CUET PG 2026 Data Science/ AI

Question Paper with Solutions

Conducted by National Testing Agency

General Instructions

1. **Duration:** The total duration of the examination is 90 minutes.
2. **Total Marks:** The complete paper carries a maximum of 300 marks.
3. **Structure:** The paper consists of 1 Section:
    - **Section A:** 75 Questions
4. **Compulsory Questions:** All 75 questions are compulsory.
5. Each question has four options. Only **one** option is correct.
6. **Right Answer:** Marks as per question scheme (Total 300 marks).
7. **Incorrect Answer:** No negative marking.
8. **Unanswered/Marked for Review:** 0 marks.

1. Which of the following string is not generated by the grammar : $S \rightarrow SaSbS \mid \epsilon$?

(A) $aabb$

(B) $abab$

(C) $aababb$

(D) $aaabb$

Correct Answer: (D) aaabb

Solution:

Concept:

- The grammar $S \rightarrow SaSbS \mid \epsilon$ generates a language where terminals 'a' and 'b' are

balanced.

- Every application of the recursive rule $SaSbS$ introduces exactly one 'a' and exactly one 'b'.
- This specific structure is the generator for the Dyck language, representing balanced strings of parentheses where 'a' is the opening and 'b' is the closing parenthesis.
- A necessary (but not sufficient) condition for any string w in this language is that the number of 'a's must equal the number of 'b's: $n_a(w) = n_b(w)$.

Step 1: Analyze the fundamental counting property of the grammar
Each production rule $S \to SaSbS$ adds exactly one 'a' and one 'b'.
The base case $S \to \epsilon$ adds zero terminals.
Therefore, any string generated by this grammar must satisfy the property $n_a(string) = n_b(string)$.
Step 2: Evaluate each option based on the counting property

- Option (A) $aabb$: Contains 2 'a's and 2 'b's. $2 = 2$. Matches property.
- Option (B) $abab$: Contains 2 'a's and 2 'b's. $2 = 2$. Matches property.
- Option (C) $aababb$: Contains 3 'a's and 3 'b's. $3 = 3$. Matches property.
- Option (D) $aaabb$: Contains 3 'a's and 2 'b's. $3 \neq 2$. Does not match.

Step 3: Verify derivations for the valid strings to ensure the grammar structure is met
For $aabb$: $S \Rightarrow SaSbS \Rightarrow aSbS \Rightarrow a(SaSbS)bS \Rightarrow aabS \Rightarrow aabb$.
For $abab$: $S \Rightarrow SaSbS \Rightarrow aSbS \Rightarrow abS \Rightarrow ab(SaSbS) \Rightarrow abab$.
Since $aaabb$ fails the basic count equality, it cannot be generated regardless of the sequence of rules used.

Quick Tip: Always check for terminal count parity in CFG questions first. If a grammar adds terminals in pairs, any string with an odd total count or mismatched pair count is immediately disqualified.

---

2. R_1 and R_2 are regular sets. Which of the following is not correct ?

(A) $R_1 \cap R_2$ needs not be regular

(B) $\Sigma^* - R_1$ is regular

(C) $R_1 \cup R_2$ is regular

(D) R_1^* is regular

Correct Answer: (A) $R_1 \cap R_2 needsnot beregular$

Solution:

Concept:

- Regular sets (or Regular Languages) are the simplest class of languages in the Chomsky hierarchy.
- They are defined by the fact that they are accepted by Finite Automata.
- One of their defining characteristics is closure under various algebraic operations.

Step 1: Examine Closure under Union and Kleene Star

By definition, the set of regular languages over an alphabet is closed under the operations used to build regular expressions.

This includes Union ($R_1 \cup R_2$) and Kleene Star (R_1^*).

Therefore, options (C) and (D) represent correct statements about regular sets.

Step 2: Examine Closure under Complementation

Regular languages are closed under complementation.

If a language L is regular, there exists a Deterministic Finite Automaton (DFA) that accepts it.

By swapping the final and non-final states of that DFA, we obtain a DFA that accepts $\Sigma^* - L$.

Therefore, option (B) represents a correct statement.

Step 3: Examine Closure under Intersection

Regular languages are closed under intersection.

This can be proven via the "Product Automaton" construction or via De Morgan's Law:

$R_1 \cap R_2 = \overline{\overline{R_1} \cup \overline{R_2}}$.

Since they are closed under union and complement, they must be closed under intersection.

The statement in (A) says it "needs not be regular", which is false because it is *always* regular.

Quick Tip: Regular languages are closed under almost all common operations: Union, Intersection, Complement, Difference, Concatenation, Reversal, and Kleene Star. Use this "completeness" to quickly spot false claims of non-regularity.

