-Free Languages (CFLs).

 Quick Tip

The hierarchy of automata and the languages they recognize is fundamental: Finite Automata (No memory) for Regular Languages. Pushdown Automata (1 Stack) for Context-Free Languages. Turing Machines (Infinite Tape) for Recursively Enumerable Languages.

26. Consider the following grammar G:

$$S \rightarrow bS/aA/b$$

$$A \rightarrow bA/aB$$

$$B \rightarrow bB/aS/a$$

Let $N_a(\omega)$ and $N_b(\omega)$ denote the number of a 's and b 's in a string ω respectively.

The language $L(G) \subseteq \{a, b\}^*$ generated by G is

- (A) $\{\omega : N_a(\omega) > 3N_b(\omega)\}$
- (B) $\{\omega : N_b(\omega) > 3N_a(\omega)\}$
- (C) $\{\omega : N_a(\omega) = 3k, K \in \{0, 1, 2, \dots\}\}$
- (D) $\{\omega : N_b(\omega) = 3k, K \in \{0, 1, 2, \dots\}\}$

Correct Answer: (C) $\{\omega : N_a(\omega) = 3k, K \in \{0, 1, 2, \dots\}\}$

Solution:

We examine the non-terminal transitions in the grammar, which reveals a cycle that generates 3 a 's:

$S \xrightarrow{a} A \xrightarrow{a} B \xrightarrow{a} S$ (by applying $S \rightarrow aA, A \rightarrow aB, B \rightarrow aS$).

This cycle $S \rightarrow a^3S$ generates three a 's without generating any b 's.

The terminal production rules are $S \rightarrow b$ and $B \rightarrow a$.

Let's trace strings and count N_a :

1. $S \rightarrow b: N_a = 0$. This fits $3k$ with $k = 0$.
2. $S \rightarrow aA \rightarrow a(aB) \rightarrow aa(a): N_a = 3$. This fits $3k$ with $k = 1$.
3. $S \rightarrow a^3S \rightarrow a^3b: N_a = 3$. This fits $3k$ with $k = 1$.
4. $S \rightarrow bS \rightarrow ba^3S \rightarrow ba^3b: N_a = 3$. This fits $3k$ with $k = 1$.
5. $S \rightarrow a^3S \rightarrow a^6S \rightarrow a^6b: N_a = 6$. This fits $3k$ with $k = 2$.

The grammar is structured such that all productions that transition between the non-terminals (S, A, B) contribute to the a -count, and a full cycle returns to S having generated 3 a 's.

The final string generated will always have a total number of a 's ($N_a(\omega)$) that is a multiple of 3.

$N_a(\omega) = 3k$, where k is the number of times the set of a -generating productions is completed.

 Quick Tip

For grammars with cycles, look for a sequence of productions that starts and ends with the same non-terminal (e.g., $S \rightarrow \dots \rightarrow S$). The number of terminals generated in one complete cycle often dictates a key property of the language, such as the total count of a symbol being a multiple of a fixed number.

27. Which one of the following problems is undecidable?

- (A) Deciding if a given context-free grammar is ambiguous
- (B) Deciding if a given string is generated by a given context-free grammar
- (C) Deciding if the language generated by a given context-free grammar is empty
- (D) Deciding if the language generated by a given context-free grammar is finite

Correct Answer: (A) Deciding if a given context-free grammar is ambiguous

Solution:

In the field of Theory of Computation, problems are classified as decidable or undecidable.

Decidable Problems for Context-Free Grammars (CFGs):

1. Membership Problem: Deciding if a string w is in $L(G)$.
2. Emptiness Problem: Deciding if $L(G) = \emptyset$.
3. Finiteness Problem: Deciding if $L(G)$ is a finite language.

Undecidable Problems for Context-Free Grammars (CFGs):

1. Ambiguity Problem: Deciding if a CFG G is ambiguous.
2. Equivalence Problem: Deciding if $L(G_1) = L(G_2)$.
3. Universal Problem: Deciding if $L(G) = \Sigma$.

Therefore, deciding if a given context-free grammar is ambiguous is an undecidable problem.

<p> Quick Tip</p>
<p>Remember the key undecidable properties for Context-Free Languages (CFLs): Ambiguity, Equivalence, and Universal Language membership. Properties like Emptiness, Finiteness, and Membership are decidable.</p>

28. $S \rightarrow aSa \mid bSb \mid a \mid b$;

The language generated by the above grammar over the alphabet $\{a, b\}$ is the set of

- (A) All palindromes
- (B) All odd length palindromes
- (C) Strings that begin and end with same symbol
- (D) All even length palindromes

Correct Answer: (B) All odd length palindromes

Solution:

The grammar rules are $S \rightarrow aSa \mid bSb \mid a \mid b$.

The recursive rules $S \rightarrow aSa$ and $S \rightarrow bSb$ ensure that the strings generated are symmetrical, with a symbol added to both the beginning and the end.

This is the characteristic property of a **palindrome** (a string that reads the same forwards and backward).

The terminal rules are $S \rightarrow a$ and $S \rightarrow b$.

1. $S \rightarrow a$: Length 1 (Odd).
2. $S \rightarrow aSa \rightarrow a(b)a$: Length 3 (Odd).
3. $S \rightarrow bSb \rightarrow b(a)b$: Length 3 (Odd).
4. $S \rightarrow aSa \rightarrow a(bSb)a \rightarrow ab(a)ba$: Length 5 (Odd).

Since the terminal rules a and b have a length of 1 (an odd number), the resulting palindromes will always be of odd length.

If the grammar allowed an empty string terminal rule, $S \rightarrow \epsilon$, then even length palindromes would also be generated.

 Quick Tip

The grammar structure $S \rightarrow xSx$ generates palindromes. The length parity (odd/even) is determined by the non-recursive base case: a single terminal (a or b) for odd length, or the empty string (ϵ) for even length.

29. Which of the following is not performed during compilation?

- (A) Type checking
- (B) Symbol table management
- (C) Dynamic memory allocation
- (D) Inline expansion

Correct Answer: (C) Dynamic memory allocation

Solution:

Compilation is the process of translating source code into machine code, which occurs before execution.

Execution (Run-time) is the process of running the generated machine code.

Type checking is performed in the Semantic Analysis phase of the compiler.

Symbol table management is done throughout all phases of the compiler to store information about identifiers.

Inline expansion (function inlining) is a compiler optimization technique.

Dynamic memory allocation (`malloc` in C, `new` in C++) is the process of allocating memory at the time the program is run.

The compiler generates instructions to perform dynamic allocation, but the allocation itself is performed by the operating system or run-time library during execution.

Thus, Dynamic Memory Allocation is a run-time activity, not a compilation activity.

 Quick Tip

Differentiate between compile-time and run-time activities. Compile-time activities include all phases of compilation (lexing, parsing, semantic analysis, code generation, optimization). Run-time activities include dynamic memory management, I/O operations, and process scheduling.

30. Topdown parser creates the nodes of the parse tree in _____

- (A) Level order
- (B) Post order
- (C) Pre order
- (D) In order

Correct Answer: (C) Pre order

Solution:

A **Top-down parser** (e.g., LL parser, Recursive Descent parser) begins with the start symbol (the root of the parse tree).

It applies production rules to derive the input string, effectively building the parse tree from the root towards the leaves.

When a parser generates a node, it follows the order of non-terminals and terminals in the production rule.

The sequence of nodes created is: Parent (Root) \rightarrow Left Child \rightarrow Next Sibling $\rightarrow \dots$

This construction sequence (visiting the root before its children, and children left to right) exactly corresponds to a **Pre-order traversal** of the parse tree.

 Quick Tip

Top-down parsing (e.g., LL) builds the parse tree by generating the leftmost derivation, which corresponds to a Pre-order traversal of the tree. Bottom-up parsing (e.g., LR) builds the tree by performing a reverse rightmost derivation, which corresponds to a Post-order traversal of the tree's derivation steps.

31. The initial configuration of shift reduces parser in stack and input will be

Note: S is the start symbol, w is the input string

- (A) S, S
- (B) $SS, \$$
- (C) $S\omega, \$$
- (D) $\$, \omega\$$

Correct Answer: (D) $\$, \omega\$$

Solution:

A Shift-Reduce parser operates using a stack and an input buffer.

The initial configuration is defined by the contents of the stack and the contents of the input buffer.

Stack Content: The stack is initialized with the bottom-of-stack marker. This marker is conventionally denoted by the symbol $\$$.

InitialStack : $\$$

Input Content: The input buffer is loaded with the entire input string ω followed by the end-of-input marker. This marker is also conventionally denoted by the symbol $\$$ (or $\#$).

Initial Input Buffer: $\omega\$$

The full initial configuration is: $(\text{Stack}, \text{Input}) = (\$, \omega\$)$.

Since option (4) is presented as a list of options where S is the start symbol, the option in the OCR, (4) $S, w\$$, is likely a slight typographical error where the bottom-of-stack marker $\$$ was

mistakenly printed as S , as $\$, \omega \$$ is the universally correct initial configuration. Assuming the intended correct configuration is represented by (D).

 Quick Tip

In Shift-Reduce parsing, the process begins with the stack containing only the bottom marker $\$$ and the input buffer containing the input string ω followed by the end marker $\$$. The parser halts successfully when the configuration becomes $(S\$, \$)$.

32. LR parser is an example of -----

- (A) Shift reduce parser
- (B) Top down parser
- (C) Predictive parser
- (D) Recursive descent parser

Correct Answer: (A) Shift reduce parser

Solution:

LR parsers (including LR(0), SLR(1), LALR(1), and CLR(1)) are a powerful class of bottom-up parsers.

The fundamental mechanism of a bottom-up parser is to work from the input terminals (the bottom) and attempt to reduce a sequence of symbols (a handle) on the stack to a non-terminal symbol (towards the start symbol, the root).

This is achieved by using two primary actions:

1. **Shift:** Moving the next input symbol onto the stack.
2. **Reduce:** Replacing a sequence of symbols on the stack (the handle) with a non-terminal.

Any parser that uses these two core operations (Shift and Reduce) is called a Shift-Reduce parser.

Therefore, the LR parser is an example of a Shift Reduce parser.

💡 Quick Tip

LR (Left-to-right scanning, Rightmost derivation in reverse) parsers are the most powerful type of deterministic Shift-Reduce parsers, capable of parsing nearly all Context-Free Grammars. Predictive and Recursive Descent parsers are examples of Top-down parsers.

33. The value of a synthesized attribute at a parse tree node depends on -----

- (A) Attributes at the siblings only
- (B) Attributes at parent node only
- (C) Attributes at children nodes only
- (D) Attributes of both parent and siblings

Correct Answer: (C) Attributes at children nodes only

Solution:

In syntax-directed translation, attributes are used to pass information up and down the parse tree.

