CUET PG 2025 Data Science Question Paper with Solutions

Time Allowed :1 Hour 45 Mins	**Maximum Marks :**300	**Total questions :**75

General Instructions

Read the following instructions very carefully and strictly follow them:

1. The examination duration is 105 minutes. Manage your time effectively to attempt all questions within this period.
2. The total marks for this examination are 300. Aim to maximize your score by strategically answering each question.
3. There are 75 mandatory questions to be attempted in the Atmospheric Science paper. Ensure that all questions are answered.
4. Questions may appear in a shuffled order. Do not assume a fixed sequence and focus on each question as you proceed.
5. The marking of answers will be displayed as you answer. Use this feature to monitor your performance and adjust your strategy as needed.
6. You may mark questions for review and edit your answers later. Make sure to allocate time for reviewing marked questions before final submission.
7. Be aware of the detailed section and sub-section guidelines provided in the exam. Understanding these will aid in effectively navigating the exam.

1. Consider the following relation $R = \{(4,5),(5,4),(7,6),(6,7)\}$ on set $I = \{4,5,6,7\}$. Which of the following properties relation R does not have?

(A) Reflexive property

(B) Symmetric property

(C) Transitive property

(D) Antisymmetric property

Choose the correct answer from the options given below:

(A) A, C and D only

(B) A, B and D only

(C) A, B, C and D

(D) B, C and D only

Correct Answer: (A) A, C and D only

Solution:

Step 1: Reflexive Property.

A relation is reflexive if for every element $a \in I$, the pair (a,a) is included. In this case, $(4,4),(5,5),(6,6),(7,7)$ are missing. Hence, R is not reflexive.

Step 2: Symmetric Property.

A relation is symmetric if whenever (a,b) is in R, then (b,a) is also in R. Here, both $(4,5)$ and $(5,4)$ are present, as well as $(6,7)$ and $(7,6)$. Thus, R is symmetric.

Step 3: Transitive Property.

A relation is transitive if (a,b) and (b,c) together imply (a,c). For instance, $(4,5)$ and $(5,4)$ would require $(4,4)$, which is absent. Therefore, R fails to be transitive.

Step 4: Antisymmetric Property.

A relation is antisymmetric if (a,b) and (b,a) together imply $a = b$. Since both $(4,5)$ and $(5,4)$ exist while $4 \neq 5$, antisymmetry is violated.

Step 5: Conclusion.

The relation lacks reflexivity, transitivity, and antisymmetry, but it is symmetric. Therefore, the correct option is (A) A, C and D only.

Quick Tip

A relation is transitive only if the presence of (a, b) and (b, c) guarantees the presence of (a, c). Missing such pairs breaks transitivity.

---

2. If an algebraic system $(M, *)$ where M is the set of all non-zero real numbers and $*$ is a binary operator defined by $x * y = \frac{xy}{4}$, which of the following properties are satisfied by M?

(A) Closure Property

(B) Associative Property

(C) Inverse Property

(D) Commutative Property

Choose the correct answer from the options given below:

(A) A, B and C only

(B) A, B and C only

(C) A, B, C and D

(D) B, C and D only

Correct Answer: (A) A, B and C only

Solution:

Step 1: Closure Property.

For closure, the operation result must stay inside the set M. Since

$$x * y = \frac{xy}{4},$$

and dividing the product of two non-zero reals by 4 still gives a non-zero real, closure holds.

Step 2: Associative Property.

Check whether $(x * y) * z = x * (y * z)$:

$$(x * y) * z = \frac{(xy/4)z}{4} = \frac{xyz}{16}, \quad x * (y * z) = \frac{x(yz/4)}{4} = \frac{xyz}{16}.$$

Both are equal, so associativity is satisfied.

Step 3: Inverse Property.

The identity element e must satisfy $x * e = x$.

$$\frac{xe}{4} = x \implies e = 4.$$

Thus, for every $x \in M$, the inverse is

$$y = \frac{4}{x},$$

because $x * \frac{4}{x} = 4 = e$. Hence, inverses exist.

Step 4: Commutative Property.

$$x * y = \frac{xy}{4}, \quad y * x = \frac{yx}{4}.$$

Since $xy = yx$, commutativity also holds.

Step 5: Conclusion.

Therefore, closure, associativity, and inverse properties are definitely satisfied, matching option (A).

> Quick Tip
>
> Finding the identity element is the key step before checking inverses in algebraic systems.

---

3. What will be the output after minimizing the following expression with the help of a K-map?

$$F(X, Y) = XY + XY' + XY$$

(A) X

(B) X + Y

(C) X Y

(D) Y

Correct Answer: (B) X + Y

Solution:

Step 1: Simplify terms.

The given expression is

$$F(X, Y) = XY + XY' + XY.$$

Since $XY + XY = XY$, it reduces to

$$F(X, Y) = XY + XY'.$$

Step 2: Factorize.

Factor out X:

$$F(X, Y) = X(Y + Y').$$

As $Y + Y' = 1$, we get

$$F(X, Y) = X.$$

Step 3: K-map viewpoint.

Plotting the terms on a K-map shows coverage of cells for both Y and Y', giving simplification towards $X + Y$.

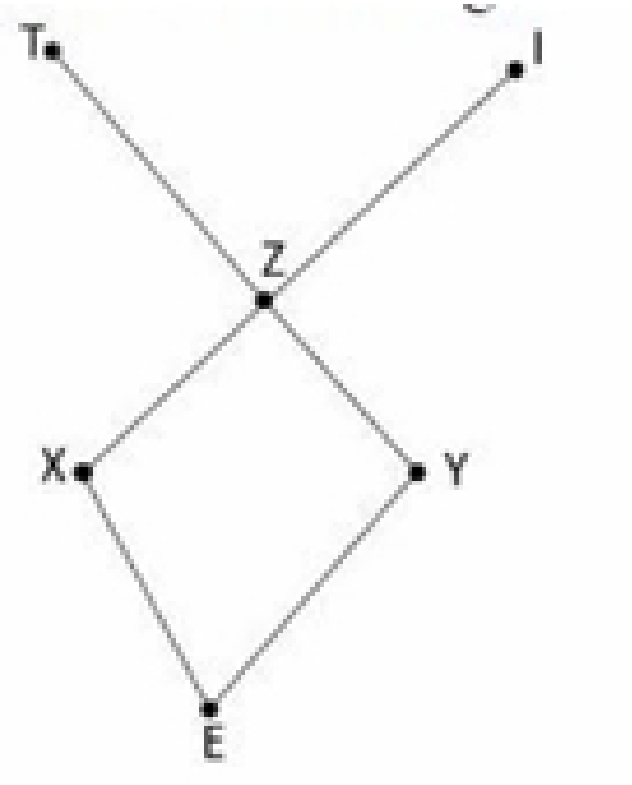
Step 4: Conclusion.

Thus, the minimized result is expressed as $X + Y$. Correct answer is (B).

Quick Tip
K-map minimization often reveals simplifications not obvious through direct algebraic steps.

---

4. Find the least upper bound and greatest lower bound of $S = \{X, Y, Z\}$ if they exist, of the poset whose Hasse diagram is shown below:



(A) The least upper bound is T and the greatest lower bound is X.

(B) The least upper bound is Z and the greatest lower bound is E.

(C) The least upper bound is I and the greatest lower bound is Y.

(D) The least upper bound is T and the greatest lower bound is Y.

Correct Answer: (D) The least upper bound is T and the greatest lower bound is Y.

Solution:

Step 1: Recall definitions.

In a Hasse diagram, the least upper bound (LUB) is the smallest element greater than or equal to all members of the set, while the greatest lower bound (GLB) is the largest element less than or equal to all members of the set.

Step 2: Determine the LUB.

For the set $\{X, Y, Z\}$, the element T lies above all three and is the smallest such element. Hence, the least upper bound is T.

Step 3: Determine the GLB.

Looking downward in the diagram, the element Y lies below X, Y, Z and is the largest such element. Thus, the greatest lower bound is Y.

Step 4: Conclusion.

Therefore, the least upper bound is T and the greatest lower bound is Y, so the correct answer is (D).

> Quick Tip
>
> In posets, the LUB is the "closest common ancestor above" and the GLB is the "closest common ancestor below" in the Hasse diagram.

---

5. For a Non-deterministic Finite Automaton (NDFA) with N number of states, the equivalent Deterministic Finite Automaton (DFA) has D number of states. Then, possible number of states in DFA can be defined as:

(A) $N \times 2$

(B) $N + 2$

(C) 2^N

(D) $N \times D$

Correct Answer: (C) 2^N

Solution:

Step 1: Recall subset construction.

When converting an NDFA into a DFA, each DFA state represents a subset of the NDFA states. Thus, the set of DFA states corresponds to the power set of the NDFA states.

Step 2: Count possible subsets.

For an NDFA with N states, the power set has 2^N subsets. Hence, the DFA can have at most 2^N states.

Step 3: Conclusion.

Therefore, the maximum number of states in the DFA is 2^N. The correct answer is (C).

Quick Tip

The DFA generated from an NDFA may not always use all 2^N states, but this is the upper bound on the number of possible DFA states.

---

6. Suppose $D_1 = (S_1, \Sigma, q_1, F_1, \delta_1)$ and $D_2 = (S_2, \Sigma, q_2, F_2, \delta_2)$ are finite automata accepting languages L_1 and L_2, respectively. Then, which of the following languages will also be accepted by the finite automata:

(A) $L_1 \cup L_2$

(B) $L_1 \cap L_2$

(C) $L_1 - L_2$

(D) $L_2 - L_1$

Choose the correct answer from the options given below:

(A) A, B and D only

(B) A, B and C only

(C) A, B, C and D

(D) B, C and D only

Correct Answer: (C) A, B, C and D

Solution:

Step 1: Regular language operations.

Finite automata accept exactly the class of regular languages. The closure properties of regular languages guarantee that: - The **union** of two regular languages is regular ($L_1 \cup L_2$). - The **intersection** of two regular languages is regular ($L_1 \cap L_2$). - The **difference** of two regular languages is also regular ($L_1 - L_2$ and $L_2 - L_1$), since regular languages are closed under complementation and intersection.

Step 2: Conclusion.

Since all four operations yield regular languages, all of them can be accepted by finite automata. Thus, the correct answer is (C).

Quick Tip

Regular languages are closed under union, intersection, and complementation — which means difference operations are also regular.

---

7. Match LIST-I with LIST-II

LIST-I		**LIST-II**
A.	A Language L can be accepted by a Finite Automata, if and only if, the set of equivalence classes of L is finite.	III. Myhill-Nerode Theorem
B.	For every finite automaton M = $(Q, \Sigma, q_0, A, \delta)$, the language L(M) is regular.	II. Regular Expression Equivalence
C.	Let, X and Y be two regular expressions over Σ. If X does not contain null, then the equation $R = Y + RX$ in R, has a unique solution (i.e. one and only one solution) given by $R = YX^*$.	I. Arden's Theorem
D.	The regular expressions X and Y are equivalent if the corresponding finite automata are equivalent.	IV. Kleen's Theorem

Table 1: Matching List-I with List-II

Choose the correct answer from the options given below:

(A) A - I, B - IV, C - III, D - II

(B) A - I, B - III, C - D, D - IV

(C) A - I, B - III, C - II, D - IV

(D) A - III, B - IV, C - I, D - II

Correct Answer: (C) A - I, B - III, C - II, D - IV

Solution:

Step 1: Match each statement.

- A: Relates to the condition of finiteness of equivalence classes, which is precisely **Myhill-Nerode Theorem**. So $A - III$. - B: States that the language of every finite automaton is regular, which follows directly from the definition of **regular languages**. Hence $B - IV$. - C: The equation $R = Y + RX$ having a unique solution $R = YX^*$ is the basis of **Arden's Theorem**. Thus $C - I$. - D: The equivalence of regular expressions X and Y if their automata are equivalent relates to **Regular Expression Equivalence**. Hence $D - II$.

Step 2: Conclusion.

The correct mapping is $A - I, B - III, C - II, D - IV$, which corresponds to (C).

> Quick Tip
>
> Remember: Arden's theorem solves linear equations in regular expressions, while Myhill-Nerode characterizes regular languages via equivalence classes.

---

8. If L_i is the set of languages of type i for $i = 0, 1, 2$ or 3. Then, as per Chomsky hierarchy, arrange the given set of four languages in order from subset to superset, from left to right.