3. Which of the following search uses the problem specific knowledge beyond the definition of the problem ?

(A) Informed search

(B) Depth first search

(C) Breadth first search

(D) Uninformed search

Correct Answer: (A) Informed search

Solution:

Concept:

- Search algorithms in AI are categorized by the amount of information they possess about the goal.
- Uninformed search (blind search) only has the problem definition (start, goal test, successor function).
- Informed search (heuristic search) uses additional guidance called a heuristic function $h(n)$.

Step 1: Define Uninformed Search

Algorithms like BFS and DFS (Options B and C) explore the state space tree without knowing if one non-goal node is "more promising" than another.

They follow a fixed strategy (expanding the shallowest or deepest node).

They only use the basic definition of the problem.

Step 2: Define Informed Search

Informed search (Option A) uses domain-specific knowledge to estimate the distance to the goal.

This is represented by a heuristic function. For example, in pathfinding, the "Straight Line Distance" is specific knowledge that helps the search prioritize nodes in the right direction.

This knowledge is "beyond the definition" because it requires understanding the semantics of the states (e.g., coordinates in space).

Step 3: Identify the correct match

Since the question asks for the search that uses "problem specific knowledge", Informed search is the only appropriate term.

Quick Tip: Heuristics = Informed Search. Common examples include A* search and Greedy Best-First Search. These are generally much more efficient than blind searches like BFS or DFS.

---

4. Consider the acronym PEAS in AI. Which of the following is not correct ?

(A) P - Perceptron

(B) E - Environment

(C) A - Actuators

(D) S - Sensors

Correct Answer: (A) P - Perceptron

Solution:

Concept:

- In AI, the PEAS framework is used to group the properties of a Task Environment for an intelligent agent.
- It is a standard way to specify exactly what an agent is supposed to do and where it is supposed to do it.

Step 1: Deconstruct the PEAS acronym

- **P: Performance Measure** - The criteria that determine how successful an agent is (e.g., safety, speed, profit).
- **E: Environment** - The external world the agent operates in (e.g., city streets, a chessboard, the internet).
- **A: Actuators** - The "tools" the agent uses to act upon the environment (e.g., wheels, robotic arms, screen displays).
- **S: Sensors** - The "organs" the agent uses to perceive the environment (e.g., cameras, microphones, keyboard input).

Step 2: Compare the acronym components to the given options

- Option (B), (C), and (D) correctly identify Environment, Actuators, and Sensors.

- Option (A) claims 'P' stands for Perceptron. This is incorrect.

Step 3: Explain the term Perceptron

A Perceptron is an algorithm for supervised learning of binary classifiers. It is a type of artificial neuron. While it is a fundamental concept in AI and Machine Learning, it is not a part of the PEAS task environment specification.

Quick Tip: PEAS describes the "What, Where, How, and How Well" of an agent. If a term refers to a specific model or algorithm (like Perceptron or CNN), it doesn't belong in the PEAS description.

---

5. In UNIX, which system call is used for inter process communication via shared memory ?

(A) pipe ()

(B) msgget ()

(C) shmget ()

(D) semctl ()

Correct Answer: (C) shmget ()

Solution:

Concept:

- Inter-Process Communication (IPC) refers to the mechanisms provided by an operating system to allow processes to manage shared data.
- System V IPC in UNIX provides three main mechanisms: Message Queues, Semaphores, and Shared Memory.

Step 1: Identify the function of each provided system call

- **pipe()**: Creates a unidirectional data channel that can be used for interprocess communication. It does not involve "shared memory" in the architectural sense.
- **msgget()**: Used to get a message queue identifier. This belongs to the Message Queue IPC mechanism.
- **semctl()**: Provides semaphore control operations (like initialization or deletion). This belongs to the Semaphore IPC mechanism.

- **shmget()**: Stands for "shared memory get". It is used to allocate a System V shared memory segment.

Step 2: Explain the Shared Memory workflow in UNIX

To use shared memory, a process typically follows these steps:

1. **shmget()**: Create the segment and get a unique ID.
2. **shmat()**: Attach the memory segment to the process's address space.
3. **shmdt()**: Detach the segment when finished.
4. **shmctl()**: Delete or modify the segment.

Step 3: Conclusion

Among the options, `shmget()` is the primary system call associated with establishing shared memory IPC.

Quick Tip: UNIX system calls are named logically: - 'msg' prefix for Message Queues. - 'sem' prefix for Semaphores. - 'shm' prefix for Shared Memory. The 'get' suffix usually indicates the creation or retrieval of the resource ID.