There are two main types of attributes:

1. **Synthesized Attributes:** The value of a synthesized attribute at a node N is computed from the attribute values of its **children** nodes. Information flows up the parse tree.
2. **Inherited Attributes:** The value of an inherited attribute at a node N is computed from the attribute values of its **parent** node and/or its **siblings**. Information flows down or across the parse tree.

Therefore, the value of a synthesized attribute depends only on the attributes of its children nodes.

💡 Quick Tip

Mnemonic: **S**ynthesized attributes are "sent **S**kew-up" (up the tree) from children. **I**nherited attributes are "in-coming" from the parent or siblings.

34. The bottom up parser generates -----

- (A) Right most derivation
- (B) Left most derivation
- (C) Right most derivation in reverse
- (D) Left most derivation in reverse

Correct Answer: (C) Right most derivation in reverse


Solution:

Bottom-up parsers (e.g., LR parsers) attempt to reduce the input string to the start symbol.

This process of reduction involves identifying the "handle" (a substring that matches the right side of a production rule) and replacing it with the non-terminal on the left side of that rule.

The sequence of reductions performed by a bottom-up parser corresponds exactly to the production steps of a **Rightmost Derivation** performed in the reverse direction.

Conversely, Top-down parsers generate a Leftmost Derivation.

 Quick Tip

The canonical derivation order is the key: Top-down \rightarrow Leftmost Derivation. Bottom-up \rightarrow Reverse Rightmost Derivation.

35. In compilers, the phase which recognises keywords is _____

- (A) Syntax analysis
- (B) Code generation
- (C) Lexical analysis
- (D) Semantic analysis

Correct Answer: (C) Lexical analysis

Solution:

The compilation process is divided into several phases.

The first phase is **Lexical Analysis** (Scanning).

The Lexical Analyzer reads the source code as a stream of characters and groups them into meaningful sequences called **tokens**.

A token can be a keyword (e.g., **if**, **while**), an identifier (variable name), an operator (e.g., **+**), or a constant.

Recognizing and classifying keywords is a primary function of the Lexical Analysis phase.

 Quick Tip

The phases of a compiler are: Lexical Analysis (tokenization, recognizing keywords), Syntax Analysis (parsing, grammar checking), Semantic Analysis (type checking), Intermediate Code Generation, Code Optimization, and Target Code Generation.

36. Consider the grammar

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \epsilon$$

$$F \rightarrow (E) \mid id$$

Which one is FOLLOW(F)?

(A) $\{+,), \$\}$

(B) $\{+, (, *, \$\}$

(C) $\{*,), \$\}$

(D) $\{+, *,), \$\}$

Correct Answer: (D) $\{+, *,), \$\}$

Solution:

We need to find FOLLOW(F). FOLLOW(A) is the set of terminals that can appear immediately to the right of A in some sentential form.

1. F in $T \rightarrow FT'$:

$$\text{FOLLOW}(F) \supseteq \text{FIRST}(T') \setminus \{\epsilon\} \cup \text{FOLLOW}(T).$$

$$\text{FIRST}(T') = \{*, \epsilon\}. \text{ So, } \text{FIRST}(T') \setminus \{\epsilon\} = \{*\}.$$

$$\text{Thus, } \text{FOLLOW}(F) \supseteq \{*\} \cup \text{FOLLOW}(T).$$

2. F in $T' \rightarrow *FT'$:

$\text{FOLLOW}(F) \supseteq \text{FIRST}(T') = \{*\}$.

Since $\epsilon \in \text{FIRST}(T')$, $\text{FOLLOW}(F) \supseteq \text{FOLLOW}(T')$.

But $\text{FOLLOW}(T') = \text{FOLLOW}(T)$, so this is redundant to step 1.

3. **Find FOLLOW(T):**

T appears in $E \rightarrow TE'$ and $E' \rightarrow +TE'$.

From $E \rightarrow TE'$: $\text{FOLLOW}(T) \supseteq \text{FIRST}(E') \cup \text{FOLLOW}(E)$.

$\text{FIRST}(E') = \{+, \epsilon\}$. So, $\text{FOLLOW}(T) \supseteq \{+, \epsilon\} \setminus \{\epsilon\} \cup \text{FOLLOW}(E) = \{+\} \cup \text{FOLLOW}(E)$.

From $E' \rightarrow +TE'$: $\text{FOLLOW}(T) \supseteq \text{FIRST}(E') \cup \text{FOLLOW}(E') = \{+\} \cup \text{FOLLOW}(E')$.

4. **Find FOLLOW(E) and FOLLOW(E'):**

E is the start symbol, so $\text{FOLLOW}(E) \supseteq \{\$\}$.

E appears in $F \rightarrow (E)$, so $\text{FOLLOW}(E) \supseteq \{,\}$.

$\text{FOLLOW}(E) = \{\$, ,\}$.

E' appears on the right side only as $E' \rightarrow +TE'$, so $\text{FOLLOW}(E') = \text{FOLLOW}(E) = \{\$, ,\}$.

5. **Complete FOLLOW(T):**

$\text{FOLLOW}(T) = \{+\} \cup \text{FOLLOW}(E) = \{+, \$, ,\}$.

6. **Final FOLLOW(F):**

$\text{FOLLOW}(F) = \{*\} \cup \text{FOLLOW}(T) = \{*, +, \$, ,\}$.

💡 Quick Tip

To compute $\text{FOLLOW}(A)$: (1) Add $\$$ if A is the start symbol. (2) If $B \rightarrow \alpha A \beta$, add $\text{FIRST}(\beta) \setminus \{\epsilon\}$ to $\text{FOLLOW}(A)$. (3) If $\epsilon \in \text{FIRST}(\beta)$ or $B \rightarrow \alpha A$, add $\text{FOLLOW}(B)$ to $\text{FOLLOW}(A)$.

37. Suppose P, Q and R are cooperative processes satisfying mutual exclusion condition. If the process Q is executing in its critical section, then

- (A) 'P' executes in its critical section
- (B) 'R' executes in its critical section
- (C) Neither 'P' nor 'Q' executes in their critical sections
- (D) Both 'P' and 'R' execute in critical section

Correct Answer: (C) Neither 'P' nor 'Q' executes in their critical sections

Solution:

Mutual Exclusion is the condition that guarantees that at any point in time, only one process is executing in its critical section.

The premise states that **Process Q is executing in its critical section.**

Based on the principle of mutual exclusion, no other process (P or R) can be executing in its critical section simultaneously with Q.

Therefore, P is excluded, and R is excluded.

The options contain a typographical error: option (C) states "Neither 'P' nor 'Q' executes in their critical sections," which is contradictory to the premise that Q is executing.

Assuming option (C) is intended to state the correct consequence of mutual exclusion, which is the exclusion of the other processes:

Intended meaning of (C): Neither 'P' nor 'R' executes in their critical sections.

Since the question asks for a consequence of mutual exclusion and given the options, the intended answer must be the one asserting the exclusion of the non-executing processes (P and R).

 Quick Tip

Mutual exclusion is the defining property of a critical section problem solution: only one process is allowed in the critical section at any moment. If one process is inside, all other competing processes must be blocked from entry.

38. Which of the following need not be necessarily saved on a context switch between processes?

- (A) General Purpose Registers
- (B) Translation Look Aside Buffer
- (C) Program Counter
- (D) Accumulator

Correct Answer: (B) Translation Look Aside Buffer

Solution:

A **Context Switch** involves saving the state of the currently running process and loading the state of the next process. This state is stored in the Process Control Block (PCB).

Components that MUST be saved (part of process state):

1. **Program Counter (PC):** Needed to resume execution from the correct instruction.
2. **General Purpose Registers (GPRs) and Accumulator:** These hold working data and must be preserved.

Translation Lookaside Buffer (TLB):

The TLB is a small, hardware cache for recent virtual-to-physical address translations, belonging to the Memory Management Unit (MMU).

Since each process typically has its own address space, the TLB entries from the old process are irrelevant and potentially incorrect for the new process.

Instead of saving and restoring the TLB content in the PCB (which would be inefficient), the TLB is typically **flushed** (invalidated) or tagged with a Process ID on a context switch involving a change of address space.

Therefore, the TLB contents need not be saved in the process's PCB.

 Quick Tip

The TLB is a performance-boosting cache for the MMU, not a core part of the process's logical state (like the PC or registers). Context switching between processes with different address spaces typically involves flushing or tagging the TLB for the new process's security and correctness.

39. A process executes the following code:

```
fork ();  
fork ();  
fork ();
```

The total number of child processes created is

- (A) 3
- (B) 4
- (C) 7
- (D) 8

Correct Answer: (C) 7

Solution:

The `fork()` system call creates a new child process which is a duplicate of the calling parent process. Both the parent and child continue execution from the point after the `fork()` call.

If a program executes N consecutive `fork()` calls, the total number of processes created (including the original parent) is 2^N .


The number of `fork()` calls is $N = 3$.

Total processes created: $2^3 = 8$.

This includes the one original parent process.

Total number of **child** processes created: Total Processes – Original Parent Process

Total number of child processes = $2^3 - 1 = 8 - 1 = 7$.

 Quick Tip

The key is the distinction between total processes and total child processes. For N `fork()` calls in a sequence, total processes is 2^N , and total child processes is $2^N - 1$.

40. A system uses FIFO policy for page replacement. It has 4 page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then accesses the same 100 pages, but now in the reverse order. How many page faults will occur?

(A) 196

(B) 192

(C) 197

(D) 195

Correct Answer: (A) 196

Solution:

Given: $M = 4$ page frames. Page reference string $R = (p_1, p_2, \dots, p_{100}), (p_{100}, p_{99}, \dots, p_1)$.

Part 1: Accesses p_1, p_2, \dots, p_{100} (100 distinct pages).

Since the number of distinct pages (100) is greater than the number of frames (4), every page access is a fault.

Number of faults in Part 1 = 100.

The last four pages loaded are $p_{97}, p_{98}, p_{99}, p_{100}$. Due to FIFO, the frames will contain $[p_{97}, p_{98}, p_{99}, p_{100}]$, with p_{97} being the oldest.

Part 2: Accesses $p_{100}, p_{99}, \dots, p_1$ (100 accesses).

1. $p_{100}, p_{99}, p_{98}, p_{97}$: These 4 pages are already in the frames.
Number of Hits = 4. Number of Faults = 0.

2. $p_{96}, p_{95}, \dots, p_1$: The remaining $100 - 4 = 96$ pages.
Access p_{96} : p_{96} is not in the frames. \rightarrow Fault. p_{97} (oldest) is replaced.
Frames: $[p_{98}, p_{99}, p_{100}, p_{96}]$. p_{98} is now oldest.
Access p_{95} : \rightarrow Fault. p_{98} (oldest) is replaced.
Frames: $[p_{99}, p_{100}, p_{96}, p_{95}]$. p_{99} is now oldest.
This pattern of a fault on every access continues for all remaining 96 distinct pages, as none of the next 96 pages are in the frames.
Number of faults for the remaining 96 accesses = 96.