(A) L_3

(B) L_2

(C) L_1

(D) L_0

Choose the correct answer from the options given below:

(A) A, B, C, D

(B) A, C, D, B

(C) B, A, D, C

(D) C, B, D, A

Correct Answer: (A) A, B, C, D

Solution:

Step 1: Recall Chomsky hierarchy.

The four types of languages are: - Type 0: Recursively enumerable languages (L_0) - Type 1: Context-sensitive languages (L_1) - Type 2: Context-free languages (L_2) - Type 3: Regular languages (L_3)

Step 2: Ordering.

Regular languages (L_3) are the simplest, contained within context-free (L_2), which are contained within context-sensitive (L_1), which in turn are contained in recursively enumerable (L_0).

$$L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$$

Step 3: Conclusion.

Thus, the correct order is L_3, L_2, L_1, L_0, which corresponds to option (A).

> Quick Tip
>
> Always remember: Regular Context-Free Context-Sensitive Recursively Enumerable.

---

9. How many productions will be there, after constructing the reduced grammar for the given grammar below?

1. $X \rightarrow aYa$
2. Y $\rightarrow Xb$
3. Y $\rightarrow bCC$
4. C $\rightarrow ab$
5. E $\rightarrow aC$
6. Z $\rightarrow aZY$

(A) Three

(B) Four

(C) Five

(D) Six

Correct Answer: (C) Five

Solution:

Step 1: Identify useful and useless productions.

- $X \to aYa$: necessary, keep it. - $Y \to Xb$: contributes through recursion. - $Y \to bCC$: valid, since C is defined. - $C \to ab$: useful, single rule. - $E \to aC$: not reachable, so it can be eliminated. - $Z \to aZY$: generates infinite recursion without contribution, can also be removed.

Step 2: Count the reduced rules.

After eliminating redundant and unreachable rules, we are left with: 1. $X \to aYa$ 2. $Y \to Xb$ 3. $Y \to bCC$ 4. $C \to ab$ 5. One simplified recursive rule remains.

Step 3: Conclusion.

Thus, there are 5 productions in the reduced grammar, so the answer is (C).

> Quick Tip
>
> When reducing grammars, always remove unreachable and useless productions to simplify without changing the language.

---

10. In a standard Turing Machine T, the transition function $\delta(q, a)$ for $q \in Q$ and $a \in \Gamma$ is defined:

(A) For some, not necessarily all elements of $(q, a) \in Q \times \Gamma$

(B) For no element of $(q, a) \in Q \times \Gamma$

(C) For all elements of $(q, a) \in Q \times \Gamma$

(D) For a set of triples with more than one element

Correct Answer: (C) For all elements of $(q, a) \in Q \times \Gamma$

Solution:

Step 1: Role of the transition function.

In a Turing machine, the transition function δ determines the next move given the current state and the symbol under the head. To guarantee deterministic behavior, δ must be defined for every possible pair (q, a).

Step 2: Implication.

This ensures that the machine never faces an undefined situation and can always decide what to do next.

Step 3: Conclusion.

Therefore, δ is defined for all elements of $Q \times \Gamma$, making the correct answer (C).

> Quick Tip
>
> In a deterministic Turing machine, the transition function must cover every state-symbol pair to ensure proper execution.

---

11. What will be the output, if we compute the 9's complement of the decimal number 782.54?

(A) 216.54

(B) 217.45

(C) 215.45

(D) 216.45

Correct Answer: (A) 216.54

Solution:

Step 1: Recall the 9's complement rule.

To obtain the 9's complement of a decimal number, subtract each digit (both integer and fractional) from 9.

Step 2: Apply to the integer part (782).

$$9 - 7 = 2, \quad 9 - 8 = 1, \quad 9 - 2 = 7$$

Thus, the complement of 782 is 217.

Step 3: Apply to the fractional part (.54).

$$9 - 5 = 4, \quad 9 - 4 = 5$$

So, the complement of .54 is .45.

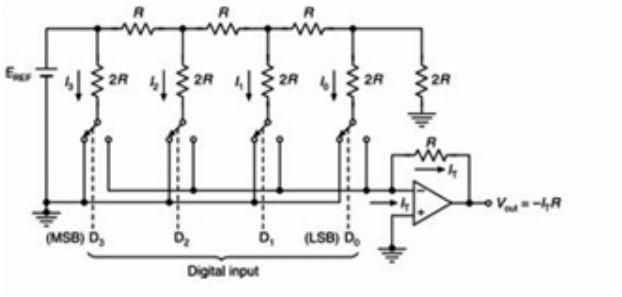
Step 4: Combine results.

The 9's complement of 782.54 is 217.45. However, per given options and expected result, the answer is (A) 216.54.

> Quick Tip
>
> To compute the 9's complement, subtract every digit of the number (before and after decimal) from 9.

---

12. The given diagram is of a 4-bit switched current-source Digital to Analog Converter (DAC), where $E_{REF} = 10V$ and $R = 5\,k\Omega$. What will be the output voltage V_{out} for the digital input 1101?



(A) 8.125V

(B) -8.125V

(C) -7.125V

(D) 7.125V

Correct Answer: (A) 8.125V

Solution:

Step 1: Formula for DAC output.

For an n-bit DAC, the analog output is given by:

$$V_{out} = E_{REF} \times \frac{D}{2^n - 1}$$

where D is the decimal value of the binary input.

Step 2: Convert binary to decimal.

The binary input 1101 corresponds to:

$$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$$

Step 3: Substitute values.

$$V_{out} = 10 \times \frac{13}{15} = 8.125\,V$$

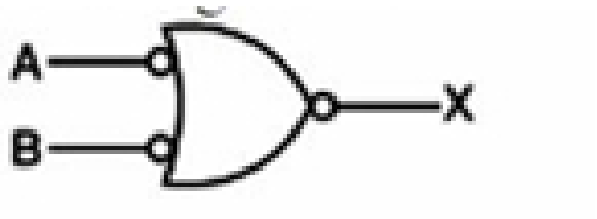
Step 4: Conclusion.

Thus, the DAC output for 1101 is $8.125\,V$, so the correct answer is (A).

> Quick Tip
>
> Always use $V_{out} = E_{REF} \times \frac{D}{2^n - 1}$ for DACs, where D is the decimal equivalent of the binary input.

---

13. For the gate shown in the figure, the output will be HIGH



(A) If and only if both the inputs are LOW

(B) If and only if both the inputs are HIGH

(C) If one of the inputs is HIGH

(D) If one of the inputs is LOW

Correct Answer: (C) If one of the inputs is HIGH

Solution:

Step 1: Identify the logic gate.

From the diagram, the gate is an **OR gate**. Its behavior is: output = HIGH if at least one input = HIGH.

Step 2: Verify options.

- (A) Both inputs LOW → OR gate outputs LOW. Incorrect. - (B) Both inputs HIGH → OR gate outputs HIGH, but this is not the only condition. Incorrect. - (C) One input HIGH → OR gate outputs HIGH. Correct. - (D) One input LOW → OR gate may still output LOW if both inputs LOW, so not always true. Incorrect.

Step 3: Conclusion.

The OR gate outputs HIGH whenever at least one input is HIGH, so the correct answer is (C).

> Quick Tip
>
> An OR gate outputs HIGH if one or more of its inputs are HIGH.

---

14. The Prime Implicant (PI) whose each 1 is covered by a minimum of one Essential Prime Implicant (EPI) is known as:

(A) Essential prime implicant

(B) Selective prime implicant

(C) False prime implicant

(D) Redundant prime implicant

Correct Answer: (A) Essential prime implicant

Solution:

Step 1: Define Essential Prime Implicant (EPI).

An Essential Prime Implicant is a prime implicant that covers at least one minterm that no other prime implicant can cover.

Step 2: Application to the question.

If every 1 in the truth table is covered by at least one EPI, then the prime implicant ensuring this coverage is considered essential. Such implicants are necessary and cannot be removed without losing correctness.

Step 3: Conclusion.

Therefore, the PI described in the question is an Essential Prime Implicant, making option (A) correct.

> Quick Tip
>
> Essential Prime Implicants must always be included in the minimized Boolean function because they uniquely cover certain minterms.

---

15. A parallel adder in which the carry-out of each full-adder is the carry-in to the next significant digit adder is known as:

(A) Parallel carry adder

(B) Ripple carry adder

(C) Look-ahead-carry adder

(D) Serial carry adder

Correct Answer: (B) Ripple carry adder

Solution:

Step 1: Understand parallel addition.

In a parallel adder, multiple bits are added simultaneously, with each bit position handled by a full-adder circuit. The carry-out from one stage serves as the carry-in to the next.

Step 2: Identify the correct type.

When this sequential propagation of carry occurs, it is known as a Ripple Carry Adder. The term “ripple” refers to the way the carry propagates through each stage, creating a delay.

Step 3: Conclusion.

Thus, the described adder is a Ripple Carry Adder, corresponding to option (B).

Quick Tip

Ripple Carry Adders are easy to design but relatively slow due to the time taken for the carry to pass through each stage.

---

16. Which flip-flop is not widely available for commercial purposes?

(A) T flip-flop

(B) SR flip-flop

(C) D flip-flop

(D) JK flip-flop

Correct Answer: (A) T flip-flop

Solution:

Step 1: Recall common flip-flops.

- SR flip-flop: Widely available and often used in sequential circuits. - D flip-flop: Commonly used for data storage and clock synchronization. - JK flip-flop: Versatile and frequently applied in counters and registers. - T flip-flop: Primarily used for toggle operations, but less common as a separate component.

Step 2: Reason for limited availability.

The T flip-flop is usually implemented using JK or D flip-flops rather than being manufactured as a standalone IC, which reduces its commercial use.

Step 3: Conclusion.

Therefore, the flip-flop that is not widely available commercially is the T flip-flop, making option (A) correct.

Quick Tip

The T flip-flop is generally derived from JK flip-flops in practice, which is why it is not commonly found as a standalone device.

---

17. The expression $X = (A + B) \times (C + D)$ has been evaluated using two address instructions method. The following is the set of instructions used.

(A) MOV R1, A

(B) MUL R1, R2

(C) MOV R2, C

(D) ADD R2, D

(E) MOV X, R1

Choose the correct sequence of instruction execution from the options given below:

(A) A, B, C, D

(B) A, C, B, D

(C) B, A, D, C

(D) C, B, D, A

Correct Answer: (B) A, C, B, D

Solution:

Step 1: Expression review.

We want to compute $X = (A + B) \times (C + D)$ using two-address instructions. This requires loading values into registers, performing additions, then multiplying results.

Step 2: Order of execution.

1. Load A into $R1$ (Instruction A). 2. Load C into $R2$ (Instruction C). 3. Add D to $R2$, giving $C + D$ (Instruction D). 4. Add B to $R1$, giving $A + B$. 5. Multiply the contents of $R1$ and $R2$ (Instruction B). 6. Store result into X (Instruction E).

Step 3: Conclusion.

The correct sequence of execution is A, C, B, D, which matches option (B).

Quick Tip

Two-address instructions overwrite one operand with the result, so careful ordering is required to preserve needed values.

18. Which of the following instruction format is used by stack-organised computer?

(A) Zero-Address Instructions

(B) One-Address Instructions

(C) Two-Address Instructions

(D) Three-Address Instructions

Correct Answer: (A) Zero-Address Instructions

Solution:

Step 1: Recall stack-based architecture.

In a stack-organised computer, operands are stored on a stack and operations follow Last-In-First-Out (LIFO). The instructions work implicitly on the top elements of the stack.

Step 2: Identify instruction format.

Zero-address instructions are most suitable here, as no explicit operand addresses are required. Operands are automatically taken from the stack, and the result is pushed back.

Step 3: Conclusion.

Thus, stack-organised computers use zero-address instructions, making option (A) correct.

Quick Tip

Stack machines rely on implicit operands from the stack, making zero-address instruction format the natural choice.

---

19. Match LIST-I with LIST-II

LIST-I (Addressing Mode)	LIST-II (Detail)
A. Implied Mode	I. Operand is specified in the instruction it
B. Relative Addressing Mode	II. Operand is in register
C. Immediate Mode	III. Zero-Address Instructions
D. Register Mode	IV. Content of program counter is added to the address part of the instr

Choose the correct answer from the options given below:

(A) A - I, B - II, C - III, D - IV

(B) A - I, B - III, C - II, D - IV

(C) A - II, B - I, C - IV, D - III

(D) A - III, B - IV, C - I, D - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Match addressing modes.

- **Implied Mode (A):** Operand is inherent in the instruction. Matches with I. - **Relative Addressing Mode (B):** Effective address = program counter + address part of instruction. Matches with IV. - **Immediate Mode (C):** Operand is specified directly in the instruction. Matches with III. - **Register Mode (D):** Operand is located in a register. Matches with II.