---

6. The messages exchanged by communicating processes reside in a temporary queue. Such queue cannot be implemented in the following way :

(A) $Zerocapacity$
(B) $Blockingcapacity$
(C) $Boundedcapacity$
(D) $Unboundedcapacity$

Correct Answer: (B) Blocking capacity

Solution:

Concept:

- Message passing systems utilize a temporary queue to hold messages during inter-process communication (IPC).
- These queues are primarily characterized by their "Capacity", which defines the maximum

number of messages they can hold.

- Communication synchronization (blocking vs. non-blocking) is a behavioral property, not a structural implementation of queue capacity.

Step 1: Analyze standard implementations of message queues

- **Zero capacity:** The queue has a maximum length of zero. The sender must block until the recipient receives the message. This is known as "rendezvous".
- **Bounded capacity:** The queue has a finite length n. If the queue is not full, the sender can continue; if it is full, the sender must block.
- **Unbounded capacity:** The queue has potentially infinite length. The sender never has to block for space.

Step 2: Evaluate the term "Blocking capacity"

"Blocking" refers to the synchronization policy of the sender/receiver (e.g., synchronous vs. asynchronous).

While queues can cause a process to block, there is no such physical implementation called "Blocking capacity".

It is a mix-up of terminology between the "capacity" of the buffer and the "synchronization" of the primitive.

Quick Tip: Remember the three capacities: Zero (no buffer), Bounded (finite buffer), and Unbounded (infinite buffer). "Blocking" and "Non-blocking" describe how the process behaves when interacting with these queues.

---

7. Which is true about the fork ( ) system call ?

(A) It allocates memory

(B) It creates a child process

(C) It creates a program

(D) It creates machine code

Correct Answer: (B) It creates a child process

Solution:

Concept:

- In UNIX-like operating systems, `fork()` is the primary system call used for process creation.
- It creates a new process by duplicating the existing process (the parent).
- Both the parent and the child continue execution from the instruction immediately following the `fork()` call.

Step 1: Understand the mechanism of fork()

When `fork()` is called, the kernel creates an exact copy of the parent process's address space.

This includes the code segment, data segment, heap, and stack.

The child gets a unique Process ID (PID).

Step 2: Identify the primary purpose

The function of `fork()` is strictly to spawn a new execution context (a child process).

It does not "create a program" (that is what `exec()` does).

It does not "create machine code" (that is what a compiler/assembler does).

While it involves memory management, "creating a child process" is its fundamental definition.

Quick Tip: To remember: Fork duplicates the process. Exec replaces the process image with a new program. Fork returns 0 to the child and the child's PID to the parent.

---

8. In hardwired control, the control signals are generated :

(A) by microprogram stored in memory

(B) by fixed combinational circuits

(C) using PLA only

(D) Sequentially from ROM

Correct Answer: (B) by fixed combinational circuits

Solution:

Concept:

- A Control Unit (CU) manages the flow of data through the CPU. There are two main

types: Hardwired and Microprogrammed.

- Hardwired control is built using physical hardware components like logic gates.
- Microprogrammed control uses software-like code (microinstructions) stored in a control memory.

Step 1: Analyze Hardwired Control implementation
Hardwired control units are designed as digital logic circuits.
They use decoders, counters, and combinational logic gates (AND, OR, NOT) to generate control signals based on the current instruction opcode and clock state.
The logic is "fixed" in the hardware and is very fast but difficult to modify.
Step 2: Compare with other options

- Options (A), (C), and (D) are characteristic of **Microprogrammed Control**.
- Microprograms are stored in a control ROM or memory.
- PLAs (Programmable Logic Arrays) can be used to store microcode in a condensed format.

Quick Tip: Hardwired = Fast, fixed, combinational logic (RISC). Microprogrammed = Slower, flexible, memory-based (CISC).

---

9. Which of the following statement is not correct about addressing modes ?

(A) Immediate addressing reduces memory access time
(B) Index addressing is used for accessing array elements
(C) Register indirect addressing allows accessing memory via a pointer
(D) Relative addressing cannot be used for branch instruction

Correct Answer: (D) Relative addressing cannot be used for branch instruction

Solution:
Concept:

- Addressing modes define how the effective address of an operand is calculated.

- Common modes include Immediate, Register, Direct, Indirect, Indexed, and Relative.

Step 1: Evaluate Statement (A)

In immediate addressing, the operand is part of the instruction itself.

The CPU does not need to fetch the operand from memory separately.

Thus, it reduces memory access time. This statement is correct.

Step 2: Evaluate Statement (B)

Index addressing calculates the address as $Base + Index$.

This is the standard mathematical way to access array elements at an offset. This statement is correct.

Step 3: Evaluate Statement (C)

Register indirect addressing uses a register that contains the address of the operand (a pointer).