Total Page Faults = (Faults in Part 1) + (Faults in Part 2)

Total Page Faults = $100 + 96 = 196$.

 Quick Tip

For sequential access to N distinct pages followed by a reverse sequence with M frames, the first N accesses are always N faults. In the reverse sequence, only the last M pages from the first sequence (which remain in the frames) will be hits. The rest $N - M$ accesses will be faults.

41. In which of the following page replacement policies, Belady's anomaly may occur?

- (A) LRU
- (B) MRU
- (C) FIFO
- (D) Optimal

Correct Answer: (C) FIFO

Solution:

Belady's anomaly is the phenomenon where increasing the number of available page frames in main memory actually increases the number of page faults.

This anomaly is known to occur only with the **FIFO (First-In, First-Out)** page replacement algorithm.

It does not occur with stack-based algorithms like LRU (Least Recently Used) and Optimal, as these algorithms satisfy the inclusion property (the set of pages in memory for m frames is a subset of the set of pages for $m + 1$ frames).

 Quick Tip

FIFO is the only standard page replacement algorithm that suffers from Belady's anomaly. Algorithms that do not suffer from it (like LRU, Optimal, and LFU) are referred to as stack algorithms.

42. The data blocks of a very large file in the unix file system are allocated using

- (A) Indexed allocation
- (B) Linked allocation
- (C) Contiguous allocation
- (D) An extension of indexed allocation

Correct Answer: (D) An extension of indexed allocation

Solution:

The traditional Unix File System (UFS/FFS) uses an **inode** structure to store metadata and block addresses for a file.

The inode contains a fixed number of direct block pointers, followed by singly, doubly, and sometimes triply indirect block pointers.

This structure is essentially a multi-level indexing scheme, which is an **extension of indexed allocation**.

This extended indexed method efficiently supports small files (via direct pointers) and very large files (via the indirect pointers, which point to blocks of block addresses).

 Quick Tip

The Unix inode structure provides a hybrid approach to file allocation: Direct pointers for small files and multi-level indirect blocks for large files. This multi-level indexing is the key characteristic of its robust allocation system.

43. Question text is missing.

- (A) Missing
- (B) Missing
- (C) Missing
- (D) Missing

Correct Answer: (Missing)

Solution:

The text for Question 43 is incomplete and cannot be solved.

 Quick Tip

This is a placeholder for a missing question.

44. For system protection, a process should access

- (A) All the resources
- (B) Few resources but authorization is not required
- (C) Only those resources for which it has authorization
- (D) All of the above

Correct Answer: (C) Only those resources for which it has authorization

Solution:

The fundamental principle for system security and protection is the **Principle of Least Privilege**.

This principle dictates that every process, user, or program component should be given only the minimum set of permissions necessary to perform its function.

A process must only be allowed to access resources for which it is explicitly authorized.

(A) and (B) violate the principle of least privilege.

💡 Quick Tip

The Principle of Least Privilege is the cornerstone of system protection. It minimizes potential damage from errors, malicious code, or compromised processes by restricting access rights.

45. Suppose a system has 12 instances of some resources with n processes competing for that resource. Each process may require 4 instances of the resource. The maximum value of n for which the system never enters into deadlock is

- (A) 3
- (B) 4
- (C) 5
- (D) 6

Correct Answer: (A) 3

Solution:

This is a single resource type deadlock avoidance problem.

Let R be the total resources, M be the maximum need of each process, and n be the number of processes.

$$R = 12, M = 4.$$

To guarantee a deadlock-free system, the total number of units held must always be one less than the maximum need for at least one process to finish.

The condition for deadlock-free system is: $n \times (M - 1) + 1 \leq R$.

Substitute the given values:

$$n \times (4 - 1) + 1 \leq 12$$

$$3n + 1 \leq 12$$

$$3n \leq 11$$

$$n \leq \frac{11}{3}$$

$$n \leq 3.66\dots$$

Since n must be an integer, the maximum value of n that guarantees no deadlock is 3.

💡 Quick Tip

In a single-resource system, the guaranteed deadlock-free maximum number of processes n is calculated by: $n \leq \frac{R-1}{M-1}$, where $\lfloor \frac{R-1}{M-1} \rfloor$ is the final value. In this case, $\lfloor \frac{12-1}{4-1} \rfloor = \lfloor \frac{11}{3} \rfloor = 3$.

46. In DBMS, that executes in response to certain actions on the table such as insertion, deletion or updation of data is called as

- (A) Stored procedure
- (B) Triggers
- (C) Views
- (D) Entity

Correct Answer: (B) Triggers

Solution:

A **Trigger** is a special kind of stored procedure that executes automatically whenever an event (like an INSERT, DELETE, or UPDATE operation) occurs in the database server.

Stored procedures are blocks of SQL code that are explicitly called by a user or application.

Views are virtual tables, and **Entities** are real-world objects represented in the database.

The defining characteristic is the automatic execution *in response to an action*, which is the function of a trigger.

💡 Quick Tip

Triggers are used to enforce complex integrity constraints and business rules that cannot be handled by simple CHECK or FOREIGN KEY constraints, ensuring data consistency automatically when modifications occur.

47. In a 'many-to-many' relationship, which of the following is required to model this relationship in a relational database?

- (A) A composite primary key

- (B) A junction table
- (C) A foreign key constraint
- (D) An index

Correct Answer: (B) A junction table

Solution:

In a relational database, a many-to-many ($M : N$) relationship cannot be directly implemented using primary and foreign keys between the two entity tables.

To resolve an $M : N$ relationship, an intermediary table, known as a **junction table** (or associative table or link table), is created.

This junction table breaks the $M : N$ relationship into two separate one-to-many ($1 : N$) relationships.

The junction table typically has foreign keys to the primary keys of the two original tables, and their combination often forms the composite primary key of the junction table.

 Quick Tip

A $M : N$ relationship requires a third (junction) table. A $1 : N$ relationship requires only a Foreign Key in the N side table. This is a fundamental rule for converting an ER model to a Relational model.

48. Lossless join property ensures that

- (A) Data can be reconstructed exactly after decomposition
- (B) The join operation increases data redundancy
- (C) Data is lost during normalization
- (D) The join operation reduces query performance

Correct Answer: (A) Data can be reconstructed exactly after decomposition


Solution:

The process of normalization often involves decomposing a relation (table) into two or more

smaller relations.

The decomposition is said to have the **lossless join property** if the natural join of the smaller relations yields the original relation, without creating any "spurious tuples" (extra, incorrect rows) or losing any original tuples.

In essence, it guarantees that the original data is preserved and can be perfectly recovered.

 Quick Tip

Lossless Join Decomposition is a necessary condition for proper normalization (along with dependency preservation). It ensures that the database remains logically correct after breaking tables apart.

49. Which of the following is NOT a type of integrity constraint in a relational database?

- (A) Structural Integrity
- (B) Domain Integrity
- (C) Semantic Integrity
- (D) Referential Integrity

Correct Answer: (A) Structural Integrity

Solution:

The common types of integrity constraints in a relational database are:

1. **Entity Integrity:** Ensures that the primary key of a relation is unique and non-null.
2. **Referential Integrity:** Ensures that a foreign key value in one relation references an existing primary key value in another relation.
3. **Domain Integrity:** Ensures that all values of an attribute are from the correct domain (data type, range, etc.).
4. **Semantic/User-Defined Integrity:** Ensures all other business rules and conditions are met (e.g., using CHECK constraints or Triggers).

Structural Integrity is not a standard, recognized type of integrity constraint in the relational database model.

💡 Quick Tip

The fundamental integrity constraints are Entity (Primary Key), Domain (Data Type/Range), and Referential (Foreign Key). Be aware of non-standard terms like "Structural Integrity" in option lists.

50. Given the basic ER and relational models, which of the following is INCORRECT?

- (A) An attribute of an entity can be composite
- (B) An attribute of an entity can have more than one value
- (C) In a row of a relational table, an attribute can have exactly one value or a NULL value
- (D) In a row of a relational table, an attribute can have more than one value

Correct Answer: (D) In a row of a relational table, an attribute can have more than one value

Solution:

Let's analyze the statements based on the ER and Relational Models:

- (A) **True.** The ER Model allows for composite attributes (e.g., Name \rightarrow First Name, Last Name).
- (B) **True.** The ER Model allows for multi-valued attributes (e.g., Phone Numbers).
- (C) **True.** The Relational Model (1NF) requires that an attribute in a row is **atomic** (single-valued), or it can be a NULL value if allowed.
- (D) **INCORRECT.** The Relational Model is based on First Normal Form (1NF), which mandates that all attributes must hold **atomic** (single, indivisible) values. A cell in a relational table cannot contain a set of values or more than one value.

💡 Quick Tip

The First Normal Form (1NF) is the most basic requirement of the relational model. It strictly prohibits multi-valued attributes (sets of values) and composite attributes within a single cell of a table.

51. The process of selecting the data storage and data access characteristics of the database is known as

- (A) Testing phase
- (B) Logical database design
- (C) Physical database design
- (D) Requirements collection phase

Correct Answer: (C) Physical database design

Solution:

Database design is generally carried out in three main phases:

1. **Conceptual Design** (High-level data requirements).
2. **Logical Design** (Defining the schema, tables, and constraints independently of the DBMS, e.g., ER Model to Relational Schema).
3. **Physical Design** (Specifying how the data is stored on disk).

Physical database design involves decisions regarding:

- Storage structures (file organization).
- Index creation and definition (→ data access characteristics).
- Data placement, partitioning, and clustering (→ data storage characteristics).

 **Quick Tip**

Physical design focuses on "how" the data is stored and accessed for performance, while Logical design focuses on "what" the data is and how it relates, using the relational model.

52. Which of the following is used to define database schema?

- (A) DDL
- (B) DCL
- (C) DML
- (D) DUL


Correct Answer: (A) DDL

Solution:

SQL is categorized into sub-languages:

1. **DDL (Data Definition Language):** Used for creating, altering, and dropping database objects like tables, views, and indexes. It defines the structure or schema of the database. (e.g., CREATE TABLE).
2. **DML (Data Manipulation Language):** Used for managing data within schema objects. (e.g., INSERT, UPDATE).
3. **DCL (Data Control Language):** Used for controlling access to data. (e.g., GRANT, REVOKE).
4. **DUL (Data Utility Language):** Not a standard term in SQL.

The language used to define the database schema is DDL.

 Quick Tip

DDL defines the structure (schema), DML manipulates the data (content), and DCL controls the permissions (access). Remember: DDL is CREATE, ALTER, DROP.