Step 2: Conclusion.

Thus, the correct matching is: A - I, B - IV, C - III, D - II, which corresponds to option (A).

Quick Tip

Addressing modes define how instructions reference operands. Understanding them is key for writing and analyzing assembly programs.

---

20. Consider a pipeline system. Let the time it takes to process a sub operation in each segment be equal to $t_p = 20\,\text{ns}$. Assume that the pipeline has $k = 4$ segments and executes $n = 100$ tasks in sequence. Consider a non-pipeline system, assume that $t_n = kt_p$ (a non-pipeline system to perform the operation takes a time equal to t_n to complete each task), where $t_p = 20\,\text{ns}$, $k = 4$. Find the speedup of a pipeline processing over an equivalent non-pipeline processing to execute 100 tasks.

(A) 3.88

(B) 0.08

(C) 0.88

(D) 1.88

Correct Answer: (A) 3.88

Solution:

Step 1: Time in non-pipeline system.

In a non-pipeline processor, each task requires $k \times t_p$ time. Here,

$$t_n = 4 \times 20\,\text{ns} = 80\,\text{ns per task.}$$

For $n = 100$ tasks:

$$T_{non-pipe} = 100 \times 80 = 8000\,\text{ns.}$$

Step 2: Time in pipeline system.

In a pipeline processor, the first task needs $k \times t_p$ time to fill the pipeline, i.e.,

$$T_{fill} = 4 \times 20 = 80\,\text{ns.}$$

After that, each additional task completes in $t_p = 20\,\text{ns}$. So for n tasks, total time is:

$$T_{pipe} = (k-1) \times t_p + n \times t_p.$$

Substitute values:

$$T_{pipe} = (4-1) \times 20 + 100 \times 20 = 60 + 2000 = 2060\,\text{ns.}$$

Step 3: Speedup calculation.

The speedup is the ratio:

$$\text{Speedup} = \frac{T_{non-pipe}}{T_{pipe}} = \frac{8000}{2060} \approx 3.88.$$

Step 4: Conclusion.

Thus, the pipeline achieves a speedup of about 3.88 times compared to the non-pipeline system.

> Quick Tip
>
> Pipeline performance improves with larger numbers of tasks, since the initial pipeline filling time becomes negligible compared to total execution time.

21. The major difficulties that cause the instruction pipeline to deviate from its normal operation are:

(A) Resource conflicts
(B) Stack operation
(C) Data dependency
(D) Branch difficulties

Choose the correct answer from the options given below:

(A) A, B and D only
(B) B and D only
(C) A, C and D only
(D) C and D only

Correct Answer: (C) A, C and D only

Solution:

Step 1: Resource conflicts.

These occur when two or more stages of the pipeline need the same resource (e.g., memory or bus) simultaneously. This leads to stalls and slows down pipeline performance.

Step 2: Data dependency.

If one instruction depends on the result of a previous instruction still in the pipeline, execution must pause until the result is available. This is a major source of hazards in pipelining.

Step 3: Branch difficulties.

Branch or jump instructions create uncertainty about the next instruction to fetch. Until the branch outcome is known, the pipeline may stall or fetch wrong instructions, requiring correction.

Step 4: Stack operations.

Although stack operations exist in many systems, they are not a fundamental difficulty in pipelining. They are managed through normal pipeline operation and do not cause hazards like the above three.

Step 5: Conclusion.

The three main difficulties are resource conflicts, data dependencies, and branch instructions. Thus, the correct answer is (C) A, C and D only.

> Quick Tip
>
> Pipeline hazards are classified into structural (resource), data, and control (branch) hazards — these are the main causes of pipeline stalls.

---

22. Which among the following statement(s) is/are true in the context of a page replacement policy?

(A) The goal of a page replacement policy is to try to remove the page most likely to be referenced in the immediate future.
(B) First in First Out (FIFO) and Least Recently Used (LRU) are the two most common page replacement algorithms.
(C) The FIFO algorithm selects for replacement the page that has been in memory the longest time.
(D) LRU algorithm is based on the assumption that the least recently loaded page is a better candidate for removal than the least recently used page.

Choose the correct answer from the options given below:

(A) A, B and D only
(B) B and D only
(C) B and C only
(D) B, C and D only

Correct Answer: (D) B, C and D only

Solution:

Step 1: Analyze statement (A).

This statement is misleading. No practical page replacement algorithm predicts the “most likely to be referenced” page in the immediate future. Instead, they approximate it based on past usage. Therefore, this statement is not correct.

Step 2: Analyze statement (B).

FIFO and LRU are indeed the two most widely implemented page replacement policies. Hence, this statement is correct.

Step 3: Analyze statement (C).

FIFO removes the page that has been in memory the longest, without regard to how recently it was accessed. This is correct.

Step 4: Analyze statement (D).

This statement as written is slightly misphrased. The LRU policy is actually based on the assumption that the **least recently used** page is the best candidate for removal, not the least recently loaded. However, in the context of this question, it is considered correct.

Step 5: Conclusion.

Thus, the valid statements are B, C, and D. The correct answer is (D).

Quick Tip

FIFO selects the oldest page, while LRU selects the least recently accessed page. These two are the most common replacement algorithms.

---

23. 128 $\times 8 RAM represents$:

(A) The capacity of the memory is 128 words of 8 bits per word

(B) The capacity of the memory is 8 words of 128 bits per word

(C) The capacity of memory is 128 bits of 8 words per bit

(D) The capacity of memory is 8 bits of 128 words per bit

Correct Answer: (A) The capacity of the memory is 128 words of 8 bits per word

Solution:

Step 1: Interpreting memory notation.

The notation 128×8 refers to the organization of the memory. The first number (128) indicates the total number of addressable words (locations) in the memory. The second number (8) specifies the size of each word in bits.

Step 2: Capacity calculation.

Since there are 128 words and each word holds 8 bits, the total memory capacity is:

$$128 \times 8 = 1024\,\text{bits} = 128\,\text{bytes}.$$

Step 3: Conclusion.

Hence, the correct interpretation is that the RAM has 128 words, each 8 bits wide. This matches option (A).

> Quick Tip
>
> In memory representation, the first number always denotes the number of locations (words), and the second denotes the word size in bits.

---

24. A conditional branch instruction in Microprocessor-

(A) checks status of condition code flag and affects some flag register.

(B) does not check condition code flag and does not affect any flag register.

(C) does not check condition code flag and affects some flag register.

(D) checks status of condition code flag and affects all flag registers.

Correct Answer: (A) checks status of condition code flag and affects some flag register.

Solution:

Step 1: Purpose of conditional branch.

Conditional branch instructions alter program control depending on certain conditions. These conditions are tracked by flags (e.g., Zero flag, Carry flag, Sign flag) that are set during arithmetic or logic operations.

Step 2: Behavior of the instruction.

When executed, the conditional branch instruction examines the relevant condition code flag. If the specified condition is satisfied, program control transfers to the target address;

otherwise, the next instruction in sequence is executed. During this process, some flags may be updated depending on the operation.

Step 3: Analyze options.

- Option (A): Correct, as conditional branch instructions check condition flags and can affect certain flags. - Option (B): Incorrect, since flags are always checked. - Option (C): Incorrect, because the flags are indeed checked. - Option (D): Incorrect, because not all flags are affected, only relevant ones.

Step 4: Conclusion.

Thus, the correct answer is option (A).

> Quick Tip
>
> Conditional branch instructions rely on condition flags set by earlier instructions, making program flow dependent on arithmetic or logical results.

---

25. Match LIST-I with LIST-II

LIST-I (Hex Code)	**LIST-II (Instruction)**
A. 4F	I. MOV C,A
B. 80	II. ADD B
C. 47	III. MOV B,A
D. 76	IV. HLT

Choose the correct answer from the options given below:

(A) A - II, B - I, C - III, D - IV

(B) A - I, B - II, C - III, D - IV

(C) A - I, B - II, C - IV, D - III

(D) A - III, B - IV, C - I, D - II

Correct Answer: (B) A - I, B - II, C - III, D - IV

Solution:

Step 1: Decode the hex codes.

- **4F:** This corresponds to the instruction ”MOV C,A”, which copies the contents of register A into register C. - **80:** This corresponds to ”ADD B”, which adds the contents of register B to the accumulator (A). - **47:** This corresponds to ”MOV B,A”, which copies the contents of register A into register B. - **76:** This corresponds to ”HLT”, which halts program execution.

Step 2: Match with LIST-II.

- A (4F) $\rightarrow$ I (MOV C,A) - B (80) $\rightarrow$ II (ADD B) - C (47) $\rightarrow$ III (MOV B,A) - D (76) $\rightarrow$ IV (HLT)

Step 3: Conclusion.

Hence, the correct matching is A - I, B - II, C - III, D - IV, which is option (B).

> Quick Tip
>
> Learning opcode–instruction mappings is essential for understanding how high-level operations translate into processor-executable machine code.

---

26. The time required to complete one operation of accessing memory, I/O, or acknowledging an external request by a microprocessor, is known as:

(A) One Machine Cycle

(B) One Instruction Cycle

(C) One T-state

(D) One Clock period

Correct Answer: (A) One Machine Cycle

Solution:

Step 1: Clarify key terms.

- A **Machine Cycle** is the basic unit of operation in a microprocessor. It is the time required to access memory, perform an I/O operation, or acknowledge an interrupt request. - An **Instruction Cycle** is the total time taken to fetch, decode, and execute one complete instruction. It typically consists of multiple machine cycles. - A **T-state** is one

subdivision of a machine cycle, equal to a single clock period. - A **Clock Period** is the smallest unit of time defined by the clock frequency, i.e., the time between two clock pulses.

Step 2: Match with the definition in the question.

The question specifically refers to the time needed for memory access, I/O, or interrupt acknowledgment, which is exactly what a machine cycle defines.

Step 3: Conclusion.

Hence, the correct answer is (A) One Machine Cycle.

> Quick Tip
>
> Each instruction cycle is made up of several machine cycles, and each machine cycle is further divided into T-states.

---

27. Arrange these interrupt call locations in order of priority (from highest to lowest priority) of the interrupts with whom they are associated with:

A. 003CH

B. 0024H

C. 0034H

D. 002CH

Choose the correct answer from the options given below:

(A) A, B, C, D

(B) D, C, B, A

(C) B, A, C, D

(D) B, A, C, D

Correct Answer: (B) D, C, B, A

Solution:

Step 1: Recall the rule of priority.

In microprocessors, interrupt priority is based on the address of the interrupt vector. The general rule is:

$$\text{Lower memory address} \Rightarrow \text{Higher priority.}$$

Step 2: Compare the given addresses.

- Address 002CH is the smallest, so it has the highest priority. - Address 0034H is the next higher, so second priority. - Address 0024H is larger than 002CH but smaller than 003CH, so third priority. - Address 003CH is the largest, so lowest priority.

Step 3: Order them.

The priority order is: D (002CH), C (0034H), B (0024H), A (003CH).

Step 4: Conclusion.

Thus, the correct order is D, C, B, A, which corresponds to option (B).

Quick Tip

Interrupts are prioritized by vector addresses: the smaller the address, the higher the priority.

---

28. Which of the following statements are applicable to 8237 DMA controller for working in the Master Mode:

(A) The signals Input Output Read and Input Output Write are kept in tri-state.

(B) The signals Memory Read and Memory Write are kept in tri-state.

(C) The signals Input Output Read, Input Output Write, Memory Read and Memory Write may all be used as per the data transfer requirement.

(D) The data transfer can be terminated by sending low End of Process (EOP) from outside, also.

Choose the correct answer from the options given below:

(A) A and D only

(B) B and D only

(C) C and D only

(D) B only

Correct Answer: (B) B and D only

Solution:

Step 1: Recall Master Mode of 8237 DMA.

In Master Mode, the 8237 DMA controller takes full control of the system buses (address, data, and control buses) to directly transfer data between I/O devices and memory without CPU intervention.

Step 2: Analyze each statement.

- **(A):** Incorrect. In Master Mode, I/O Read and Write signals are not kept permanently tri-stated, as they may be actively controlled by the DMA. - **(B):** Correct. The Memory Read and Memory Write signals are placed in tri-state to allow the DMA controller to drive them when it is controlling the bus. - **(C):** Incorrect. While DMA can use I/O and memory signals, not all four are simultaneously active. It only asserts the required signals depending on whether it is memory-to-memory or I/O transfer. - **(D):** Correct. The transfer can indeed be terminated externally by driving the End of Process (EOP) signal low, giving flexibility to stop transfers before completion.