This is the core of pointer implementation in high-level languages. This statement is correct.

Step 4: Evaluate Statement (D)

Relative addressing calculates the target address relative to the Program Counter (PC).

It is **specifically designed** for branch and jump instructions to allow for relocatable code.

The statement "cannot be used" is factually incorrect.

Quick Tip: Relative addressing is the backbone of loops and if-statements in assembly. It makes the code "position independent," meaning it can be loaded anywhere in memory and still work.

---

10. Which type of firewall filters is based on application - layer data such as URL and HTTP headers ?

(A) Packet-filtering firewall

(B) Stateful inspection firewall

(C) Application proxy firewall

(D) Circuit level gateway

Correct Answer: (C) Application proxy firewall

Solution:

Concept:

- Firewalls filter network traffic based on rules. They operate at different layers of the OSI

model.

- The depth of inspection increases as we move up the layers.

Step 1: Differentiate firewall types by OSI layers

- **Packet-filtering (L3/L4):** Inspects IP addresses and Port numbers only.
- **Circuit-level gateway (L5):** Inspects the TCP handshake to ensure valid sessions.
- **Stateful inspection (L3/L4+):** Tracks the state of active connections.
- **Application proxy (L7):** Acts as an intermediary and can "read" the content of the traffic.

Step 2: Analyze the requirement

The question mentions URLs and HTTP headers. These are Layer 7 (Application Layer) entities. Only an Application Proxy Firewall (also known as an Application Layer Gateway) can perform deep packet inspection to view and filter this data.

Quick Tip: L3 = IP, L4 = Ports, L7 = Content (URLs, Email, Data). Only Application firewalls can stop specific commands (like a 'DELETE' request in HTTP) because they understand the protocol.

---

11. Which of the following is not a valid address in an internet employing the TCP/IP protocols ?

(A) Frame address
(B) Physical address
(C) Logical address
(D) Port address

Correct Answer: (A) Frame address

Solution:

Concept:

- The TCP/IP protocol suite uses distinct types of addresses at different layers to facilitate communication.

- These addresses correspond to the physical hardware, the network routing layer, and the application service layer.
- Standard addresses include Physical addresses, Logical addresses, Port addresses, and Specific addresses (like email or URL).

Step 1: Identify valid TCP/IP address types

- **Physical Address:** Also known as MAC address or Link-layer address. It operates at the Data Link layer and identifies devices on the same local network.
- **Logical Address:** Also known as an IP address. It operates at the Network layer and uniquely identifies a host across a global network.
- **Port Address:** Operates at the Transport layer (TCP/UDP) and identifies a specific process or application running on a host.

Step 2: Analyze "Frame address"

A "Frame" is the Protocol Data Unit (PDU) at the Data Link layer.

While a frame *contains* an address (which is the Physical/MAC address), there is no standard TCP/IP address terminology known as a "Frame address".

Therefore, it is not a valid classification of an address type in the TCP/IP suite.

Quick Tip: Associate address types with TCP/IP layers: Data Link Layer = Physical/MAC Address (48 bits). Network Layer = Logical/IP Address (32 or 128 bits). Transport Layer = Port Address (16 bits).

---

12. Suppose we can download 10 pages per minute. A page has 24 lines on average and a line has 80 characters on average. Assuming 8 bits per character, the bit rate is :

(A) 6.4×10^5 bps

(B) 1.536×10^5 bps

(C) 8.64×10^3 bps

(D) 2.56×10^3 bps

Correct Answer: (D) $2.56 \times 10^3 bps$

Solution:

Concept:

- Bit rate is defined as the number of bits transmitted or received per unit of time (typically per second, denoted as bps).
- To calculate the bit rate, we must determine the total data volume in bits and divide it by the total time in seconds.

Step 1: Calculate the number of bits in one page

Each line has 80 characters, and each character is 8 bits:

Bits per line $= 80 \times 8 = 640$ bits

Each page has 24 lines:

Bits per page $= 24 \times 640 = 15360$ bits

Step 2: Calculate the total bits downloaded per minute

We download 10 pages in one minute:

Total bits per minute $= 10 \times 15360 = 153600$ bits/minute

Step 3: Convert the rate to bits per second (bps)

Since there are 60 seconds in a minute:

Bit rate $= \frac{153600 \text{ bits}}{60 \text{ seconds}} = 2560$ bps

Step 4: Express in scientific notation to match options

$2560 = 2.56 \times 10^3$ bps.

Quick Tip: Always check the time unit in networking problems. Converting "per minute" to "per second" requires dividing by 60 at the final step to get standard bps.