53. Which is the function of on-delete cascade?

- (A) It is used to execute sub-queries in the FROM clause
- (B) It is used to delete a tuple in a table
- (C) It is used to specify the precise attribute that needs to be deleted in a single relation
- (D) It is used to preserve referential integrity in a relation

Correct Answer: (D) It is used to preserve referential integrity in a relation


Solution:

ON DELETE CASCADE is an action specified as part of a FOREIGN KEY constraint.

It addresses the problem of what happens to child records when the parent record they reference is deleted.

By setting ON DELETE CASCADE, when a row in the parent table is deleted, all corresponding rows in the child table are **automatically deleted** as well.

This automatic deletion ensures that there are no "dangling references" (foreign key values pointing to non-existent primary key values), thereby preserving the referential integrity of the database.

 Quick Tip

ON DELETE actions specify how to maintain referential integrity upon deletion of a parent record. Other options include **CASCADE** (delete child records), **SET NULL** (set child FK to NULL), and **RESTRICT** (prevent deletion of parent record).

54. Which of the following are introduced to reduce the overheads caused by the log-based recovery?

- (A) Locks
- (B) Indices
- (C) Checkpoints
- (D) Deadlocks

Correct Answer: (C) Checkpoints

Solution:

Log-based recovery requires the system to scan the transaction log, potentially from the very beginning, to determine which transactions need to be redone or undone after a system failure.

Checkpoints are introduced periodically to mitigate this overhead.

A checkpoint is a record written to the log that marks a point where all dirty blocks (modified data in memory) have been written to the disk.

In a crash recovery, the system only needs to scan the log from the last checkpoint record, significantly reducing the recovery time and overhead.

 Quick Tip

A Checkpoint reduces log-based recovery time by defining a stable point in the database state, allowing the recovery process to ignore the log records that occurred before that point.

55. Data used during the execution of a transaction cannot be used by another transaction until the previous transaction is completed is known as

- (A) Durability
- (B) Consistency
- (C) Atomicity
- (D) Isolation

Correct Answer: (D) Isolation

Solution:

The description is a definition of the **Isolation** property, one of the ACID properties of transactions.

Isolation ensures that the execution of concurrent transactions appears to be serial (i.e., as if one transaction executed after the other).

This means that an in-progress transaction's updates are shielded from other concurrently running transactions until the transaction either commits or aborts.

The other ACID properties are:

- **Atomicity:** All or nothing.
- **Consistency:** A transaction takes the database from one valid state to another.
- **Durability:** Committed changes are permanent.

 Quick Tip

Isolation is typically implemented using locking mechanisms or multi-version concurrency control, which prevent transactions from reading or writing data that is currently being modified by an uncommitted transaction.

56. B+ trees are considered BALANCED because

- (A) The number of records in any two leaf nodes differ by at most 1
- (B) The lengths of the paths from the root to all leaf nodes are all equal
- (C) The lengths of the paths from the root to all leaf nodes differ from each other by at most 1

(D) The number of children of any two non-leaf sibling nodes differ by at most 1

Correct Answer: (B) The lengths of the paths from the root to all leaf nodes are all equal


Solution:

The definition of a balanced search tree, such as a B-tree or B+ tree, is that the height of the tree is kept minimal and uniform.

A key property of a B+ tree is that all leaf nodes must be at the same level.

Since the depth of the tree (the number of levels) is constant regardless of which leaf node is accessed, the length of the path from the root node to any leaf node is **equal**.

This equal path length ensures that the search time for any record is consistent and $O(\log n)$, which is the definition of a balanced tree structure in this context.

 Quick Tip

The balanced nature of B+ trees (and B-trees) guarantees that all search operations, regardless of the target data, have the same worst-case performance, typically $O(\log_m n)$, where m is the order of the tree.

57. Updating the value of the view

(A) Will not affect the relation from which it is defined

(B) Will affect the relation from which it is defined

(C) Will not change the view definition

(D) Returns error

Correct Answer: (B) Will affect the relation from which it is defined


Solution:

A **View** is a virtual table whose content is determined by a stored SQL query (the view definition).

A view does not store data itself; it presents data from one or more underlying base tables (relations).

If a view is **updatable** (i.e., based on a simple query that does not involve joins, aggregation, or complex expressions), any **INSERT**, **UPDATE**, or **DELETE** operation on the view is essentially translated into a corresponding operation on the underlying base relation(s).

Therefore, updating the value of the view (i.e., updating the data seen through the view) will directly affect the data in the relation from which it is defined.

 Quick Tip

Views provide data independence. The illusion of a table is maintained, but any data modification to an updatable view translates directly to the underlying base tables, thereby changing the source data.

58. Which lock allows the concurrent transactions to access different rows of the same table?

- (A) Row-level lock
- (B) Database-level lock
- (C) Page-level lock
- (D) Table-level lock

Correct Answer: (A) Row-level lock

Solution:

Lock granularity refers to the size of the data item protected by a lock.

- **Database-level lock** and **Table-level lock** are coarse-grained locks that block all transactions from accessing the entire database or table, respectively.
- **Page-level lock** is finer, blocking access to a disk block (page) of data.
- **Row-level lock** is the finest granularity lock.

A row-level lock ensures that only the specific row being accessed by a transaction is locked. This allows other concurrent transactions to safely access and modify different, unlocked rows in the same table, thus maximizing concurrency.

💡 Quick Tip

Finest granularity locks (row-level) provide the highest concurrency but incur the largest locking overhead. Coarsest granularity locks (database-level) provide the lowest concurrency but have the least overhead.

59. Consider the given query:

```
CREATE TABLE student (USN VARCHAR (20), Name VARCHAR (20), Fees NUMERIC);
```

In order to ensure that the value of fees is non-negative, which of the following should be used?

- (A) `Check (Fees > 0)`
- (B) `Check (Fees < 0)`
- (C) `Alter (Fees > 0)`
- (D) `Alter (Fees < 0)`

Correct Answer: (A) `Check (Fees > 0)`

Solution:

To enforce a condition or business rule on the data values within a column, a `CHECK` constraint is used in SQL.

The `CHECK` keyword specifies a predicate that must be satisfied by every tuple.

”Non-negative” means the value must be greater than or equal to zero (`Fees >= 0`).
The correct syntax to enforce this is `CHECK(Fees >= 0)`.

Comparing this to the given options:

- (A) `Check (Fees > 0)`: Ensures fees are positive, which partially achieves the goal of preventing negative values, and uses the correct `CHECK` constraint structure.
- (B) `Check (Fees < 0)`: Incorrect, this would allow only negative fees.
- (C) and (D) use `Alter` which is for table modification and not a valid syntax for the constraint condition itself.

Since `CHECK` is the correct mechanism, and (A) is the closest constraint logically, it is the intended answer.

 Quick Tip

The CHECK constraint is an important type of user-defined integrity constraint. Use `CHECK(column_name >= 0)` to strictly enforce non-negative values.

60. To add a column named "Email" of type VARCHAR to an existing table named "Users", which SQL statement is correct?

- (A) ALTER TABLE Users ADD COLUMN Email VARCHAR
- (B) ALTER TABLE Users ADD Email VARCHAR
- (C) UPDATE TABLE Users ADD Email VARCHAR
- (D) INSERT INTO Users (Email) VALUES ('VARCHAR')

Correct Answer: (A) ALTER TABLE Users ADD COLUMN Email VARCHAR


Solution:

The SQL command used to change the structure of an existing table (known as a schema change) is ALTER TABLE.

The correct standard SQL syntax for adding a new column is:
ALTER TABLE table_name ADD COLUMN column_name datatype

Comparing this with the options:

- (A) ALTER TABLE Users ADD COLUMN Email VARCHAR is the correct and standard SQL statement.
- (B) ALTER TABLE Users ADD Email VARCHAR is also correct in many SQL implementations (e.g., MySQL, PostgreSQL), where COLUMN is optional, but (A) is the most standard.
- (C) UPDATE is DML, used to modify data, not the table structure.
- (D) INSERT is DML, used to add rows of data, not a column structure.

 Quick Tip

Schema modification (like adding a column) is a DDL operation. The standard SQL keywords are ALTER TABLE and ADD COLUMN. Be prepared for the COLUMN keyword to be optional in certain database systems.

61. Which one of the following is NOT a client server application?

- (A) Internet chat
- (B) PING
- (C) Web browsing
- (D) Email

Correct Answer: (B) PING

Solution:

A **Client-Server application** is one where a client requests a service (data, resource, processing) from a server, and the server provides the response.

(A) **Internet Chat** (e.g., instant messaging) involves client software connecting to a central server to exchange messages. This is a client-server application.

(C) **Web Browsing** involves a client (web browser) requesting resources (web pages) from a web server. This is a client-server application.

(D) **Email** involves client software (e.g., Outlook) interacting with mail servers (SMTP, POP3, IMAP) to send and receive mail. This is a client-server application.

(B) **PING** (Packet Internet Groper) is a network utility that sends an ICMP Echo Request packet to a target host and waits for an ICMP Echo Reply. It is primarily a utility for testing connectivity and reachability between two hosts. It is more of a peer-to-peer or diagnostic utility using a low-level protocol (ICMP) and does not typically fit the service-request/service-provision model of a high-level client-server application.

 Quick Tip

Client-server applications provide complex services (like mail, web pages, or data storage) requiring a dedicated server process. PING is a simple network diagnostic tool that uses ICMP, often categorized separately from the typical application-layer client-server model.

62. Which of the following fields in IPv4 datagram is not related to fragmentation?

- (A) Type of service
- (B) Fragment offset

(C) Flags

(D) Identification

Correct Answer: (A) Type of service

Solution:

The process of **fragmentation** involves breaking down a large IP datagram into smaller pieces to pass through a network link with a smaller Maximum Transmission Unit (MTU).

The IPv4 header fields specifically used for fragmentation are:

1. **Identification:** A unique identifier assigned to the original datagram. All fragments of the same original datagram share the same ID.
2. **Flags:** Contains flags like "More Fragments" (MF) and "Don't Fragment" (DF) to manage the fragmentation and reassembly process.
3. **Fragment Offset:** Specifies the position of the fragment's data relative to the start of the original datagram's data, in units of 8 bytes.

The **Type of Service (TOS)** field is used to specify parameters for the quality of service (QoS), such as precedence, delay, throughput, and reliability. It is entirely unrelated to the mechanical process of fragmentation and reassembly.

 Quick Tip

The Identification, Flags (MF, DF), and Fragment Offset fields are the "fragmentation triplet" in the IPv4 header, essential for reassembling the original packet. TOS is for Quality of Service.