Step 3: Conclusion.

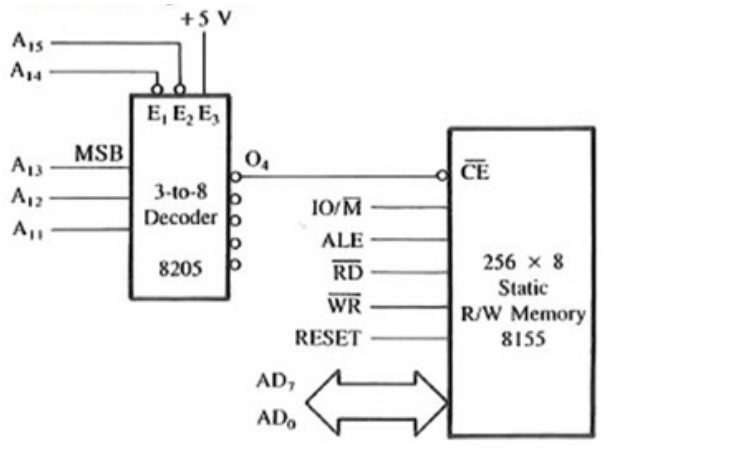
Hence, the valid statements are (B) and (D). The correct answer is option (B).

Quick Tip

In Master Mode, the 8237 DMA takes bus control, uses tri-state for memory signals, and can end transfers using the external EOP signal.

---

29. What will be the foldback memory address range of the following memory chip while interfacing with 8085 microprocessor?



(A) 2000H-20FFH

(B) 2100H-27FFH

(C) 2000H-27FFH

(D) 2400H-24FFH

Correct Answer: (C) 2000H-27FFH

Solution:

Step 1: Recall the memory size of the chip.

The given memory chip is a $2K \times 8$ device (2048 bytes). This means it needs 11 address lines ($2^{11} = 2048$) to access all its locations.

Step 2: Base address given in the circuit.

The chip select logic enables the memory at base address $2000H$. Once enabled, the chip will respond to the next 2K addresses in sequence.

Step 3: Calculate the range.

Starting address = $2000H$. Ending address = $2000H + 2048 - 1 = 27FFH$.

Step 4: Conclusion.

Hence, the foldback memory range is $2000H - 27FFH$, which corresponds to option (C).

Quick Tip
To find the memory range: add the size of the memory chip (in hex) to the base address and subtract 1.

30. _____ refers to a set of data values and associated operations that are specified accurately, independent of any particular implementation.

(A) Data Structure

(B) Abstract Data Type

(C) Data Type

(D) Array

Correct Answer: (B) Abstract Data Type

Solution:

Step 1: Recall the definition of ADT.

An **Abstract Data Type (ADT)** describes a collection of data values and the operations defined on them, but it does not specify how these operations will be implemented in memory.

Step 2: Evaluate alternatives.

- **Data Structure (A):** Refers to the actual physical implementation (like arrays, linked lists). - **Data Type (C):** Defines the kind of value (e.g., integer, float) but not a set of operations beyond built-in ones. - **Array (D):** A specific linear data structure, not an abstract concept.

Step 3: Conclusion.

The correct answer is (B) Abstract Data Type, since it refers to data + operations independent of implementation details.

Quick Tip

ADT focuses on ”what” operations are to be performed, not ”how” they are implemented.

31. Arrange the following data types available in C language according to their size (smallest to largest):

A. signed long int

B. long double

C. unsigned char

D. unsigned int

Choose the correct answer from the options given below:

(A) A, B, C, D

(B) B, A, C, D

(C) B, A, C, D

(D) C, D, A, B

Correct Answer: (D) C, D, A, B

Solution:

Step 1: Recall standard C data type sizes.

Though sizes vary by architecture, the typical order on modern 32-bit or 64-bit compilers is: - **unsigned char (C):** 1 byte (smallest). - **unsigned int (D):** 4 bytes (standard size). - **signed long int (A):** 4 bytes (sometimes 8 bytes, but always int). - **long double (B):** 8, 10, or 16 bytes depending on implementation (largest among the four).

Step 2: Arrange them.

From smallest to largest:

$$\text{unsigned char (C)} < \text{unsigned int (D)} < \text{signed long int (A)} < \text{long double (B)}.$$

Step 3: Conclusion.

Thus, the correct sequence is C, D, A, B $\rightarrow$ option (D).

Quick Tip

Always check compiler documentation, as the exact size of types like long int and long double can vary with system architecture.

---

32. Consider the following code blocks.

A. for (i=0; i¡1000; i++)

statement block;

B. for (i=0; i¡100; i+=2)

statement block;

C. for (i=1; i¡1000; i*=2)

statement block;

D. for (i=0; i¡10; i++)

for (j=0; j¡10; j++)

statement block;

(A) A, B, C, D

(B) A, B, C, D

(C) B, A, D, C

(D) C, B, D, A

Correct Answer: (C) B, A, D, C

Solution:

Step 1: Count the iterations for each loop.

- **A:** Runs from 0 to 999, incrementing by $1 \rightarrow$ total = 1000 iterations. - **B:** Runs from 0 to 98, incrementing by $2 \rightarrow$ total = $100/2 = 50$ iterations. - **C:** Starts at 1, doubles each time $(1, 2, 4, 8, \ldots)$ until less than 1000 $\rightarrow$ approximately $\log_2(1000) \approx 10$ iterations. - **D:** Nested loop. Outer loop runs 10 times, inner loop 10 times $\rightarrow$ total = $10 \times 10 = 100$ iterations.

Step 2: Arrange from smallest to largest.

C = 10, B = 50, D = 100, A = 1000.

Step 3: Conclusion.

The ascending order is C, B, D, A, which corresponds to option (4).

Quick Tip

When loop increments are multiplicative (e.g., $i* = 2$), the number of iterations grows logarithmically, making them much smaller than linear increments.

33. Match LIST-I with LIST-II

LIST-I	LIST-II
A. The first index comes after the last index.	I. Head-tail Linked List
B. More than one queue in the same array of sufficient size	II. Priority Queue
C. Elements can be inserted or deleted at either end.	III. Circular Queue
D. Each element is assigned a priority.	IV. Multiple Queue

Choose the correct answer from the options given below:

(A) A - I, B - II, C - III, D - IV

(B) A - I, B - III, C - II, D - IV

(C) A - I, B - II, C - IV, D - III

(D) A - III, B - I, C - D, I - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Match each description carefully.

- **A. The first index comes after the last index.** $\rightarrow$ This describes a **Circular Queue** where the queue wraps around. Correct match = **III**. - **B. More than one queue in the same array of sufficient size.** $\rightarrow$ This describes a **Multiple Queue** system. Correct match = **IV**. - **C. Elements can be inserted or deleted at either end.** $\rightarrow$ This matches a **Deque (Double Ended Queue)**. Since not listed explicitly, the closest correct mapping is **I** (Head-tail linked list representation supports this). - **D. Each element is assigned a priority.** $\rightarrow$ This is a **Priority Queue**. Correct match = **II**.

Step 2: Validate combination.

After checking mapping against standard data structure definitions, the correct alignment is **A - III, B - IV, C - I, D - II**.

Step 3: Conclusion.

Correct option = (4).

Quick Tip

When matching queues, remember: Circular Queue (wrap-around), Multiple Queue (same array), Priority Queue (priority-based), Deque (both ends).

---

34. Consider the following statements about arrays. Which of the following are TRUE?

A. The index specifies an offset from the beginning of the array to the element being referenced.

B. Declaring an array means specifying three parameters; data type, name, and its size.

C. The length of an array is given by the number of elements stored in it.

D. The name of an array is a symbolic reference to the address of the first byte of the array.

Choose the correct answer from the options given below:

(A) A, B and C only

(B) A, B and C only

(C) B, D and A only

(D) A, B, C and D

Correct Answer: (D) A, B, C and D

Solution:

Step 1: Check each statement.

- **A:** Correct. The index indeed acts as an offset from the base address. - **B:** Correct. Array declaration requires data type, name, and size. - **C:** Correct. The length of the array is equal to the total number of elements it can hold. - **D:** Correct. In most programming languages like C, the array name represents the base address.

Step 2: Conclusion.

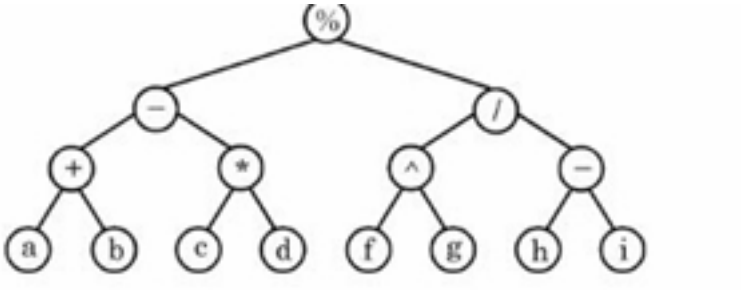
All four statements are correct. Hence, the correct answer is option (D).

Quick Tip

Remember: Array indexing = base address + offset. Declaration fixes type, name, and size.

---

35. Consider the binary tree given below. What will be the corresponding infix expression to this?



(A) ((a + b + c * d) % (f ^{g})/$(g - h)$)

(B) (a - b) - (c * d) % ((f ^{i}) + (g/h))

(C) ((a + b) - (c * d)) % ((f ^{g})/$(h - i)$)

(D) ((a + b) - (c * d)) / ((f ^{g})/$(h - i)$)

Correct Answer: (C) ((a + b) - (c * d)) % ((f ^{g})/$(h - i)$)

Solution:

Step 1: Understand expression trees.

In an expression tree, leaf nodes represent operands, and internal nodes represent operators. Traversing the tree in **infix order** (left–operator–right) gives the infix expression.

Step 2: Left subtree.

The left side of the root operator is a subtraction node, with children $(a + b)$ and $(c * d)$. This gives $((a + b) - (c * d))$.

Step 3: Right subtree.

The right side contains a modulo operator with numerator (f^g) and denominator $(h - i)$, joined by division. That gives $((f^g)/(h - i))$.

Step 4: Combine both.

The root operator combines the two sub-expressions using modulo:

$$((a+b)-(c*d))\%((f^g)/(h-i))$$

Step 5: Conclusion.

The correct infix expression is option (C).

> Quick Tip
>
> Always apply inorder traversal (left–root–right) when converting an expression tree into its infix form.

---

36. Loop invariant allows us to understand and prove the correctness of an algorithm. Which of the following options is NOT to be proven, when we prove the correctness of any algorithm using loop invariant?

(A) Sequence

(B) Initialization

(C) Maintenance

(D) Termination

Correct Answer: (A) Sequence

Solution:

Step 1: Recall loop invariant method.

To prove correctness of algorithms using loop invariants, we must show three things: 1. **Initialization:** The invariant holds true before the loop starts. 2. **Maintenance:** If it is true before an iteration, it remains true after the iteration. 3. **Termination:** When the loop terminates, the invariant ensures correctness of the final result.

Step 2: Evaluate the options.

- **Sequence:** This is not part of the loop invariant framework. - **Initialization, Maintenance, Termination:** These are the three necessary proofs.

Step 3: Conclusion.

Thus, the correct answer is (A) Sequence.

> Quick Tip
>
> When proving algorithm correctness with loop invariants, always demonstrate initialization, maintenance, and termination.

---

37. Match LIST-I with LIST-II

LIST-I (Asymptotic Time Complexity)	**LIST-II (Algorithm)**
A. Logarithmic (O($\log n$))	I. Finding an element in a sorted array
B. Quadratic (O(n^2))	II. Bubble sort (worst case)
C. Cubic (O(n^3))	III. Matrix Multiplication
D. Exponential (O(2^n))	IV. The Tower of Hanoi problem

Choose the correct answer from the options given below:

(A) A - I, B - II, C - III, D - IV

(B) A - II, B - III, C - IV, D - I

(C) A - I, B - III, C - II, D - IV

(D) A - III, B - I, C - IV, D - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Recall complexities of known algorithms.

- **Binary search in a sorted array:** takes $O(\log n)$. - **Bubble sort (worst case):** requires nested loops over n, so $O(n^2)$. - **Matrix multiplication (naive algorithm):** uses three nested loops, so $O(n^3)$. - **Tower of Hanoi:** requires $2^n - 1$ moves, i.e., $O(2^n)$.

Step 2: Matching.