---

13. Which of the following statement correctly differentiates substitution and transposition ciphers ?

(A) Substitution changes the order of symbols; transposition replaces symbol

(B) Both change symbol order and values simultaneously

(C) Substitution replaces symbol with others; transposition rearranges their position

(D) Substitution is always weaker than transposition in modern cryptography

Correct Answer: (C) Substitution replaces symbol with others; transposition rearranges their position

Solution:

Concept:

- Classical encryption algorithms are built on two fundamental building blocks: Substitution and Transposition (also called Permutation).
- Claude Shannon identified these as the mechanisms to achieve "Confusion" and "Diffusion" respectively.

Step 1: Define Substitution Ciphers

In a substitution cipher, the identity of the characters is changed, but their relative positions remain the same.

For example, the Caesar cipher replaces 'A' with 'D'.

This replaces a symbol with another symbol.

Step 2: Define Transposition Ciphers

In a transposition cipher, the identity of the characters remains unchanged, but their order (positions) is shuffled.

For example, the word "CAT" might become "TAC".

This rearranges their position without altering the symbols themselves.

Step 3: Select the accurate differentiation

Option (C) correctly describes substitution as replacing symbols and transposition as rearranging positions.

Option (A) has the definitions swapped.

Quick Tip: Substitution = Confusion (changes WHAT the character is). Transposition = Diffusion (changes WHERE the character is).

---

14. In a secure communication model, if an attacker intercepts a message and replaces it with a fake one before forwarding it, this attack is best classified as :

(A) Replay attack

(B) Masquerade attack

(C) Man in the middle

(D) Denial of service

Correct Answer: (C) Man in the middle

Solution:

Concept:

- Active security attacks involve the modification of a data stream or the creation of a false stream.
- A Man-in-the-Middle (MITM) attack occurs when an attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.

Step 1: Analyze the described attack scenario

The attacker:

1. Intercepts a legitimate message in transit.
2. Replaces the original message with a fake/modified one.
3. Forwards the fake message to the intended recipient.

This requires the attacker to be positioned between the sender and the receiver.

Step 2: Compare with other attack types

- **Replay attack:** Capturing a valid message and re-transmitting it later to produce an unauthorized effect (no replacement needed).
- **Masquerade attack:** Pretending to be an authorized entity to gain access (usually at the start of communication).
- **Denial of Service (DoS):** Preventing legitimate users from accessing a service (blocking traffic, not selectively replacing it).

Step 3: Conclusion

Intercepting and replacing messages mid-flight is the textbook definition of an active Man-in-the-Middle (MITM) attack.

Quick Tip: MITM attacks compromise both Confidentiality (eavesdropping) and Integrity (modifying/replacing messages).

15. Which of the following is a universal gate ?

(A) AND

(B) OR

(C) NAND

(D) Inverter

Correct Answer: (C) NAND

Solution:

Concept:

- A logic gate is called "Universal" if any Boolean function can be implemented using only that type of gate.
- To be universal, a gate must be able to realize the basic boolean operations: AND, OR, and NOT.
- There are only two standard universal gates in digital logic: NAND and NOR.

Step 1: Evaluate basic gates (AND, OR, NOT/Inverter)

- An AND gate alone cannot perform inversion (NOT).
- An OR gate alone cannot perform inversion (NOT).
- An Inverter alone can only negate; it cannot combine two different inputs.
- Thus, Options (A), (B), and (D) are not universal.

Step 2: Evaluate the NAND gate

We can implement all basic gates using only NAND gates:

- **NOT** X**:** Connect both inputs of a NAND gate together: $(X \cdot X)' = X'$.
- A **AND** B**:** Follow a NAND gate with a NOT (made from NAND): $((A \cdot B)')' = A \cdot B$.
- A **OR** B**:** Invert inputs before a NAND gate: $(A' \cdot B')' = A + B$ (De Morgan's Law).

Therefore, NAND is a universal gate.

Quick Tip: NAND and NOR are universal because they inherently combine a logic operation with inversion. This inversion is what allows them to generate all other Boolean functions.

---

16. Which of the following boolean expression is related to De-Morgan theorem ?

(A) $x + xy = x$

(B) $x + (y + z) = (x + y) + z$

(C) $x(y + z) = xy + xz$

(D) $(x + y)' = x'y'$

Correct Answer: (D) (x + y)' = x'y'

Solution:

Concept:

- De-Morgan's Theorems are fundamental laws in Boolean algebra used to simplify expressions by relating the complement of a sum/product to the product/sum of the complements.
- The first law states: The complement of the sum of two variables is equal to the product of the complements of the variables: $(A + B)' = A' \cdot B'$.
- The second law states: The complement of the product of two variables is equal to the sum of the complements of the variables: $(A \cdot B)' = A' + B'$.