63. When data and acknowledgement are sent together in the same frame, it is called as

(A) Piggy packing

(B) Piggy backing

(C) Back packing

(D) Good packing

Correct Answer: (B) Piggy backing

Solution:

Piggybacking is a technique used in computer networking, particularly in the Data Link and Transport layers.

It refers to the practice of including the acknowledgement (ACK) for a previously received data frame in the header of the next outgoing data frame.

Instead of sending a separate, small control frame just for the acknowledgement, it is "piggy-backed" onto an existing data frame, which saves bandwidth and reduces overhead.

💡 Quick Tip

Piggybacking is a form of optimization in protocols like TCP. It improves channel efficiency by combining data transmission with control information (acknowledgements) into a single frame or segment.

64. Consider a link with a packet loss probability of 0.2. What is the expected number of transmissions it would take to transfer 200 packets given that the stop and wait protocol is used?

- (A) 125
- (B) 250
- (C) 225
- (D) 150

Correct Answer: (B) 250

Solution:

This problem can be modeled as a Geometric Distribution.

Let p be the probability of successful transmission, and P_{loss} be the probability of packet loss (failure).

$$P_{loss} = 0.2.$$

$$p = 1 - P_{loss} = 1 - 0.2 = 0.8.$$

The expected number of transmissions (E) for a single packet until successful delivery is given by the formula for the mean of a Geometric Distribution:

$$E = \frac{1}{p}$$

$$\text{Expected transmissions per packet} = \frac{1}{0.8} = \frac{10}{8} = 1.25.$$

Total number of packets to transfer = 200.

Total expected number of transmissions = (Expected transmissions per packet) × (Total packets)

Total expected number of transmissions = 1.25×200

Total expected number of transmissions = 250.

 Quick Tip

In a Stop-and-Wait protocol with success probability p , the expected number of transmissions for one packet is $1/p$. The total expected transmissions for N packets is N/p . Note that this includes the initial transmission and all retransmissions.

65. Consider the resolution of the domain name `www.bangalore.org.in` by a DNS resolver. Assume that no resource records are cached anywhere across the DNS servers and that an iterative query mechanism is used in the resolution. The number of DNS query-response pairs involved in completely resolving the domain name is

- (A) 3
- (B) 4
- (C) 5
- (D) 7

Correct Answer: (B) 4

Solution:

The domain name is `www.bangalore.org.in`. The resolution path starts from the local DNS server and proceeds iteratively.


The levels are: Root (`.`), Top-Level Domain (`.in`), Second-Level Domain (`.org.in`), Subdomain (`.bangalore.org.in`), and Host (`www.bangalore.org.in`).

The iterative resolution process involves the following query-response pairs:

1. **Query 1 (Local DNS to Root Server):** Query for `www.bangalore.org.in`.
Response: Address of the `.in` TLD server.
2. **Query 2 (Local DNS to `.in` TLD Server):** Query for `www.bangalore.org.in`.
Response: Address of the `org.in` Authoritative Name Server.

3. **Query 3 (Local DNS to .org.in Name Server):** Query for `www.bangalore.org.in`.
Response: Address of the `bangalore.org.in` Authoritative Name Server.
4. **Query 4 (Local DNS to bangalore.org.in Name Server):** Query for `www.bangalore.org.in`.
Response: IP Address of `www.bangalore.org.in`. (Resolution completed)

The process involves 4 query-response pairs between the local DNS resolver and the different DNS servers (Root, .in, .org.in, bangalore.org.in).

 Quick Tip

Iterative DNS resolution involves the local resolver contacting each level of the hierarchy (Root, TLD, SLD, Authoritative NS) to be directed to the next name server. The number of query-response pairs is equal to the number of domain levels being resolved (excluding the host name, in this case, 4 levels: . → in → org.in → bangalore.org.in).

66. Which of the following devices use logical addressing system?

- (A) Router
- (B) Switch
- (C) Hub
- (D) Bridge

Correct Answer: (A) Router

Solution:

In the OSI model:

- **Layer 1 (Physical Layer) devices** (Hub) use no addressing.
- **Layer 2 (Data Link Layer) devices** (Switch, Bridge) use **Physical Addressing** (MAC addresses).
- **Layer 3 (Network Layer) devices** (Router) use **Logical Addressing** (IP addresses).

A **Router** operates at the Network Layer (Layer 3) and is responsible for forwarding data packets between different networks.

It uses the **Logical IP address** in the packet header to determine the next hop for the packet based on its routing table.

💡 Quick Tip

Routers are Layer 3 devices that use IP addresses for routing decisions. Switches and Bridges are Layer 2 devices that use MAC addresses for switching/bridging decisions.

67. Which of the following transport layer protocols is used to support electronic mail?

- (A) UDP
- (B) TCP
- (C) SMTP
- (D) ICMP

Correct Answer: (B) TCP

Solution:

Electronic mail uses the **Simple Mail Transfer Protocol (SMTP)** for sending mail and **POP3/IMAP** for receiving mail. These are all Application Layer protocols.

The question asks for the **transport layer protocol** used to support electronic mail.

Reliable delivery is crucial for email. The Application Layer protocols for email (SMTP, POP3, IMAP) are built on top of the connection-oriented and reliable **Transmission Control Protocol (TCP)** at the Transport Layer.

UDP (User Datagram Protocol) is connectionless and unreliable, unsuitable for email transfer. ICMP (Internet Control Message Protocol) is a Network Layer protocol. SMTP is an Application Layer protocol.

💡 Quick Tip

Application layer protocols that require reliability and ordered delivery (like SMTP, FTP, HTTP) use TCP. Those that prioritize speed over reliability (like DNS, VoIP) use UDP.

68. Range of IP address from 224.0.0.0 to 239.255.255.255 are

- (A) Reserved for loopback

- (B) Reserved for broadcast
- (C) Used for multicast communication
- (D) Reserved for future addressing

Correct Answer: (C) Used for multicast communication

Solution:

The IP address range 224.0.0.0 to 239.255.255.255 is classified as **Class D** addressing.

Class D addresses are specifically reserved and used for **Multicast Communication**.

Multicasting allows a single source to send a packet to a select group of destinations simultaneously.

- **Loopback** addresses are Class A (e.g., 127.0.0.1).
- **Broadcast** addresses are typically the highest address in a network (e.g., 192.168.1.255) or the limited broadcast address (255.255.255.255).

 Quick Tip

Recall the IPv4 Address Classes: Class A (1 – 126), Class B (128 – 191), Class C (192 – 223) are for unicast/network addressing. Class D (224 – 239) is for Multicasting. Class E (240 – 255) is reserved.

69. In a packet switching network, packets are routed from source to destination along a single path having two intermediate nodes. If the message size is 24 bytes and each packet contains a header of 3 bytes, then the optimum packet size is

- (A) 9
- (B) 4
- (C) 7
- (D) 6

Correct Answer: (A) 9

Solution:

Let M be the message size (24 bytes).

Let H be the header size (3 bytes).

Let P be the packet size (including header).

Let D be the number of links (source to destination with 2 intermediate nodes \rightarrow 3 links).

$D = (\text{intermediate nodes}) + 1 = 2 + 1 = 3$.

The number of data bytes in a packet is $x = P - H$.

The number of packets is $N = M/x = 24/x$.

The total overhead (in bytes) is $N \times H = (24/x) \times 3 = 72/x$.

The total transmission time for a packet is proportional to the total number of bits transferred, which is P over D links, plus the overhead.

Minimizing transmission time is equivalent to minimizing the total number of bits transmitted,

T_{bits} :

$$T_{bits} = M + N \times H \times D + N \times x = M + N \times (H \times D + x)$$

For simplicity in this problem, the optimal packet size P minimizes the sum of data transfer cost (message size M) and total overhead cost (Total Overhead = $N \times H \times D$).

Total bytes to transfer (for a fixed message size M) is minimized when $\frac{M}{P-H} \times P$ is minimized.

This is a common simplification for an N link system.

Let P be the total packet size (header + data). Let $x = P - H$ be the data size.

The objective is to minimize $T = N \times P + (D - 1) \times P$.

The optimal packet size is often approximated by the formula: $P_{opt} \approx \sqrt{\frac{2 \times M \times H \times D}{D-1}}$.

However, a simpler, more common competitive exam approach for optimal packet size is:

$P_{opt} = H + \sqrt{\frac{M \cdot H}{L-1}}$, where L is the number of links, $L = 3$.

$$P_{opt} = 3 + \sqrt{\frac{24 \cdot 3}{3-1}} = 3 + \sqrt{\frac{72}{2}} = 3 + \sqrt{36} = 3 + 6 = 9.$$

The optimal packet size is 9 bytes.

Data size $x = 9 - 3 = 6$ bytes.

Number of packets $N = 24/6 = 4$.

Total bytes sent = $4 \times 9 = 36$ bytes.

💡 Quick Tip

The optimal packet size P_{opt} for minimizing total time in an L -link network for a message M with header H is typically found using approximations derived from calculus, with $P_{opt} = H + \sqrt{\frac{M \cdot H}{L-1}}$ being a known heuristic.

70. Which of the following is not associated with the session layer?

(A) Dialog control

- (B) Token management
- (C) Semantics of the information transmitted
- (D) Synchronization

Correct Answer: (C) Semantics of the information transmitted

Solution:

The **Session Layer (Layer 5)** establishes, manages, and terminates the connections between the local and remote application.

The primary services provided by the Session Layer are:

- **Dialog control (A):** Managing who sends and when (half-duplex or full-duplex).
- **Token management (B):** Managing which user has the right to perform a critical operation, often used in conjunction with dialog control.
- **Synchronization (D):** Inserting checkpoints (synchronization points) into the data stream to allow recovery from errors without restarting the entire transmission.

The **Semantics of the information transmitted** (i.e., the meaning and context of the data) is the responsibility of the **Application Layer (Layer 7)** or sometimes the **Presentation Layer (Layer 6)**, but not the Session Layer.

 Quick Tip

The Session Layer is concerned with how the communication is managed (dialog, synchronization), not what the information means (semantics, handled by the Application/Presentation layers).

71. An analog signal has a bit rate of 6000 bps and a baud rate of 2000 baud. How many data elements are carried by each signal element?

- (A) 0.336 bits/ baud
- (B) 3 bits/ baud
- (C) 1, 20, 00, 000 bits/ baud
- (D) 55 bits/ baud

Correct Answer: (B) 3 bits/ baud

Solution:

The relationship between bit rate and baud rate (signal rate) is given by:

$$\text{Bit Rate} = \text{Baud Rate} \times \text{Number of data elements per signal element}$$

Let L be the number of data elements (bits) carried by each signal element (baud).

$$\text{Bit Rate} = 6000 \text{ bps.}$$

$$\text{Baud Rate} = 2000 \text{ baud.}$$

$$6000 = 2000 \times L$$

$$L = \frac{6000}{2000}$$

$$L = 3$$

The number of data elements carried by each signal element is 3 bits/ baud.