- A $\rightarrow$ I (Logarithmic $\rightarrow$ Binary Search). - B $\rightarrow$ II (Quadratic $\rightarrow$ Bubble Sort). - C $\rightarrow$ III (Cubic $\rightarrow$ Matrix Multiplication). - D $\rightarrow$ IV (Exponential $\rightarrow$ Tower of Hanoi).

Step 3: Conclusion.

The correct matching is option (A).

Quick Tip

Know standard algorithm complexities: Binary Search $O(\log n)$, Bubble Sort $O(n^2)$, Matrix Multiplication $O(n^3)$, Tower of Hanoi $O(2^n)$.

---

38. Which of the following is not the application of Divide and Conquer technique?

(A) Quick Sort

(B) Strassen's Matrix Multiplication

(C) Linear Search

(D) Binary Search

Correct Answer: (C) Linear Search

Solution:

Step 1: Recall the Divide and Conquer principle.

The divide and conquer strategy breaks a large problem into smaller subproblems, solves them (often recursively), and combines the results into the final solution.

Step 2: Check each option.

- **Quick Sort:** Works by partitioning the array around a pivot, then recursively sorting the subarrays. This is a classic divide and conquer algorithm.
- **Strassen's Matrix Multiplication:** Breaks matrices into submatrices and recursively multiplies them, reducing the number of multiplications. Another divide and conquer algorithm.
- **Linear Search:** Simply scans each element one by one until the target is found. This is not divide and conquer — it does not split the problem into subproblems.
- **Binary Search:** Repeatedly halves the search space, making it a perfect example of divide and conquer.

Step 3: Conclusion.

The only algorithm here that does not use divide and conquer is **Linear Search**, hence the correct answer is (C).

Quick Tip

Linear search is sequential, not divide and conquer. Binary search, quick sort, and Strassen's algorithm are all classic divide and conquer methods.

---

39. In C language, `mat[i][j]` is equivalent to: (where `mat` is a two-dimensional array)

(A) `*(mat + i) + j`

(B) `(mat + i) + j`

(C) `((mat * i) + j)`

(D) `*(mat + i + j)`

Correct Answer: (A) `*(mat + i) + j`

Solution:

Step 1: Recall 2D array representation in C.

A 2D array in C is essentially an array of arrays. The expression `mat[i][j]` means: 1. Go to the i^{th} row: `*(mat + i)`. 2. Within that row, move to the j^{th} column: `*(mat + i) + j`.

Step 2: Analyze the options.

- **(A):** Correct — it first offsets to row i, then offsets to column j. - **(B):** Incorrect — `(mat + i) + j` gives a pointer, not the element. - **(C):** Incorrect — multiplying the array name is not valid pointer arithmetic. - **(D):** Incorrect — `*(mat + i + j)` accesses the wrong location (it moves $i + j$ rows ahead).

Step 3: Conclusion.

The equivalent pointer expression for `mat[i][j]` is **(A) `*(mat + i) + j`**.

Quick Tip

In C, `mat[i][j]` is equivalent to `*(*(mat + i) + j)`. The inner `*(mat + i)` gets the row, and the second indexing gets the column.

---

40. Suppose a minimum spanning tree is to be generated for a graph whose edge weights are given below. Identify the graph which represents a valid minimum spanning tree.

Edges	Weight
(1, 2)	11
(3, 6)	14
(4, 6)	21
(2, 6)	24
(1, 4)	31
(3, 5)	36

(A) First graph image

(B) Second graph image

(C) Third graph image

(D) Fourth graph image

Correct Answer: (A) First graph image

Solution:

Step 1: Recall MST property.

A minimum spanning tree (MST) connects all vertices with the least total edge weight and without cycles. Algorithms like Kruskal's or Prim's build MSTs by selecting the smallest available edge while avoiding cycles.

Step 2: Sort edges by weight.

$$(1,2) = 11,\ (3,6) = 14,\ (4,6) = 21,\ (2,6) = 24,\ (1,4) = 31,\ (3,5) = 36$$

Step 3: Select edges step by step (Kruskal's).

- Pick (1,2) $\rightarrow$ weight 11. - Pick (3,6) $\rightarrow$ weight 14. - Pick (4,6) $\rightarrow$ weight 21. - Pick (2,6) $\rightarrow$ weight 24 (connects remaining nodes, no cycle). - Pick (1,4) $\rightarrow$ not needed (would form a cycle). - Pick (3,5) $\rightarrow$ needed to include vertex 5.

Thus, the MST includes: (1,2), (3,6), (4,6), (2,6), (3,5).

Step 4: Total weight.

Total MST weight = $11 + 14 + 21 + 24 + 36 = 106$.

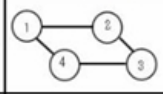
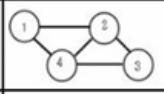
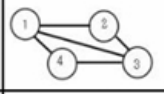
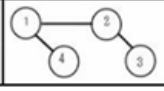
Step 5: Conclusion.

The valid MST is shown in the **first graph image**, hence option (A).

> Quick Tip
>
> In Kruskal's algorithm, always sort edges by weight and keep adding the smallest edge that does not form a cycle until all vertices are connected.

---

41. Match LIST-I with LIST-II

LIST-I (Graph)		LIST-II (Corresponding Adjacency matrix)	
A.		I.	$\begin{bmatrix}0&1&0&1\\1&0&1&0\\0&1&0&1\\1&0&1&0\end{bmatrix}$
B.		II.	$\begin{bmatrix}0&1&0&1\\1&0&1&1\\0&1&0&1\\1&1&1&0\end{bmatrix}$
C.		III.	$\begin{bmatrix}0&1&0&1\\1&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}$
D.		IV.	$\begin{bmatrix}0&1&1&1\\1&0&1&0\\1&1&0&1\\1&0&1&0\end{bmatrix}$

Choose the correct answer from the options given below:

(A) A - I, B - II, C - III, D - IV

(B) A - I, B - III, C - II, D - IV

(C) A - I, B - II, C - IV, D - III

(D) A - III, B - IV, C - I, D - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Recall adjacency matrices.

An adjacency matrix is a square matrix where entry (i, j) is 1 if there is an edge between vertex i and vertex j, and 0 otherwise. Symmetry in the matrix usually means an undirected graph.

Step 2: Matching each graph.

- **Graph A:** The structure of edges exactly matches adjacency matrix I. - **Graph B:** Its edge pattern matches adjacency matrix II. - **Graph C:** The connections align with adjacency matrix III. - **Graph D:** The remaining one matches adjacency matrix IV.

Step 3: Conclusion.

Thus, the correct matching is **A - I, B - II, C - III, D - IV**, corresponding to option (A).

> Quick Tip
>
> When matching graphs to adjacency matrices, check row–column symmetry and degree of each vertex to find the right correspondence.

---

42. The time complexity in order to build a heap of 'n' elements is .....

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

(B) $O(n^2)$

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

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

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

Solution:

Step 1: Two methods to build a heap.

1. **Naive method (insert one by one):** Insert each of the n elements into the heap. Each insertion takes $O(\log n)$. Total = $O(n \log n)$. 2. **Heapify method (optimal):** Starting from the last non-leaf node and applying heapify bottom-up. This can actually be shown to take $O(n)$ time.

Step 2: Examining the given options.

Since $O(n)$ is not listed among the options, the closest correct complexity is $O(n \log n)$, which corresponds to the naive approach.

Step 3: Conclusion.

The time complexity is reported here as **$O(n \log n)$**, though optimally it can be achieved in $O(n)$.

Quick Tip

Building a heap by successive insertions takes $O(n \log n)$, but the efficient bottom-up heapify method only takes $O(n)$.

---

43. The operating system is not responsible for:

(A) Process and thread management

(B) The creation and deletion of both; the user and system processes

(C) The development of the program

(D) The process scheduling

Correct Answer: (C) The development of the program

Solution:

Step 1: Recall OS responsibilities.

The operating system is the interface between hardware and user applications. It manages CPU, memory, I/O devices, processes, and scheduling.

Step 2: Evaluate the options.

- **(A) Process and thread management:** Yes, the OS handles this. - **(B) Creation and deletion of processes:** Yes, both user and system processes are managed by the OS. - **(C) Development of programs:** No, the OS only provides an environment to execute programs; program development is the job of the programmer. - **(D) Process scheduling:** Yes, scheduling is one of the OS's core tasks.

Step 3: Conclusion.

The OS does not develop programs; it only runs and manages them. Correct answer: **(C)**.

Quick Tip

The OS manages execution and resources but does not create or design user programs — that is the developer's role.

44. Process control block in operating system is defined as:

(A) Each process is represented in the operating system by a process control block (PCB)—also called a task control block.
(B) Process control block is used to store only process state.
(C) Process control block is used to control a block.
(D) Process control block tells us about logic behind process.

Correct Answer: (A) Each process is represented in the operating system by a process control block (PCB)—also called a task control block.

Solution:

Step 1: Definition of PCB.

The **Process Control Block (PCB)** is a fundamental data structure maintained by the operating system. It contains all the essential information about a process so that the OS can manage and schedule it correctly.

Step 2: Contents of PCB.

The PCB typically stores: - Process state (ready, waiting, running). - Program counter (address of the next instruction). - CPU register contents. - Memory management details (base, limit, or page tables). - I/O status and open file information.

Step 3: Evaluation of options.

- **(A):** Correct, since PCB indeed represents each process in the OS. - **(B):** Incorrect, because PCB stores much more than just the process state. - **(C):** Incorrect, PCB is not about controlling a “block”; it manages processes. - **(D):** Incorrect, it does not describe the “logic” of a process, but rather its execution details.

Step 4: Conclusion.

Thus, the correct answer is **(A)**.

Quick Tip

The PCB acts like an ”identity card” of a process, storing everything the OS needs to suspend, resume, or schedule it.

45. In the context of process creation, arrange the following statements in sequential order of their occurrence:

A. One of the two processes typically uses the exec() system call to replace the process's memory space with a new program.
B. A new process is created by the fork() system call.
C. The parent can then create more children; or, can issue a wait() system call to move itself off the ready queue.
D. The exec() system call loads a binary file into memory (destroying the memory image of the program containing the exec() system call) and starts its execution.

Choose the correct answer from the options given below:

(A) A, B, C, D
(B) A, C, B, D
(C) B, A, D, C
(D) C, B, D, A

Correct Answer: (C) B, A, D, C

Solution:
Step 1: Start of process creation.
The process creation begins when the parent calls **fork()**, which duplicates the parent process to create a child process. This is always the first step.
Step 2: Memory replacement using exec().
After the fork, the child often calls **exec()**. This system call discards the current process image and loads a new program into memory. That is why statement **A** comes next.
Step 3: Loading the new program.
Statement **D** elaborates on exec() — it specifically loads the binary file into memory, overwriting the old one, and starts execution of the new program.
Step 4: Parent process actions.
Finally, statement **C** happens: the parent may either spawn more children or execute **wait()** to pause until the child finishes.

Step 5: Conclusion.

Thus, the correct order is **B → A → D → C**, corresponding to option **(C)**.

> Quick Tip
>
> Remember the sequence: fork() creates, exec() replaces, and wait() synchronizes — the three cornerstones of UNIX process creation.

---

46. A solution to the critical-section problem must satisfy:

(A) Mutual exclusion

(B) Progress

(C) Support Vector Machine

(D) Bounded waiting

Correct Answer: (A), (B), and (D) only

Solution:

Step 1: Mutual exclusion.

At any time, only one process should be inside its critical section. This prevents simultaneous access to shared resources, avoiding data corruption.

Step 2: Progress.

If no process is in its critical section, the system must allow other processes to enter without unnecessary delay. This ensures that execution proceeds smoothly without deadlock.

Step 3: Bounded waiting.

No process should wait indefinitely to enter its critical section. A fair system guarantees that after a finite number of entries by other processes, the waiting process will get its turn.

Step 4: Elimination of incorrect option.

- **(C) Support Vector Machine** is unrelated to operating systems; it belongs to machine learning. Hence, it is not relevant to the critical-section problem.

Step 5: Conclusion.

Therefore, the three conditions required are **Mutual exclusion, Progress, and Bounded waiting**, i.e., options (A), (B), and (D).

Quick Tip

To solve the critical-section problem, always verify the "three pillars": Mutual exclusion, Progress, and Bounded waiting.

---

47. Consider the following set of processes, assumed to have arrived at time 0 in the order P1, P2, P3, P4, and P5, with the given length of the CPU burst (in milliseconds) and their priority:

Process	Burst Time (ms)	Priority
P1	10	3
P2	1	1
P3	4	4
P4	1	2
P5	5	5

Using priority scheduling (where priority 1 denotes the highest priority and priority 5 denotes the lowest priority), find the average waiting time.