Step 1: Identify the nature of the other options

- Option (A) $x + xy = x$: This is the **Absorption Law**.
- Option (B) $x + (y + z) = (x + y) + z$: This is the **Associative Law**.
- Option (C) $x(y + z) = xy + xz$: This is the **Distributive Law**.

Step 2: Match the target theorem

Option (D) $(x + y)' = x'y'$ is exactly the mathematical formulation of the first De-Morgan's theorem. It shows that an OR gate followed by an Inverter is logically equivalent to an AND gate with inverted inputs (Bubbled AND).

Quick Tip: To easily remember De-Morgan's: "Break the bar, change the sign." If you break a complement bar over a '+' sign, it becomes a '·' (AND), and vice versa.

---

17. After execution of ANI 0FH with accumulator = A9H, which flags are affected ?

(A) Sign, zero and carry

(B) Carry and zero

(C) Zero and parity

(D) Sign, zero, parity and carry

Correct Answer: (D) Sign, zero, parity and carry

Solution:

Concept:

- In 8085/8086 microprocessors, logical instructions like AND (ANA/ANI) affect the status flags in the flag register based on the result.
- For the 8085, logical operations (AND, OR, XOR) affect the Sign (S), Zero (Z), and Parity (P) flags according to the result, while the Carry (CY) flag is always cleared (0).

Step 1: Perform the binary logic operation

Accumulator $A = A9H = 1010\ 1001_2$

Immediate Data $0FH = 0000\ 1111_2$

Operation: 1010 1001 AND 0000 1111 = $0000\ 1001_2$ (which is $09H$).

Step 2: Determine flag status based on the result $0000\ 1001_2$

- **Sign (S):** The Most Significant Bit (MSB) is 0, so S = 0.
- **Zero (Z):** The result is $09H$, not zero, so Z = 0.
- **Parity (P):** The result has 2 ones (even), so P = 1.
- **Carry (CY):** Logical AND instructions always reset the carry flag, so CY = 0.

Step 3: Conclusion

Since S, Z, P, and CY are all either updated based on the result or forced to a known state (cleared) by the instruction, all four are considered "affected".

Quick Tip: In 8085, for all logical instructions: CY is ALWAYS cleared (reset to 0). AC is ALWAYS set (set to 1 for ANA/ANI). S, Z, P are updated based on the actual result.

---

18. After ADI 01H when A = FFH, the accumulator and carry flags are :

(A) A = 00H, CY = 0

(B) A = 00H, CY = 1

(C) A = 01H, CY = 1

(D) A = FEH, CY = 1

Correct Answer: (B) A = 00H, CY = 1

Solution:

Concept:

- `ADI` (Add Immediate) is an 8-bit addition instruction.
- If the result of the addition exceeds 255 (FFH), the 8-bit accumulator wraps around to zero, and the carry flag is set to 1.

Step 1: Perform hexadecimal addition

Current Accumulator $A = FFH = 255_{10}$

Add immediate value $01H = 1_{10}$

Sum $= FFH + 01H = 100H = 256_{10}$

Step 2: Analyze the 8-bit result

In binary:

$FFH = 1111\ 1111_2$

$01H = 0000\ 0001_2$

Sum $= 1\ 0000\ 0000_2$

Step 3: Determine register values

- The 8-bit accumulator stores the lower 8 bits: $0000\ 0000_2 = 00H$.
- The 9th bit is the carry out, which sets the Carry Flag ($CY = 1$).

Quick Tip: Adding 1 to the maximum 8-bit value (FFH) always results in an overflow to 00H and a Carry of 1. This is the equivalent of an odometer rolling over from 999 to 000.

---

19. In the fractional knapsack problem, the greedy choice is based on (i^{th} item has worth/value as V_i and weight as W_i) :

(A) Maximum value (V_i)

(B) Minimum weight (W_i)

(C) Maximum value/weight (V_i/W_i) ratio

(D) Random choice

Correct Answer: (C) Maximum value/weight (V_i/W_i)$ratio$

Solution:

Concept:

- The fractional knapsack problem aims to maximize the total value in a knapsack with limited capacity, where we can take fractions of items.
- Unlike the 0/1 knapsack problem (which requires Dynamic Programming), the fractional version can be solved optimally using a Greedy approach.

Step 1: Identify the optimal greedy strategy

To maximize profit per unit of space occupied, we should prioritize items that give the most value for every kilogram of weight they add.

This metric is the "value density" or the value-to-weight ratio: V_i/W_i.

Step 2: Outline the algorithm

- Calculate the ratio V_i/W_i for every item.
- Sort the items in descending order of this ratio.
- Add the highest ratio items into the knapsack first.
- If an item cannot fit completely, take a fraction of it to fill the remaining capacity.