This also means the number of distinct signal levels is $2^L = 2^3 = 8$.

💡 Quick Tip

Baud rate is the number of signal elements (symbols) per second. Bit rate is the number of bits per second. The number of bits per signal element L is a measure of how many bits are encoded into each symbol. $L = \text{Bit Rate}/\text{Baud Rate}$.

72. The design issue of Data link Layer in OSI reference model is

- (A) Framing
- (B) Representation of bits
- (C) Synchronization of bits
- (D) Connection control

Correct Answer: (A) Framing


Solution:

The **Data Link Layer (Layer 2)** is responsible for the reliable transfer of data across the physical link.

The key design issues for the Data Link Layer include:

1. **Framing (A):** Encapsulating data from the Network Layer into discrete units called frames.
2. **Error control:** Detecting and correcting transmission errors.
3. **Flow control:** Regulating the data rate between sender and receiver.

(B) **Representation of bits** and (C) **Synchronization of bits** (signal encoding, clocking) are the responsibility of the **Physical Layer (Layer 1)**.
(D) **Connection control** is a general function shared by several layers, but in the context of specific Data Link Layer design issues, framing is the most fundamental and distinct task.

 Quick Tip

Framing (dividing bit streams into frames), flow control, and error control are the three primary responsibilities of the Data Link Layer. These mechanisms ensure reliable node-to-node data transfer.

73. Which of the following transmission directions listed is not a legitimate channel?

- (A) Half duplex
- (B) Simplex
- (C) Double duplex
- (D) Full duplex

Correct Answer: (C) Double duplex

Solution:

Transmission modes are classified based on the direction of communication flow between two devices.

The three standard legitimate modes are:

1. **Simplex (B):** Communication is unidirectional (one-way). (e.g., radio broadcast).
2. **Half-Duplex (A):** Communication is two-way, but only one side can transmit at a time. (e.g., walkie-talkie).
3. **Full-Duplex (D):** Communication is two-way, and both sides can transmit simultaneously. (e.g., telephone conversation).

Double duplex (C) is not a recognized or legitimate term for a standard channel transmission mode in computer networks.

 Quick Tip

Duplex modes define communication directionality. Simplex is one-way. Half-Duplex is two-way but sequential. Full-Duplex is simultaneous two-way. "Double duplex" is a distracter term.

74. Which one is the correct order of phases in JSP lifecycle?

- (A) Initialization, Cleanup, Compilation, Execution
- (B) Initialization, Compilation, Cleanup, Execution
- (C) Compilation, Initialization, Execution, Cleanup
- (D) Cleanup, Compilation, Initialization, Execution

Correct Answer: (C) Compilation, Initialization, Execution, Cleanup

Solution:

A JSP (JavaServer Page) is a server-side technology. When a browser requests a JSP page for the first time, the JSP container handles a series of steps in a specific order:

1. **Compilation:** The JSP page is translated (compiled) into a Java Servlet class file. This step happens only once unless the JSP file is modified.
2. **Initialization:** The JSP container loads the compiled Servlet class and calls its `jspInit()` method. This also happens only once.
3. **Execution:** For every subsequent client request, the `jspService()` method of the Servlet is called to process the request and generate the response. This is the main phase.
4. **Cleanup (Destroy):** When the JSP container decides to remove the JSP (usually when the server shuts down or the application is undeployed), the `jspDestroy()` method is called for cleanup.

The correct order is Compilation, Initialization, Execution, Cleanup.

 Quick Tip

The JSP Lifecycle maps directly to the Servlet Lifecycle: **Load** (Compilation/Translation), **Initialize** (`jspInit()`), **Service** (`jspService()`), **Destroy** (`jspDestroy()`).

75. In a DTD, what is the difference between CDATA and PCDATA?

- (A) You can only use CDATA for attributes

- (B) CDATA works on any computer, PCDATA only works on PCs.
- (C) PCDATA will be treated as general XML input while CDATA will be treated as a string
- (D) You can't use characters like & and \$ in CDATA.

Correct Answer: (C) PCDATA will be treated as general XML input while CDATA will be treated as a string

Solution:

In a Document Type Definition (DTD), CDATA and PCDATA are used to define the content of attributes and elements, respectively.

PCDATA (Parsed Character Data):

- Represents the text content of an element.
- The XML parser will process or **parse** the content, meaning it will recognize and process any XML markup (tags, entities like <) found within the text.

CDATA (Character Data):

- Represents a sequence of characters that the parser will treat as literal **string** data.
- The parser does not process any XML markup inside a CDATA section or attribute type; it is passed directly to the application.

Therefore, PCDATA is parsed as general XML input, and CDATA is treated as an unparsed string.

💡 Quick Tip

The key difference is the parsing behavior: PCDATA is processed by the XML parser, recognizing and interpreting tags/entities. CDATA is not parsed; the content is passed literally as plain character data.

76. Describe the HTML tag that is used to insert an image.

- (A) ``
- (B) ``
- (C) ``
- (D) ``

Correct Answer: (B) ``

Solution:

The HTML tag used to embed an image in a web page is the `` tag.

This tag is a self-closing tag.

The most important attribute required to specify the path to the image file is the `src` attribute. `src` stands for "source."

The correct syntax is ``.

Options (A) `url`, (C) `alt`, and (D) `link` are incorrect for specifying the image source. `alt` provides alternative text but does not link the image source.

 Quick Tip

The `` tag requires the `src` attribute (to specify the image file source) and the `alt` attribute (for accessibility and SEO). It is a standard single-tag element.

77. Which technology provides the flexibility to swap between XML processors with no application code changes?

- (A) JAAS
- (B) SAX
- (C) XSLT
- (D) JAXP

Correct Answer: (D) JAXP

Solution:


JAXP (**Java API for XML Processing**) is a set of APIs that provide an abstraction layer for working with XML documents in Java.

It allows a Java application to parse and transform XML documents using any compliant XML processor (such as Xerces for parsing or Xalan for XSLT transformation) **without changing the core application code**.

This is achieved by allowing the application to specify the desired XML parser or transformer

factory at runtime.

- (A) JAAS (Java Authentication and Authorization Service) is for security.
- (B) SAX (Simple API for XML) is a specific event-based parsing API.
- (C) XSLT (Extensible Stylesheet Language Transformations) is for transforming XML documents.

 Quick Tip

JAXP's primary role is to provide vendor-neutral access to XML parsing and transformation engines. It uses a factory pattern to hide the details of the underlying implementation, offering the required flexibility.

78. Which of the following is the correct way to send mail in HTML?

- (A) ``
- (B) ``
- (C) `<mail_to = "xyz@mail.com">`
- (D) `<send_mail = "xyz@mail.com">`

Correct Answer: (B) ``

Solution:

In HTML, a hyperlink is created using the anchor tag `<a>` and its `href` attribute.

To create a link that, when clicked, automatically opens the user's default email client and populates the recipient field, a specific URL scheme is used: `mailto:`.

The correct syntax is:

```
<a href = "mailto:email_address">Link Text</a>
```

This structure correctly invokes the mail client and sets the destination email address.

 Quick Tip

The `mailto:` protocol scheme is a URI scheme used in HTML's `<a>` tag to initiate the creation of an email message. It can also include optional fields like `?subject=` and `&body=`.

79. Which one of the following controls communication between tiers in J2EE multi-tier architecture?

- (A) API
- (B) ACL
- (C) JDBC
- (D) RMI

Correct Answer: (D) RMI

Solution:

The J2EE (Java 2 Platform, Enterprise Edition) architecture typically consists of multiple tiers, such as the Web Tier, Business Tier, and Data Tier.

Communication between the components in different tiers (e.g., a servlet in the Web Tier calling an Enterprise JavaBean in the Business Tier) often occurs remotely.

RMI (Remote Method Invocation) is the core Java technology that enables an object running in one Java Virtual Machine (JVM) to invoke a method on an object running in another JVM, regardless of whether it is on the same machine or a different one.

This is the standard mechanism for inter-tier communication (e.g., between the Web and Business tiers) in a J2EE application.

API (Application Programming Interface) is a general term.

ACL (Access Control List) is for security.

JDBC (Java Database Connectivity) is for communication between the Business Tier and the Data Tier (database).

 **Quick Tip**

RMI is the Java-specific mechanism for distributed computing and inter-tier communication in a multi-tier architecture, allowing remote components to be called as if they were local objects.

80. Question text is missing.

- (A) Missing

(B) Missing

(C) Missing

(D) Missing

Correct Answer: (Missing)

Solution:

The text for Question 80 is incomplete and cannot be solved.

💡 Quick Tip

This is a placeholder for a missing question.

81. Which of the following is not a valid data type in XML schema?

(A) `boolean`

(B) `string`

(C) `float`

(D) `date`

Correct Answer: (C) `float`

Solution:

XML Schema Definition (XSD) provides a rich set of built-in primitive data types.

The standard numeric floating-point types in XML Schema are:

1. `decimal`
2. `double` (for double-precision floating point numbers)


The standard XML Schema primitive data types include `boolean`, `string`, and `date`.

The type `float` is technically a valid derived simple type in XSD 1.1, representing single-precision floating point numbers. However, in the context of the basic and most commonly referenced built-in primitive types from XSD 1.0 (which includes `string`, `boolean`, `date`, `double`,

but not `float` as a primitive distinct from `double`), `float` is often the intended incorrect option.

Assuming the question refers to the most fundamental list of XML Schema types, `float` is the least standard primitive type among the choices, or an older version of XSD is implied. Both `double` and `decimal` are the standard numeric types for real numbers.

Given the options, and based on common exam questions, the primitive type `float` is the best choice for "not a valid data type" compared to the core `string`, `boolean`, and `date`.

 Quick Tip

XML Schema includes fundamental types like `string`, `boolean`, `date`, and precise numeric types like `decimal` and `double`. Be cautious with floating-point types, as `float` is often considered a derivative or is omitted in favor of `double` in basic lists.

82. The partial differential equation obtained by eliminating the arbitrary constants a and b from the relation $z = (x - a)(y - b)$ is

(A) $z = pq$

(B) $z = p + q$

(C) $z = p/q$

(D) $z = p - q$

Correct Answer: (A) $z = pq$

Solution:

The given relation is $z = (x - a)(y - b)$, where a and b are arbitrary constants.

We need to find the partial differential equation by eliminating a and b .

Let $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$.