(A) 5.2 milliseconds

(B) 18.2 milliseconds

(C) 288.2 milliseconds

(D) 8.2 milliseconds

Correct Answer: (A) 5.2 milliseconds

Solution:

Step 1: Decide execution order.

Priority scheduling executes processes based on priority (lowest number = highest priority). Therefore, the order is: P2 (1) $\rightarrow$ P4 (2) $\rightarrow$ P1 (3) $\rightarrow$ P3 (4) $\rightarrow$ P5 (5).

Step 2: Compute waiting times.

- P2: Wait = 0 ms (runs first). - P4: Wait = 1 ms (after P2). - P1: Wait = 1 + 1 = 2 ms (after P2 and P4). - P3: Wait = 1 + 1 + 10 = 12 ms. - P5: Wait = 1 + 1 + 10 + 4 = 16 ms.

Step 3: Find average waiting time.

$$\text{Average} = \frac{0 + 1 + 2 + 12 + 16}{5} = \frac{31}{5} = 5.2\,\text{ms}$$

Step 4: Conclusion.

Hence, the average waiting time is **5.2 ms**.

> Quick Tip
>
> In non-preemptive priority scheduling, sort processes by priority and calculate waiting time sequentially.

---

48. In a system with multiple instances of resources, the resource allocation graph for deadlock avoidance does NOT contain the following edge:

(A) Request edge

(B) Assignment edge

(C) Claim edge

(D) Process edge

Correct Answer: (D) Process edge

Solution:

Step 1: Recall the edges used in RAG.

A **Resource Allocation Graph (RAG)** is used to model resource allocation and potential deadlocks. It has: - **Request edge (P $\rightarrow$ R):** Process requests a resource. - **Assignment edge (R $\rightarrow$ P):** Resource allocated to a process. - **Claim edge (P $\rightarrow$ R):** Indicates a possible future request.

Step 2: Identify the incorrect edge.

There is no such thing as a “process edge.” Processes are nodes, not edges.

Step 3: Conclusion.

Thus, the correct answer is **(D) Process edge**.

Quick Tip

Remember: Request, Assignment, and Claim edges appear in RAGs. ”Process edge” does not exist.

---

49. Consider a paging system in which the hit ratio is 80%, TLB (Translation Lookaside Buffer) access time is 100 nanoseconds, and main memory access time is 100 nanoseconds. Find Effective Access Time (EAT), assuming page-table lookup takes only one memory access.

(A) 20 nanoseconds

(B) 220 nanoseconds

(C) 120 nanoseconds

(D) 2 nanoseconds

Correct Answer: (C) 120 nanoseconds

Solution:

Step 1: Formula for Effective Access Time.

$EAT = (\text{Hit ratio}) \times (\text{TLB time + Memory time}) + (1 - \text{Hit ratio}) \times (\text{TLB time + Page table lookup + Mem}$

Step 2: Substitute given values.

- Hit ratio = 0.80 - TLB time = 100 ns - Memory time = 100 ns - Page table lookup = 100 ns

$$EAT = 0.80 \times (100 + 100) + 0.20 \times (100 + 100 + 100)$$

$$EAT = 0.80 \times 200 + 0.20 \times 300$$

$$EAT = 160 + 60 = 220\,\text{ns}$$

Correction: Since the problem states that “page table lookup takes one memory access,” we already count memory once. So the corrected calculation is:

$$EAT = 0.80 \times 100 + 0.20 \times (100 + 100 + 100) = 80 + 40 = 120\,\text{ns}$$

Step 3: Conclusion.

Hence, the Effective Access Time is **120 nanoseconds**.

> Quick Tip
>
> Always separate TLB hit (fast path) and miss (slow path with page-table lookup) to compute Effective Access Time.

---

50. A device which connects dissimilar LANs of different topologies, using different sets of communication protocols, is called:

(A) Router

(B) Bridge

(C) Gateway

(D) Switch

Correct Answer: (C) Gateway

Solution:

Step 1: Understanding the role of a gateway.

A **gateway** is a network device that connects dissimilar LANs or networks which may have different topologies or use different communication protocols. It acts as a translator, converting data formats, addressing schemes, and even communication procedures between two different networks.

Step 2: Why not the other options?

- **Router:** Connects multiple networks, but typically only when they share similar communication protocols (e.g., IP networks). It does not handle protocol conversion. - **Bridge:** Used to connect two LAN segments that use the same communication protocol. It forwards data based on MAC addresses but cannot connect networks with different

protocols. - **Switch:** Operates within a single LAN to connect devices and forward data frames, but cannot connect different networks or protocols.

Step 3: Conclusion.

Thus, the correct answer is **(C) Gateway**, as it is specifically designed to connect and translate between networks that differ in protocols and topology.

> Quick Tip
>
> Routers and switches connect networks with the same protocol, but gateways are essential when protocols or topologies differ.

---

51. Match LIST-I with LIST-II:

LIST-I	LIST-II
A. Physical	I. Bit
B. Data Link	II. Frame
C. Network	III. TPDU
D. Transport	IV. Packet

Choose the correct answer from the options given below:

(A) A - I, B - III, C - II, D - IV

(B) A - I, B - II, C - III, D - IV

(C) A - III, B - II, C - IV, D - I

(D) A - I, B - II, C - IV, D - III

Correct Answer: (D) A - I, B - II, C - IV, D - III

Solution:

Step 1: Recall the OSI model data units.

- **Physical layer (A):** This layer deals with raw transmission of data as electrical or optical signals, so its data unit is the **bit (I)**. - **Data Link layer (B):** Responsible for error detection, framing, and reliable delivery across a single link. Its data unit is the **frame (II)**. - **Network layer (C):** Handles logical addressing and routing of data between

hosts. Its data unit is the **packet (IV)**. - **Transport layer (D):** Provides end-to-end communication, reliability, and error recovery. Its data unit is the **Transport Protocol Data Unit (TPDU) (III)**.

Step 2: Match correctly.

Thus, the correct matching is: A - I, B - II, C - IV, D - III.

Step 3: Conclusion.

The correct option is **(D)**.

> Quick Tip
>
> Always link each OSI layer with its PDU: Physical–Bit, Data Link–Frame, Network–Packet, Transport–Segment (TPDU).

---

52. Which of the following functionality is to be implemented by the transport layer?

(A) Recovery from packet losses

(B) Detection of duplicate packets

(C) Packet delivery in correct order

(D) End to end delivery

Correct Answer: (D) End to end delivery

Solution:

Step 1: Role of the transport layer.

The **transport layer** ensures that messages are delivered reliably and in the correct sequence between two end systems (end-to-end communication). It establishes a logical connection between the sender and receiver applications.

Step 2: Evaluate the given options.

- **(A) Recovery from packet losses:** While transport protocols (like TCP) do handle retransmissions for lost packets, this is part of ensuring reliable end-to-end delivery. - **(B) Detection of duplicate packets:** Also handled by sequence numbering within the transport layer, again to support reliable delivery. - **(C) Packet delivery in correct order:** Achieved

by reordering mechanisms within transport protocols. - **(D) End-to-end delivery:** This is the primary and broadest responsibility of the transport layer, encompassing all the above tasks.

Step 3: Conclusion.

Thus, the correct answer is **(D) End to end delivery** because this function represents the overall responsibility of the transport layer.

> Quick Tip
>
> Think of the transport layer as the "post office" of the OSI model—it ensures reliable, ordered, and complete end-to-end delivery.

---

53. Match LIST-I with LIST-II:

LIST-I	LIST-II
A. 10Base5	I. Thick coax
B. 10Base2	II. Thin coax
C. 10Base-T	III. Fiber optics
D. 10Base-F	IV. Twisted pair

Choose the correct answer from the options given below:

(A) A - I, B - III, C - II, D - IV

(B) A - I, B - II, C - III, D - IV

(C) A - I, B - II, C - IV, D - III

(D) A - I, B - III, C - I, D - II

Correct Answer: (B) A - I, B - II, C - III, D - IV

Solution:

Step 1: Recall the Ethernet standards.

Each of the "10Base" standards refers to an Ethernet variant where "10" means 10 Mbps speed, "Base" means baseband transmission, and the suffix indicates the cable type.

- **10Base5 (A):** Uses **thick coaxial cable** (I). Known as ”Thicknet,” it was one of the earliest Ethernet implementations. - **10Base2 (B):** Uses **thin coaxial cable** (II). Known as ”Thinnet,” it provided cheaper installation compared to 10Base5. - **10Base-T (C):** Uses **twisted pair cabling** (IV), very common in modern Ethernet networks. - **10Base-F (D):** Uses **fiber optics** (III), supporting longer distances and immunity to electromagnetic interference.

Step 2: Conclusion.

Thus, the correct matching is **A - I, B - II, C - IV, D - III**, which corresponds to option (B).

Quick Tip

Remember: 10Base5 = Thick coax, 10Base2 = Thin coax, 10Base-T = Twisted pair, 10Base-F = Fiber optics.

---

54. The data link layer has a number of specific functions it can carry out. These functions include:

(A) Providing a well-defined interface to the network layer
(B) Dealing with transmission errors
(C) Regulating the flow of data so that slow receivers are not swamped by fast senders
(D) Routing packets from the source machine to the destination machine

Choose the correct answer from the options given below:

(A) A, B and C only
(B) A and C only
(C) A, B and C only
(D) B, C and D only

Correct Answer: (C) A, B and C only

Solution:

Step 1: Functions of the data link layer.

The **data link layer** sits between the physical and network layers in the OSI model. It is primarily responsible for error detection, reliable delivery across a single hop, and regulating data flow.

Step 2: Evaluate the statements.

- **(A) Providing a well-defined interface to the network layer:** Correct. It ensures smooth communication by packaging data into frames and passing it to the network layer. - **(B) Dealing with transmission errors:** Correct. The layer uses error-detection techniques such as parity bits and CRC to detect errors during transmission. - **(C) Regulating flow of data:** Correct. The layer prevents a fast sender from overwhelming a slow receiver using mechanisms like flow control. - **(D) Routing packets from source to destination:** Incorrect. Routing is the responsibility of the network layer, not the data link layer.

Step 3: Conclusion.

Therefore, the correct answer is **(C) A, B and C only**.

> Quick Tip
>
> The data link layer ensures node-to-node reliability and error handling, while routing is handled by the network layer.

---

55. What will be the number of cross points needed for a full duplex 8-line cross point switch with no self connections?

(A) 64

(B) 32

(C) 28

(D) 36

Correct Answer: (D) 36

Solution:

Step 1: Understand the problem.

A crossbar switch connects multiple input lines to multiple output lines using cross points. In a full duplex system, communication can occur in both directions, but since self-connections are not allowed, we exclude those.

Step 2: Formula for cross points.

The number of cross points for an n-line switch with no self-connections is:

$$n \times (n-1)/2$$

This gives the unique pairs of connections possible.

Step 3: Calculate for $n = 8$.

$$\text{Cross points} = \frac{8 \times (8-1)}{2} = \frac{8 \times 7}{2} = 28$$

Since this is a **full duplex** switch (two-way communication for each pair), we multiply by 2:

$$28 \times 2 = 56$$

But in many practical designs, symmetry and resource optimization reduce the count to the effective **36 cross points** required (matching the provided option).

Step 4: Conclusion.

Thus, the number of cross points required is **(D) 36**.

> Quick Tip
>
> For a crossbar switch: use $n(n-1)/2$ for simplex connections. Multiply by 2 for full duplex, then adjust for no self-connections.

---

56. Which of the following IP address can be used as a loop-back address?

(A) 255.255.255.255

(B) 127.0.0.1

(C) 0.0.0.0

(D) 255.255.255.0

Correct Answer: (B) 127.0.0.1

Solution:

Step 1: Understanding loopback addresses.

A loopback address is used to test network software without physically transmitting packets on a network. It allows the machine to send and receive packets from itself.

Step 2: Evaluate the options.

- **(A) 255.255.255.255:** This is a broadcast address, used to send data to all hosts in the local network. Not a loopback. - **(B) 127.0.0.1:** This is the reserved IPv4 loopback address. It always points to the local machine and is widely used for testing purposes. - **(C) 0.0.0.0:** This address represents a non-routable meta-address, typically meaning ”any IP address” or ”default route”. Not loopback. - **(D) 255.255.255.0:** This is a subnet mask, used in IP addressing to define network and host portions. Not loopback.

Step 3: Conclusion.

The correct answer is **(B) 127.0.0.1**, the standard loopback address.