Step 3: Evaluation

Picking based on pure value or pure weight alone is not guaranteed to be optimal. The ratio captures the relative worth of the item compared to the space it consumes.

Quick Tip: Fractional Knapsack = Greedy (Ratio). 0/1 Knapsack = Dynamic Programming (cannot be solved by Greedy). The greedy choice works for fractional because we are never forced to "waste" space.

---

20. Which of the following statement is true ?

(A) Greedy and dynamic programming both require optimal sub structure

(B) Greedy requires overlapping sub problems

(C) Dynamic programming does not require optimal sub structure

(D) Greedy always gives optimal solution

Correct Answer: (A) Greedy and dynamic programming both require optimal sub structure

Solution:

Concept:

- Both Greedy algorithms and Dynamic Programming (DP) are strategies used for solving optimization problems.
- They share certain theoretical requirements but differ in how they build solutions.

Step 1: Identify shared requirements

An optimization problem must exhibit **Optimal Substructure** to be solved by either method. This means an optimal solution to the problem contains within it optimal solutions to its subproblems.

Therefore, Statement (A) is correct and Statement (C) is false.

Step 2: Identify differences

- **Overlapping Subproblems:** This is a requirement for DP to be efficient (so results can be memoized). Greedy does not require this; it makes a local choice and never re-visits subproblems. Thus, (B) is false.
- **Greedy Choice Property:** Greedy works only if a local optimal choice leads to a global optimal. Many problems (like 0/1 knapsack) do not satisfy this, so Greedy does NOT always give an optimal solution. Thus, (D) is false.

Quick Tip: Greedy = Optimal Substructure + Greedy Choice Property. DP = Optimal Substructure + Overlapping Subproblems. Common Ground: Optimal Substructure.

---

21. For the Dijkstra's algorithm applying on a graph, which of the following condition is required ?

(A) Graph must be directed acyclic

(B) Graph must not have negative edge weights

(C) All edges must have equal weight

(D) Graph must be complete

Correct Answer: (B) Graph must not have negative edge weights

Solution:

Concept:

- Dijkstra's algorithm is a greedy algorithm used to find the shortest path from a single source to all other vertices in a weighted graph.
- It functions by maintaining a set of "visited" nodes and repeatedly picking the unvisited node with the smallest tentative distance.
- The core assumption of Dijkstra's is that once a node is added to the "visited" set, its shortest path has been found and will not be improved by any other path.

Step 1: Evaluate the impact of negative weights

Dijkstra's algorithm assumes that adding an edge to a path can only increase (or keep the same) its total weight.

If negative edge weights are present, a path that looks longer currently could eventually become shorter by including a negative edge.

Because Dijkstra's "locks in" the shortest distance greedily, it fails to reconsider nodes and thus produces incorrect results in the presence of negative edges.

Step 2: Verify other options

- **Directed Acyclic Graph (DAG):** While Dijkstra's works on DAGs, it is not *required*. It works perfectly fine on graphs with cycles as long as weights are non-negative.

- **Equal weights:** If all weights are equal, Dijkstra's effectively becomes Breadth-First Search (BFS). This is a special case, not a requirement.
- **Complete graph:** Dijkstra's works on both sparse and dense (complete) graphs.

Quick Tip: Dijkstra = Non-negative weights only. Bellman-Ford = Can handle negative weights (but not negative cycles). If you see negative edges in a shortest-path problem, avoid Dijkstra.

---

22. Which of the following operation is not a partial order relation ?

(A) "less than or equal ($\leq$)" on $\mathbb{R}$

(B) "subset ($\subseteq$)" on power set of set A

(C) "Divides" on $\mathbb{N}$

(D) "less than ($<$)" on $\mathbb{R}$

Correct Answer: (D) "less than ($<$)" on $\mathbb{R}$

Solution:

Concept:

- A relation R on a set S is a **Partial Order Relation** if it satisfies three properties:
- **Reflexivity:** aRa for all $a \in S$.
- **Antisymmetry:** If aRb and bRa, then $a = b$.
- **Transitivity:** If aRb and bRc, then aRc.

Step 1: Check Option (A): $\leq$ on $\mathbb{R}$

$x \leq x$ (Reflexive). If $x \leq y$ and $y \leq x$, then $x = y$ (Antisymmetric). If $x \leq y$ and $y \leq z$, then $x \leq z$ (Transitive). This is a partial order.

Step 2: Check Option (B): $\subseteq$ on Power Set

$X \subseteq X$ (Reflexive). If $X \subseteq Y$ and $Y \subseteq X$, then $X = Y$ (Antisymmetric). Inclusion is transitive. This is a partial order.