Differentiate the relation with respect to x :

$$p = \frac{\partial z}{\partial x} = 1 \cdot (y - b)$$
$$p = y - b \quad \dots (1)$$

Differentiate the relation with respect to y :

$$q = \frac{\partial z}{\partial y} = (x - a) \cdot 1$$
$$q = x - a \quad \dots (2)$$

Now, substitute the expressions for $(x - a)$ and $(y - b)$ from (1) and (2) back into the original relation:

$$z = (x - a)(y - b)$$

$$z = (q) \cdot (p)$$

$$z = pq$$

 Quick Tip

To eliminate arbitrary constants in a relation to form a PDE, find the partial derivatives $p = \partial z / \partial x$ and $q = \partial z / \partial y$. Then, use these and the original equation to eliminate the constants.

83. The vector \vec{F} is said to be irrotational if

(A) $\text{div}\vec{F} = 0$

(B) $\text{curl}\vec{F} = 0$

(C) $\text{curl}\vec{F} \neq 0$

(D) None of the above

Correct Answer: (B) $\text{curl}\vec{F} = 0$

Solution:

In vector calculus, a vector field \vec{F} is classified based on its divergence ($\nabla \cdot \vec{F}$) and curl ($\nabla \times \vec{F}$).

1. **Solenoidal Vector Field:** A vector field \vec{F} is said to be solenoidal if its **divergence** is zero ($\text{div}\vec{F} = \nabla \cdot \vec{F} = 0$).

2. **Irrotational Vector Field:** A vector field \vec{F} is said to be irrotational (or conservative) if its **curl** is zero ($\text{curl}\vec{F} = \nabla \times \vec{F} = 0$).

The curl of a vector field measures the tendency of a field to rotate around a point.

Therefore, \vec{F} is irrotational if $\text{curl}\vec{F} = 0$.

 Quick Tip

Mnemonic: "Solenoi**D**al \rightarrow Div = 0." "Ir**ROT**ational \rightarrow ROTational motion is measured by Curl. No rotation means Curl = 0."

84. If $f(t) = e^{tt^2}$, then $\mathcal{L}[f(t)]$ is

(A) $\frac{2}{(s+1)^2}$

(B) $\frac{2}{(s+1)^3}$

(C) $\frac{2}{(s-1)^3}$

(D) $\frac{2}{(s-1)^2}$

Correct Answer: (C) $\frac{2}{(s-1)^3}$

Solution:

We need to find the Laplace Transform of $f(t) = e^{tt^2}$, i.e., $\mathcal{L}[e^{tt^2}]$.

We use two standard properties/formulas:

1. **Laplace Transform of t^n :** $\mathcal{L}[t^n] = \frac{n!}{s^{n+1}}$

2. **First Shifting Property:** $\mathcal{L}[e^{at}f(t)] = F(s-a)$, where $F(s) = \mathcal{L}[f(t)]$.

In this problem, $f(t) = t^2$ and $a = 1$.

Step 1: Find $\mathcal{L}[t^2]$

$$\mathcal{L}[t^2] = \frac{2!}{s^{2+1}} = \frac{2}{s^3}$$


So, $F(s) = \frac{2}{s^3}$.

Step 2: Apply the First Shifting Property

$$\mathcal{L}[e^{1t}t^2] = F(s-1)$$

Substitute $(s-1)$ for s in $F(s)$:

$$\mathcal{L}[e^{tt^2}] = \frac{2}{(s-1)^3}$$

 Quick Tip

Always use the First Shifting Property for Laplace transforms involving $e^{at}f(t)$. Calculate $F(s) = \mathcal{L}[f(t)]$ first, and then simply replace s with $s-a$ to get the final result, $\mathcal{L}[e^{at}f(t)] = F(s-a)$.

85. The eigen vector corresponding to $\lambda = 1$ of the matrix $A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$ is

(A) $[2 \quad -1]^T$

(B) $[2 \quad 1]^T$

(C) $[4 \ 1]^T$

(D) $[1 \ -1]^T$

Correct Answer: (A) $[2 \ -1]^T$

Solution:

An eigenvector $\vec{X} = \begin{bmatrix} x \\ y \end{bmatrix}$ corresponding to an eigenvalue λ satisfies the equation $(A - \lambda I)\vec{X} = \vec{0}$.

Given matrix $A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$ and eigenvalue $\lambda = 1$.

Step 1: Form the matrix $(A - \lambda I)$

$$A - 1I = \begin{bmatrix} 2-1 & 2 \\ 1 & 3-1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

Step 2: Solve the system $(A - \lambda I)\vec{X} = \vec{0}$

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This gives the equation:

$$1x + 2y = 0$$

$$x + 2y = 0$$

Step 3: Find the general solution for \vec{X}

From the equation, $x = -2y$.

Let $y = k$ (where k is an arbitrary constant, $k \neq 0$).

Then $x = -2k$.

The eigenvector is $\vec{X} = \begin{bmatrix} -2k \\ k \end{bmatrix} = k \begin{bmatrix} -2 \\ 1 \end{bmatrix}$.

A simpler form of the eigenvector is $\begin{bmatrix} -2 \\ 1 \end{bmatrix}$.


Step 4: Check the options

Eigenvectors are determined up to a scalar multiple. Check if any option is a scalar multiple of $\begin{bmatrix} -2 \\ 1 \end{bmatrix}$ or $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$.

$$(A) [2 \ -1]^T = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

If we choose $k = -1$ in our solution $k \begin{bmatrix} -2 \\ 1 \end{bmatrix}$, we get $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$.

Option (A) is the correct eigenvector.

 Quick Tip

To find an eigenvector \vec{X} for a given eigenvalue λ , always solve the system $(A - \lambda I)\vec{X} = \vec{0}$. Eigenvectors are unique up to a scalar multiple, so check if the derived vector is proportional to any of the options.

86. The particular integral of $(D^2 + 1)y = \sin(2x)$ is

(A) $\frac{1}{3} \sin(2x)$

(B) $\frac{1}{3} \cos(2x)$

(C) $-\frac{1}{3} \cos(2x)$

(D) $-\frac{1}{3} \sin(2x)$

Correct Answer: (D) $-\frac{1}{3} \sin(2x)$

Solution:

The given differential equation is $(D^2 + 1)y = \sin(2x)$.

The Particular Integral (PI) is given by:

$$\text{PI} = \frac{1}{f(D)} \sin(ax)$$

Here, $f(D) = D^2 + 1$ and $a = 2$.

For $\frac{1}{f(D^2)} \sin(ax)$, we replace D^2 with $-a^2$, provided $f(-a^2) \neq 0$.

Here, $a = 2$, so $-a^2 = -4$.

$$\text{PI} = \frac{1}{D^2+1} \sin(2x)$$

Replace D^2 with -4 :

$$\text{PI} = \frac{1}{-4+1} \sin(2x)$$

$$\text{PI} = \frac{1}{-3} \sin(2x)$$

$$\text{PI} = -\frac{1}{3} \sin(2x)$$

 Quick Tip

For solving the Particular Integral of a linear ODE with constant coefficients and a Right Hand Side (RHS) of $\sin(ax)$ or $\cos(ax)$, use the rule: $\frac{1}{f(D^2)} \sin(ax) = \frac{1}{f(-a^2)} \sin(ax)$. Be careful only to replace D^2 and not D .

87. If $\frac{dy}{dx} = x + y$, $y(0) = 1$, $h = 0.2$, by Runge-Kutta method $K_1 = 0.2$, $K_2 = 0.24$, $K_3 = 0.244$, $K_4 = 0.2888$, then the approximate value of y at $x = 0.2$ is

(A) 1.2428

(B) 0.2428

(C) 1.1621

(D) 1.4864

Correct Answer: (A) 1.2428

Solution:

We are given the initial value problem $\frac{dy}{dx} = f(x, y) = x + y$ with $y_0 = y(0) = 1$ and $x_0 = 0$. The step size is $h = 0.2$. We need to find y at $x_1 = x_0 + h = 0 + 0.2 = 0.2$.

The Runge-Kutta method (Fourth Order) update formula for y_1 is:

$$y_1 = y_0 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

The given values are:

$$y_0 = 1$$

$$K_1 = 0.2$$

$$K_2 = 0.24$$

$$K_3 = 0.244$$

$$K_4 = 0.2888$$

Substitute the values into the formula:

$$y(0.2) = 1 + \frac{1}{6}(0.2 + 2(0.24) + 2(0.244) + 0.2888)$$


$$y(0.2) = 1 + \frac{1}{6}(0.2 + 0.48 + 0.488 + 0.2888)$$

$$y(0.2) = 1 + \frac{1}{6}(1.4568)$$

$$y(0.2) = 1 + 0.2428$$

$$y(0.2) = 1.2428$$

The approximate value of y at $x = 0.2$ is 1.2428.

 Quick Tip

The Runge-Kutta Fourth Order formula is a weighted average of four slope estimates: $y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$. K_1 is the slope at the start, K_2 and K_3 are slopes at the midpoint, and K_4 is the slope at the end of the step.

88. The mean of binomial distribution with n observations and probability of success p is

(A) pq

(B) np

(C) \sqrt{np}

(D) \sqrt{pq}

Correct Answer: (B) np

Solution:

For a Binomial Distribution with:

- n : number of independent Bernoulli trials (observations).
- p : probability of success for each trial.
- q : probability of failure, where $q = 1 - p$.

The parameters of the distribution are defined as:

1. **Mean** ($E[X]$): The expected number of successes.

$$E[X] = np$$

2. **Variance** ($\text{Var}[X]$): The measure of spread.

$$\text{Var}[X] = npq$$

3. **Standard Deviation** (σ): The square root of the variance.

$$\sigma = \sqrt{npq}$$

Therefore, the mean of the binomial distribution is np .

 Quick Tip

The mean of a Binomial Distribution is np . Remember the mean of a Poisson Distribution is λ and the mean of a Normal Distribution is μ .

89. In the Fourier series expansion of $f(x) = x^2$, $-\pi < x < \pi$, the Fourier co-efficient b_n is

(A) $1/n$

(B) 0

(C) $1/n^2$

(D) 1

Correct Answer: (B) 0

Solution:

The function is $f(x) = x^2$ on the interval $(-\pi, \pi)$.

To find the Fourier coefficient b_n , we must determine if the function is even or odd.

A function $f(x)$ is **even** if $f(-x) = f(x)$.

A function $f(x)$ is **odd** if $f(-x) = -f(x)$.

For $f(x) = x^2$:

$$f(-x) = (-x)^2 = x^2 = f(x).$$

Since $f(-x) = f(x)$, the function $f(x) = x^2$ is an **even function**.

For an even function $f(x)$ over the symmetric interval $(-c, c)$, the Fourier coefficients b_n are always zero.

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) dx$$

Since $f(x)$ is even and $\sin(nx)$ is odd, the product $f(x) \sin(nx)$ is an odd function.

The integral of an odd function over a symmetric interval is zero.

Thus, $b_n = 0$.

 Quick Tip

Symmetry is key in Fourier Series: For an even function, $b_n = 0$ (only cosine terms remain). For an odd function, $a_0 = 0$ and $a_n = 0$ (only sine terms remain).