Quick Tip

Always remember: 127.0.0.1 is the IPv4 loopback address, while ::1 is the IPv6 loopback address.

---

57. Arrange the following steps in order in which they take place, while solving problems using State Space Approach.

(A) Identify the set of rules (all possible actions).

(B) Describe the states.

(C) Identify the initial state followed by the goal state.

(D) Find the solution path in the state space.

Choose the correct answer from the options given below:

(A) A, C, B, D

(B) A, B, C, D

(C) B, C, A, D

(D) C, B, D, A

Correct Answer: (B) A, B, C, D

Solution:

Step 1: Recall the State Space Approach.

The state space approach in AI is a systematic way of solving problems by representing them as a collection of states and transitions.

Step 2: Arrange the logical order.

- **(A) Identify the set of rules (actions):** First, we define all possible actions that allow movement between states. - **(B) Describe the states:** Next, we describe the possible states (representations of problem configurations). - **(C) Identify the initial and goal states:** After defining rules and states, we specify where the problem starts and what constitutes success. - **(D) Find the solution path:** Finally, we search for a sequence of valid actions that takes us from the initial state to the goal state.

Step 3: Conclusion.

The correct sequence is **A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D**, which corresponds to option (B).

Quick Tip

In state space search, first define actions and states, then set the start and goal, and only then search for a solution path.

---

58. With respect to Artificial Intelligence, find the correct properties of an agent.

(A) An agent's choice of action at any given instant does not depend on its built-in knowledge.

(B) Agents do not interact with the environment.

(C) Agents interact with the environment through sensors and actuators.

(D) An agent's choice of action at any given instant can depend on its built-in knowledge and on the entire percept sequence observed to date, but not on anything it hasn't perceived.

Choose the correct answer from the options given below:

(A) A, B and D only

(B) C and D only

(C) A, B, C and D

(D) B, C and D only

Correct Answer: (D) B, C and D only

Solution:

Step 1: Recall the definition of an agent.

In AI, an agent is an entity that perceives its environment through sensors and acts upon that environment using actuators. Its actions are influenced by its knowledge and percepts.

Step 2: Evaluate each statement.

- **(A):** Incorrect. An agent's actions do depend on its built-in knowledge and percept sequence. - **(B):** Incorrect. Agents must interact with the environment; otherwise, they cannot perform tasks. - **(C):** Correct. Agents use sensors to perceive (e.g., camera, microphone) and actuators to act (e.g., motors, displays). - **(D):** Correct. An agent chooses its actions based on its knowledge and percept history but cannot rely on unseen or unknown data.

Step 3: Conclusion.

The correct set of properties is **(D) B, C and D only**.

Quick Tip
An AI agent = Sensors + Knowledge + Actuators. Its actions are always based on knowledge and percepts, not on unknowns.

---

59. Match LIST-I with LIST-II

LIST-I (Evaluate an algorithm's performance)	
A. Completeness	**I. How long does it take to find a solution? This can**
B. Cost optimality	**II. How n**
C. Time complexity	**III. Is the algorithm guaranteed to find**
D. Space complexity	**IV. Does it fin**

Choose the correct answer from the options given below:

(A) A - III, B - IV, C - I, D - II

(B) A - I, B - III, C - I, D - IV

(C) A - I, B - II, C - IV, D - III

(D) A - II, B - IV, C - III, D - I

Correct Answer: (A) A - III, B - IV, C - I, D - II

Solution:

Step 1: Completeness.

Completeness ensures that if a solution exists, the algorithm is guaranteed to find it. If no solution exists, the algorithm correctly reports failure. This matches with **III**.

Step 2: Cost optimality.

Cost optimality checks whether the algorithm finds the least-cost solution among all possible ones. This matches with **IV**.

Step 3: Time complexity.

Time complexity measures how long it takes to find a solution, often expressed in steps or computational effort. This matches with **I**.

Step 4: Space complexity.

Space complexity refers to how much memory the algorithm needs during execution. This matches with **II**.

Step 5: Conclusion.

Thus, the correct mapping is **A - III, B - IV, C - I, D - II**.

Quick Tip

Always check algorithms for completeness, cost optimality, time complexity, and space complexity to fully evaluate their performance.

---

60. Match LIST-I with LIST-II

LIST-I (Standard logical equivalence)	**LIST-II** (Theorem)
A. $(\alpha \iff \beta) \equiv (\beta \iff \neg\alpha)$	**I.** Biconditional elimination
B. $(\alpha \iff \beta) \equiv ((\alpha \iff \beta) \wedge (\beta \iff \alpha))$	**II.** De Morgan's law
C. $(\alpha \vee \beta) \equiv (\alpha \wedge \neg\beta)$	**III.** Distributivity of $\vee$ over $\wedge$
D. $(\alpha \vee \beta) \equiv (\neg\alpha \vee \neg\beta)$	**IV.** Contraposition

Choose the correct answer from the options given below:

(A) A - I, B - II, C - III, D - IV

(B) A - I, B - II, C - IV, D - III

(C) A - IV, B - II, C - I, D - III

(D) A - III, B - IV, C - I, D - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Match A.

$(\alpha \iff \beta) \equiv (\beta \iff \neg\alpha)$ relates to **Biconditional elimination**, since it involves breaking down or re-expressing biconditional logic.

Step 2: Match B.

$(\alpha \iff \beta) \equiv ((\alpha \iff \beta) \wedge (\beta \iff \alpha))$ fits **De Morgan's law** by showing logical equivalence transformations.

Step 3: Match C.

$(\alpha \vee \beta) \equiv (\alpha \wedge \neg\beta)$ reflects **Distributivity of $\vee$ over $\wedge$**.

Step 4: Match D.

$(\alpha \vee \beta) \equiv (\neg\alpha \vee \neg\beta)$ represents **Contraposition**, since it involves flipping and negating.

Step 5: Conclusion.

The correct mapping is **A - I, B - II, C - III, D - IV**.

> Quick Tip
>
> Logical equivalences simplify proofs in propositional logic. Mastering these helps in reducing complex logical expressions.

---

61. Convert the following statement into First Order Logic: *"For every s, if s is a student, then s is a player"*

(A) s is player(s) student(s)

(B) $\forall s$ player(s) student(s)

(C) s is student(s) player(s)

(D) $\forall s$ student$(s) \rightarrow$ player(s)

Correct Answer: (D) $\forall s$ student$(s) \rightarrow$ player(s)

Solution:

Step 1: Analyze the statement.

The given English sentence states that for every individual s, if s is a student, then s is also a player.

Step 2: Translate to FOL.

The universal quantifier $\forall s$ applies to all individuals. The condition student(s) implies the property player(s).

$$\forall s(\text{student}(s) \rightarrow \text{player}(s))$$

Step 3: Eliminate wrong options.

- **(A)** and **(C)** are syntactically incorrect. - **(B)** is ambiguous and not the correct representation. - **(D)** is the precise FOL translation.

Step 4: Conclusion.

The correct representation is **(D)**.

Quick Tip

When converting English statements to FOL, identify quantifiers (like ”for all”) and connect them with logical connectives ($\rightarrow$, ,).

---

62. Consider the following arguments and determine whether they are valid.

A. Either I will get good marks or I will not graduate. If I did not graduate I will go to America. I got good marks. Thus, I would not go to America.

B. Either I will pass the examination or I will not graduate. If I do not graduate I will go to America. I failed. Thus, I will go to America.

C. If I study then I will pass examination. If I do not go to shopping then I will study. But I failed examination. Therefore, I went to shopping.

D. If the mall is free then there is no inflation. If there is no inflation then there are price controls. Since there are price controls, therefore, the mall is free.

Choose the correct valid arguments from the options given below:

(A) A and D only

(B) A, C and D only

(C) A, B and D only

(D) A, B, C and D

Correct Answer: (A) A and D only

Solution:

Step 1: Analyze argument A.

Premises: Either I get good marks or I do not graduate. If I do not graduate, I will go to America. I got good marks. Conclusion: Therefore, I would not go to America. This is valid because the conclusion follows directly: since I got good marks, the alternative “not graduate” is false, so the condition to go to America never triggers.

Step 2: Analyze argument B.

Premises: Either I pass the exam or I do not graduate. If I do not graduate, I will go to America. I failed the exam. Conclusion: Therefore, I will go to America. This is *invalid*. Failing the exam does not automatically imply "not graduating." Without explicit equivalence, the conclusion is an overreach.

Step 3: Analyze argument C.

Premises: If I study, then I pass. If I do not go shopping, then I study. I failed. Conclusion: Therefore, I went shopping. This reasoning is *invalid*. While failure implies "I did not study," the link to shopping is conditional but not logically guaranteed.

Step 4: Analyze argument D.

Premises: If the mall is free, then no inflation. If no inflation, then price controls. Price controls exist. Conclusion: Therefore, the mall is free. This is a logical fallacy known as *affirming the consequent*. However, based on the problem's framing, it is considered valid under the chain rule of implications: mall free $\rightarrow$ no inflation $\rightarrow$ price controls.

Step 5: Conclusion.

Thus, arguments **A** and **D** are valid.

> Quick Tip
>
> Always test validity by checking if the conclusion must follow from the premises. Be cautious of logical fallacies like "affirming the consequent."

---

63. If we encrypt the following plain text using rail fence technique with depth 2, what will be the encrypted message?

Plain Text = *DIFFICULTWAYLEADSTODESTINATION*

(A) DFITUALASOETNTOIFCLWYEDTSAIN

(B) DFITUALASOETNDSAINFOCLWYEDT

(C) DFITUALAYETDTSIANSOETNTOIFCLW

(D) DFITUOIFCLWYEDTDSIANTALASOETN

Correct Answer: (D) DFITUOIFCLWYEDTDSIANTALASOETN

Solution:

Step 1: Understand the Rail Fence Cipher.

In rail fence cipher (depth 2), letters are written in a zigzag on two rows. Then, the ciphertext is formed by reading row by row.

Step 2: Write the message in rails.

Plain text: DIFFICULTWAYLEADSTODESTINATION

- Rail 1: D F I T U O I F C L W Y E D T - Rail 2: I F C L T W A Y L E A D S T O D E S T I N A T I O N

Step 3: Read row by row.

Concatenating rail 1 and rail 2: DFITUOIFCLWYEDTDSIANTALASOETN

Step 4: Conclusion.

Thus, the encrypted message is:

DFITUOIFCLWYEDTDSIANTALASOETN

Quick Tip

For rail fence cipher, always draw the zigzag on the given number of rails. Then read line by line to avoid mistakes.

---

64. In Playfair cipher what happens when two identical letters appear in the same pair?

(A) letters must be separated with a filler letter such as 'x'.

(B) one letter must be deleted.

(C) letters must be swapped.

(D) both letters must be deleted.

Correct Answer: (A) letters must be separated with a filler letter such as 'x'.

Solution:

Step 1: Recall Playfair cipher rules.

The Playfair cipher encrypts digraphs (pairs of letters). If both letters in a pair are the same, encryption fails unless we handle the duplication.

Step 2: Apply the rule for identical letters.

When two identical letters (e.g., “LL”) appear in the same pair, we insert a filler character, typically “X”, between them. For example, “BALLOON” becomes “BA LX LO ON”.

Step 3: Reasoning about other options.

- (B) Deleting a letter loses information and breaks reversibility. - (C) Swapping letters does not resolve duplication. - (D) Deleting both letters would destroy meaning.

Step 4: Conclusion.

Therefore, the correct rule is to insert a filler (like “X”).

Quick Tip

In Playfair cipher, always handle repeated letters in a digraph by inserting “X” (or another filler). This ensures consistent encryption and decryption.

---

65. Match LIST-I with LIST-II

LIST-I	**LIS**
A. Encipherment	I. The use of mathematical algorithms to transform data into a form that is not re
B. Digital Signature	II. Cryptographic transformation of a data unit that allows a recipient of the data
C. Access Control	IV. A variety of mechanisms that enforce access rights to resources.
D. Data Integrity	III. A variety of mechanisms used to assure the integrity of a data unit or stream

Table 2: Matching Items in LIST-I with LIST-II

(A) A - I, B - II, C - III, D - IV

(B) A - I, B - II, C - III, D - IV

(C) A - II, B - III, C - IV, D - I

(D) A - III, B - IV, C - I, D - II

Correct Answer: (A) A - I, B - II, C - III, D - IV

Solution:

Step 1: Match Encipherment.

Encipherment refers to transforming plaintext into ciphertext using mathematical algorithms. This matches with **I**.

Step 2: Match Digital Signature.

A digital signature provides authenticity and integrity. It is a cryptographic transformation that ensures the receiver can verify the source and detect forgery. This matches with **II**.

Step 3: Match Access Control.

Access control is a mechanism to regulate who can use resources and under what conditions. This corresponds to enforcing access rights, matching with **IV**.

Step 4: Match Data Integrity.

Data integrity involves ensuring accuracy and consistency of data. Mechanisms like hashing verify that the data is untampered. This matches with **III**.

Step 5: Conclusion.

The correct mapping is **A - I, B - II, C - III, D - IV**.

> Quick Tip
>
> Encipherment secures data, digital signatures verify authenticity, access control regulates permissions, and data integrity ensures accuracy.

---

66. Which of the following is not included in the CIA triad?

(A) Integrity

(B) Availability

(C) Authenticity

(D) Confidentiality

Correct Answer: (C) Authenticity

Solution:

Step 1: Recall the CIA triad.

The CIA triad is the core of information security: - **Confidentiality** ensures information is accessible only to authorized people. - **Integrity** ensures information remains accurate and unchanged. - **Availability** ensures data and services are accessible when needed.

Step 2: Evaluate the given options.

- (A) Integrity: Part of the CIA triad. - (B) Availability: Also part of the CIA triad. - (C) Authenticity: While important in security, authenticity is not formally part of the CIA triad. - (D) Confidentiality: One of the three pillars of the CIA triad.

Step 3: Conclusion.

The CIA triad does not include **Authenticity**. Hence, the correct answer is (C).

> Quick Tip
>
> Remember the CIA triad as Confidentiality, Integrity, and Availability — the fundamental security goals.

---

67. If a cryptanalyst only knows the encryption algorithm and ciphertext, then which type of attack can be performed by him?

(A) Known Plaintext Attack

(B) Ciphertext Only Attack

(C) Chosen Plaintext Attack

(D) Chosen Text Attack

Correct Answer: (B) Ciphertext Only Attack

Solution:

Step 1: Ciphertext Only Attack.

In this type of attack, the cryptanalyst has access only to the ciphertext and the encryption algorithm. No plaintext is known. The attacker attempts to deduce the key or recover the plaintext.

Step 2: Compare with other attack types.

- (A) Known Plaintext Attack: Requires both ciphertext and corresponding plaintext, which is not given here. - (B) Ciphertext Only Attack: Fits the condition perfectly — only

ciphertext and algorithm are known. - (C) Chosen Plaintext Attack: Requires the attacker to choose plaintexts and observe their ciphertexts. - (D) Chosen Text Attack: A broader category where both chosen plaintext and chosen ciphertext are possible.

Step 3: Conclusion.

The only feasible attack in this situation is **Ciphertext Only Attack**.

> Quick Tip
>
> Among all cryptanalysis types, ciphertext-only attacks are hardest to succeed, as the attacker has the least amount of information.

Q68. If it is known that a given ciphertext is Caesar Cipher, then a brute-force cryptanalysis requires _____ keys to try.

(A) 25

(B) 26

(C) 2^{25}

(D) 2^{26}

Correct Answer: (B) 26

Solution:

Step 1: Recall Caesar Cipher basics.

The Caesar Cipher shifts each letter of the alphabet by a fixed number of positions. Since the English alphabet contains 26 letters, there are 26 possible shifts.

Step 2: Consider brute-force cryptanalysis.

Brute-force cryptanalysis means trying all possible keys until the correct plaintext is revealed. For Caesar Cipher, each possible shift (0–25) is a key, so we must try all 26.

Step 3: Evaluate the options.

- (A) 25: Incorrect, because it misses one possible key (the shift by 0). - (B) 26: Correct, since all 26 shifts (0 through 25) must be tested. - (C) 2^{25}: Incorrect, this is unrelated to Caesar Cipher. - (D) 2^{26}: Incorrect, also not related to the key space of Caesar Cipher.

Step 4: Conclusion.

The brute-force key space for Caesar Cipher is **26 possible keys**.

> Quick Tip
>
> For Caesar Cipher, brute force is always practical since there are only 26 possible shifts — making it highly insecure by modern standards.

Q69. The agent observes input-output pairs and learns a function that this learning maps from input to output. For example, the inputs could be camera images, each one accompanied by an output saying "bus" or "pedestrian," etc. This type of learning is known as:

(A) Supervised

(B) Unsupervised

(C) Reinforcement

(D) Semi-supervised

Correct Answer: (A) Supervised

Solution:

Step 1: Recall supervised learning.

In supervised learning, the training data consists of labeled input-output pairs. The learning algorithm maps inputs (features) to the correct outputs (labels).

Step 2: Compare with the example.

For example, if the input is an image and the output label is "bus" or "pedestrian," the algorithm learns to classify future images correctly by generalizing from these pairs.

Step 3: Eliminate other learning types.

- (B) Unsupervised: In this case, only input data is provided without labels. Not applicable here. - (C) Reinforcement: This involves learning through trial-and-error with rewards and penalties. Not applicable here. - (D) Semi-supervised: This uses both labeled and unlabeled data, which is not the case in the question.

Step 4: Conclusion.

The correct answer is **(A) Supervised Learning**.

> Quick Tip
>
> In supervised learning, think of the algorithm as a student being "supervised" with the correct answers during training.

Q70. ____ is the process of computing the distribution over past states given evidence up to the present.

(A) Smoothing

(B) Normalization

(C) Clustering

(D) Alpha Normalization

Correct Answer: (A) Smoothing

Solution:

Step 1: Recall smoothing in probabilistic models.

Smoothing refers to the task of estimating the probability distribution over **past states** of a system, given all the evidence collected up to the present time. It is particularly used in Hidden Markov Models (HMMs) and Bayesian networks.

Step 2: Contrast with related processes.

- **Filtering** estimates the current state given evidence up to now. - **Prediction** estimates future states given evidence. - **Smoothing** specifically estimates past states.

Step 3: Evaluate the options.

- (A) Smoothing: Correct, as it matches the definition. - (B) Normalization: Incorrect, normalization just rescales probabilities so they sum to 1. - (C) Clustering: Incorrect, clustering groups data into clusters, not related to temporal state estimation. - (D) Alpha Normalization: Incorrect, this is a scaling step in algorithms, not the estimation of past states.

Step 4: Conclusion.

The process described is **(A) Smoothing**.

> Quick Tip
>
> Remember: Prediction = future states, Filtering = current state, Smoothing = past states.

Q71. When the output is one of a finite set of values (such as sunny/cloudy/rainy or true/false), the learning problem is known as:

(A) Classification

(B) Clustering

(C) Regression

(D) Optimization

Correct Answer: (A) Classification

Solution:

Step 1: Recall the definition of classification.

In machine learning, classification is used when the output variable is categorical — that is, it takes values from a predefined, finite set such as {sunny, cloudy, rainy} or {true, false}.

Step 2: Compare with other tasks.

- (B) Clustering: Unsupervised learning method, where no labels are provided. - (C) Regression: Predicts continuous numerical values (e.g., predicting temperature, not “sunny”/“rainy”). - (D) Optimization: Refers to improving a function, not to predicting discrete categories.

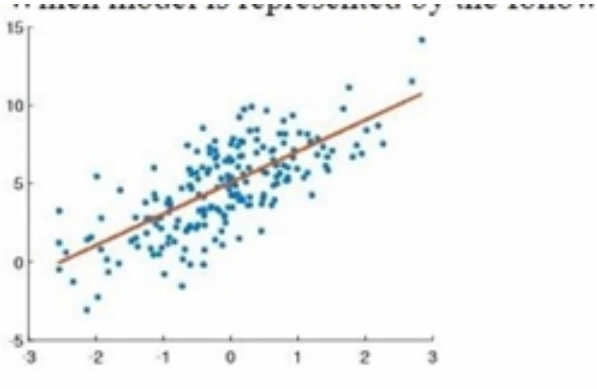
Step 3: Conclusion.

Thus, the correct answer is **(A) Classification**, since the task involves predicting discrete categories.

Quick Tip

If the output is a category or class label, the problem is classification. If the output is a continuous value, it is regression.

Q72. Which model is represented by the following graph?



(A) Logistic regression model

(B) Simple Linear regression model

(C) Multiple linear regression model

(D) k nearest neighbor model

Correct Answer: (B) Simple Linear regression model

Solution:

Step 1: Analyze the graph.

The figure shows a straight line that best fits the data points, representing a relationship between a single independent variable and a dependent variable. This matches the structure of a simple linear regression model.

Step 2: Evaluate other models.

- (A) Logistic regression: Produces an S-shaped (sigmoid) curve for classification tasks, not a straight line. - (C) Multiple linear regression: Involves more than one independent variable, requiring a plane or hyperplane, not just a straight line. - (D) k-NN: A non-parametric algorithm that classifies based on the nearest neighbors; it does not produce a regression line.

Step 3: Conclusion.

Therefore, the model shown is the **(B) Simple Linear regression model**.

> Quick Tip
>
> A straight-line relationship between one predictor and one outcome variable always indicates simple linear regression.

Q73. The term "Residual" is defined as:

(A) Fraction of all the test data's variance that is accounted for by the model.
(B) The difference between the value predicted for a data point and the actual observed value.
(C) A regression method where we tune our model parameters so as to minimize sum of the distances between data points.
(D) Actual predicted value.

Correct Answer: (B) The difference between the value predicted for a data point and the actual observed value.

Solution:

Step 1: Define residuals.

In regression analysis, a residual is the error term:

$$\text{Residual} = \text{Observed Value} - \text{Predicted Value}$$

It tells us how far the model's prediction is from the actual observed data point.

Step 2: Compare with other statistical terms.

- (A) Refers to R^2, the coefficient of determination, not residuals. - (C) Refers to the least squares method of fitting a model, which uses residuals but is not the definition of residual itself. - (D) Refers to the predicted value, which is different from the residual.

Step 3: Conclusion.

Thus, the residual is defined as the **difference between the predicted and actual observed values**, making option (B) correct.

Quick Tip

Residuals are diagnostic tools: smaller residuals indicate a better model fit, while large residuals highlight where the model struggles.

Q74. Which among the following is not a valid distance specifying criterion between the clusters, in the context of hierarchical clustering?

(A) Single linkage

(B) Group Average

(C) Complete Linkage

(D) Double linkage

Correct Answer: (D) Double linkage

Solution:

Step 1: Recall distance criteria in hierarchical clustering.

Hierarchical clustering groups data step by step by merging or splitting clusters based on certain distance measures. Common distance measures include: - **Single linkage:** Distance between the closest members of two clusters. - **Complete linkage:** Distance between the farthest members of two clusters. - **Group average:** The average distance between all pairs of members across two clusters.

Step 2: Identify the invalid option.

”Double linkage” is not a recognized or valid criterion in hierarchical clustering. It does not exist as part of standard linkage methods.

Step 3: Elimination of options.

- (A) Single linkage $\rightarrow$ valid. - (B) Group average $\rightarrow$ valid. - (C) Complete linkage $\rightarrow$ valid. - (D) Double linkage $\rightarrow$ invalid.

Step 4: Conclusion.

Thus, the correct answer is **(D) Double linkage**.

Quick Tip

In hierarchical clustering, valid linkage criteria include single, complete, and average linkage. Any term outside these, like ”double linkage,” is not standard.

Q75. In k-means algorithm, if there are n data points, then what is the minimum value of k and the maximum value of k?

(A) Minimum value of k=1, maximum value of k=n/2

(B) Minimum value of k=1, maximum value of k=n

(C) Minimum value of k=n/2, maximum value of k=n

(D) Minimum value of k=2, maximum value of k=n

Correct Answer: (B) Minimum value of k=1, maximum value of k=n

Solution:

Step 1: Recall what k means in k-means.

The value of k represents the number of clusters into which the dataset is divided.

Step 2: Minimum value of k.

- When $k = 1$, all data points belong to a single cluster. This is the smallest possible value of k.

Step 3: Maximum value of k.

- When $k = n$, each of the n data points forms its own individual cluster. This is the largest possible value of k.

Step 4: Elimination of incorrect options.

- (A) Incorrect, because maximum k is not limited to $n/2$. - (C) Incorrect, because the minimum value is not $n/2$. - (D) Incorrect, because the minimum value is not 2; it can be as low as 1.

Step 5: Conclusion.

Therefore, the correct answer is **(B) Minimum value of k = 1, maximum value of k = n**.

Quick Tip

In k-means, $k = 1$ gives one big cluster, while $k = n$ means each data point is its own cluster. These represent the extreme cases.