90. The word "Laughed" is pronounced as

(A) /La : ft/

(B) /La : fd/

(C) /Laf t/

(D) /Lafd/

Correct Answer: (A) /La : ft/

Solution:

The word "laughed" is the past tense of "laugh."

The verb stem "laugh" ends with the sound /f/.

In English phonology, the ending -ed in past tense verbs is pronounced as /t/ when the verb stem ends in a voiceless consonant sound (like /f/, /k/, /s/, /p/, /sh/, /ch/).

The correct phonetic transcription for "laughed" is /lft/ (or /læft/ depending on dialect), where the gh is pronounced /f/ and the -ed is pronounced /t/.

Option (A) is /La : ft/ (or a similar transcription), which correctly represents the /t/ sound at the end.

 Quick Tip

Remember the three pronunciations of the past tense suffix -ed: /t/ after voiceless consonants (laugh-ed → /t/), /d/ after voiced consonants and vowels (call-ed → /d/), and /d/ after t or d (want-ed → /d/).

91. Pure vowels are also known as

- (A) Diphthongs
- (B) Monophthongs
- (C) Triphthongs
- (D) Monothongs

Correct Answer: (B) Monophthongs

Solution:

Vowels are generally classified into two groups based on their articulation:

1. **Monophthongs:** Vowels produced with a single, unvarying tongue position. They are single, pure sounds. Hence, they are also known as **pure vowels**.
2. **Diphthongs:** Vowels produced by the tongue moving from one vowel position to another within the same syllable, resulting in a gliding sound.

A **Triphthong** is a vowel sound that involves three distinct vowel qualities produced sequentially in the same syllable.

Therefore, pure vowels are also known as Monophthongs.

 Quick Tip

The terms are derived from Greek: 'Mono' (single), 'Di' (two), 'Tri' (three), and 'phthongos' (sound/voice). Monophthongs are single, pure vowel sounds.

92. Choose the correct order of the parts given below to make it a meaningful sentence

- i. not only for
- ii. but also for
- iii. lumbering
- iv. construction purposes
- v. as an occupation
- vi. on modern lines
- vii. the manufacture of wood pulp, paper, resins etc.
- viii. owing to the great demand for timber
- ix. has developed

(A) iii, ix, vi, v, viii, i, iv, ii, vii

(B) iii, viii, ix, v, iv, vii, ii, i, vi

(C) iii, v, ix, vi, viii, i, iv, ii, vii

(D) v, iii, ix, vi, viii, i, vii, ii, iv

Correct Answer: (A) iii, ix, vi, v, viii, i, iv, ii, vii

Solution:

We need to find the logical and grammatical flow for the meaningful sentence.

The sentence is likely about the development of **lumbering** (iii).

Start with (iii): lumbering

What happened to lumbering? It has developed (ix). → (iii, ix)

How has it developed? on modern lines (vi). → (iii, ix, vi)

In what context? as an occupation (v). → (iii, ix, vi, v)

Why has it developed? owing to the great demand for timber (viii). → (iii, ix, vi, v, viii)


The demand is for what? (Not only... but also... structure) not only for (i) → (iii, ix, vi, v, viii, i)

The first purpose is: construction purposes (iv). → (iii, ix, vi, v, viii, i, iv)

The second part of the structure: but also for (ii). → (iii, ix, vi, v, viii, i, iv, ii)

The second purpose is: the manufacture of wood pulp, paper, resins etc. (vii). → (iii, ix, vi, v, viii, i, iv, ii, vii)

Final Sentence: "Lumbering has developed on modern lines as an occupation, owing to the great demand for timber not only for construction purposes but also for the manufacture of wood pulp, paper, resins etc."

 Quick Tip

For jumbled sentences, identify the subject (iii. lumbering) and the main verb (ix. has developed) to establish the core. Then, look for conjunctions and paired structures like "not only" (i) and "but also" (ii) to link the remaining phrases logically.

93. Choose the word or phrase which is most nearly opposite in meaning to the word "JEOPARDISE"

- (A) Help
- (B) Encourage
- (C) Take care
- (D) Preserve

Correct Answer: (D) Preserve

Solution:

The word **Jeopardise** means to put (someone or something) into a situation in which there is a danger of loss, harm, or failure; to endanger or threaten.

We are looking for the word most nearly **opposite** in meaning (an antonym).

- (A) **Help:** To assist. (Too general)
- (B) **Encourage:** To give support, confidence, or hope. (Not a direct opposite)
- (C) **Take care:** To be cautious. (Too general)
- (D) **Preserve:** To maintain (something) in its original or existing state; to keep safe from injury or destruction.

Since "jeopardise" means to put in danger, the most direct opposite is to keep safe, which is **Preserve**.

- (A) a
- (B) b
- (C) c
- (D) d

Correct Answer: (D) d

Solution:

The sentence is: In this way nuclear fission or the splitting of the atom have been achieved.

The sentence structure involves two subjects connected by the conjunction or:

Subject 1: nuclear fission (singular)

Subject 2: the splitting of the atom (singular)

Rule: When two singular subjects are joined by or or nor, the verb must be singular.

The verb used is have been achieved (plural verb).

It should be has been achieved (singular verb).

The error is in part (d) due to the use of the plural verb **have** with singular subjects connected by or.

 Quick Tip

For compound subjects joined by 'or' or 'nor', the verb agrees with the subject closest to it. If both are singular, the verb must be singular. Correct usage: "nuclear fission or the splitting of the atom **has** been achieved."

96. The range of index for an array of size 'n' is -----

- (A) 0 to n
- (B) 1 to n
- (C) 0 to $n - 1$
- (D) 1 to $n - 1$

Correct Answer: (C) 0 to $n - 1$

Solution:

In most modern programming languages (like C, C++, Java, Python), array indexing starts at 0.

For an array of size n , this means there are n elements.

The indices of these n elements will be:

- First element: index 0.
- Second element: index 1.
- ...
- n^{th} (Last) element: index $n - 1$.

The range of indices runs from 0 up to $n - 1$.

 Quick Tip

Zero-based indexing is a core concept in programming. An array of size N has N elements, indexed from 0 to $N - 1$. Accessing index N results in an Array Index Out of Bounds error.

97. What is the output of the following program?

```
#include <stdio.h>
void main ()
{
int x = 1, y = 1, z;
z = x++ + ++y;
printf("x = %d, y = %d", x, y);
}
```

- (A) $x = 1, y = 1$
- (B) $x = 2, y = 1$
- (C) $x = 1, y = 2$
- (D) $x = 2, y = 2$

Correct Answer: (D) $x = 2, y = 2$

Solution:


The program initializes $x = 1$ and $y = 1$.

The core operation is: `z = x++ + ++y;`

1. **Pre-increment** `++y`: The value of `y` is incremented immediately before its use. `y` becomes 2. The value used in the expression is 2.
2. **Post-increment** `x++`: The original value of `x` is used in the expression, and then `x` is incremented. The value used in the expression is 1. `x` becomes 2.
3. **Evaluation**: `z = (value of x++) + (value of ++y) = 1 + 2 = 3.`
4. **Final values**: After the expression, `x = 2` and `y = 2.`

The `printf` statement prints the final values of `x` and `y`.

Output: `x = 2, y = 2`

 Quick Tip

Remember the order of operations for increment/decrement: Pre-increment (`++var`) updates the variable before its value is used in the expression. Post-increment (`var++`) uses the variable's current value in the expression, and then updates the variable.

98. Which of the following is a Unary operator in C Program?

- (A) `&`
- (B) `&&`
- (C) `<<`
- (D) `sizeof ()`

Correct Answer: (D) `sizeof ()`

Solution:

A **Unary operator** operates on a single operand.

- (A) `&` is the address-of operator (unary), OR it is the bitwise AND operator (binary). In C, `&` as the address-of operator is a unary operator.
- (B) `&&` is the logical AND operator (binary, operates on two operands).
- (C) `<<` is the bitwise left shift operator (binary, operates on two operands).

(D) `sizeof()` is the size-of operator. It takes a single operand (a type or a variable) and returns its size in bytes. This is a **Unary operator**.

In the strict list of options, `sizeof()` is definitively a unary operator. While `&` (address-of) is also unary, `sizeof()` is a clear and unambiguous example of a unary operator. Given the multiple-choice format, `sizeof()` is an expected unary operator.

 Quick Tip

Common unary operators include: address-of (`&`), value-at (`()`), increment (`++`), decrement (`--`), logical NOT (`!`), unary minus (`-`), and `sizeof()`.

99. Which of the following sorting algorithms is also called sinking sort?

- (A) Selection sort
- (B) Bubble sort
- (C) Insertion sort
- (D) Merge sort

Correct Answer: (B) Bubble sort

Solution:

Bubble Sort is an algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order.

In Bubble Sort, the largest (or smallest) element "bubbles up" to its correct position at the end of the list in each pass.

Conversely, the smallest (or largest) elements tend to "sink" slowly to the beginning of the list. Due to this movement of smaller elements, Bubble Sort is sometimes informally referred to as **Sinking Sort** (referring to the gradual movement of small elements to the beginning).

 Quick Tip

The terms "Sinking Sort" (referring to the small elements moving to the bottom/start) and "Bubble Sort" (referring to the large elements moving to the top/end) are synonymous and refer to the same basic comparison-and-swap algorithm.

100. What is the output of the following C program?

```
#include <stdio.h>
#include <string.h>
void main ()
{
char str1[] = "bengaluru";
char str2[] = "bengaluru";
if (strcmp (str1, str2))
printf("equal");
else
printf("Not equal");
}
```

- (A) Equal
- (B) Not equal
- (C) Compilation error
- (D) Depends on the compiler

Correct Answer: (B) Not equal

Solution:

The program compares two identical strings "bengaluru" using `strcmp()` and then uses the return value in an `if` condition.

1. `strcmp(str1, str2)`: This standard library function compares two strings lexicographically.

- It returns **0** if the strings are identical.
- It returns a negative value if `str1` is lexicographically less than `str2`.
- It returns a positive value if `str1` is lexicographically greater than `str2`.

Since `str1` and `str2` are both "bengaluru", `strcmp (str1, str2)` returns **0**.

2. `if (0)`: In C, the condition for an `if` statement evaluates non-zero values as **TRUE** and 0 as **FALSE**.

Since the condition evaluates to 0 (false), the `if` block (`printf("equal")`) is skipped.

3. **Execution**: The `else` block is executed.

```
printf("Not equal");
```

The output of the program is `Not equal`.

💡 Quick Tip

Remember the return value of `strcmp()`: it is 0 for equal strings. When using `strcmp()` for a boolean check of equality, you must explicitly check for the return value of 0, e.g., `if (strcmp(s1, s2) == 0)`.