Step 3: Check Option (C): "Divides" on $\mathbb{N}$

$n|n$ (Reflexive). If $a|b$ and $b|a$, then $a = b$ in natural numbers (Antisymmetric). If $a|b$ and $b|c$, then $a|c$ (Transitive). This is a partial order.

Step 4: Check Option (D): $<$ on $\mathbb{R}$

For a relation to be reflexive, $x < x$ must be true for all x. However, $5 < 5$ is **false**.

Since it fails reflexivity, it cannot be a partial order. It is actually a "Strict Partial Order".

Quick Tip: Partial Order = Reflexive + Antisymmetric + Transitive. Strict Order = Irreflexive + Asymmetric + Transitive. The difference is whether the "equal to" case is allowed.

---

23. if $f(x) = x^2 + 1$ and $g(x) = 2x - 3$, then $(f \circ g)(x)$ is equal to :

(A) $4x^2 - 12x + 9$

(B) $4x^2 - 12x + 10$

(C) $4x^2 - 12x + 13$

(D) $2x^2 - 6x + 10$

Correct Answer: (B) $4x^2 - 12x + 10$

Solution:

Concept:

- Function composition $(f \circ g)(x)$ means evaluating the function f at the output of function g.
- Mathematically: $(f \circ g)(x) = f(g(x))$.

Step 1: Substitute $g(x)$ into the expression

We are given $g(x) = 2x - 3$.

Therefore, $(f \circ g)(x) = f(2x - 3)$.

Step 2: Apply the definition of $f(x)$

The function f is defined as $f(\text{input}) = (\text{input})^2 + 1$.

Substituting our input $(2x - 3)$:

$(f \circ g)(x) = (2x - 3)^2 + 1$

Step 3: Expand the algebraic expression

Using the identity $(a - b)^2 = a^2 - 2ab + b^2$:

$(2x - 3)^2 = (2x)^2 - 2(2x)(3) + (3)^2$

$(2x - 3)^2 = 4x^2 - 12x + 9$

Step 4: Add the final constant

$(f\,o\,g)(x) = (4x^2 - 12x + 9) + 1$

$(f\,o\,g)(x) = 4x^2 - 12x + 10$

Quick Tip: Be careful with the order! $(f\,o\,g)(x)$ is $f(g(x))$, while $(g\,o\,f)(x)$ is $g(f(x))$. Always perform the inner function first, then substitute that entire result into the outer function.

---

24. Consider the group $G = \{+1, -1, +i, -i\}$ under multiplication operation. Which of the following is not correct ?

(A) G is an abelian group

(B) G is a cyclic group

(C) $\{1, -1\}$ is a subgroup of G

(D) $\{1, -i, i\}$ is a subgroup of G

Correct Answer: (D) {1, -i, i} is a subgroup of G

Solution:

Concept:

- A subset H of a group G is a subgroup if it is a group under the same operation.
- Fundamental properties of subgroups:
- **Closure:** If $a, b \in H$, then $a \cdot b \in H$.
- **Identity:** The identity of G (which is 1 here) must be in H.
- **Inverses:** If $a \in H$, then $a^{-1} \in H$.
- **Lagrange's Theorem:** The order of a subgroup must divide the order of the group.

Step 1: Analyze the group G

The group $G = \{1, -1, i, -i\}$ has 4 elements ($|G| = 4$).

It is abelian (multiplication of complex numbers is commutative).

It is cyclic because $i^1 = i, i^2 = -1, i^3 = -i, i^4 = 1$. Thus, i is a generator.

Statements (A) and (B) are correct.

Step 2: Analyze Option (C)

$H_1 = \{1, -1\}$.

Closure: $1 \cdot 1 = 1, 1 \cdot (-1) = -1, (-1) \cdot (-1) = 1$. All results are in H_1.

Identity $1 \in H_1$. Inverses exist $(1^{-1} = 1, (-1)^{-1} = -1)$.

Statement (C) is correct.

Step 3: Analyze Option (D)

$H_2 = \{1, -i, i\}$.

This subset has 3 elements. By Lagrange's Theorem, the order of a subgroup must divide the order of the group.

3 does not divide 4. Therefore, it cannot be a subgroup.

Alternatively, check closure: $i \cdot i = -1$. Since $-1 \notin \{1, -i, i\}$, the closure property fails.

Quick Tip: Lagrange's Theorem is the fastest way to debunk a subgroup. If the number of elements in the subset doesn't divide the total number of elements in the group, it's impossible for it to be a subgroup.

---

25. Which visualization method is most effective for detecting outliers in multidimensional numerical data ?

(A) Line graph

(B) Pie chart

(C) Box plot

(D) Heat map

Correct Answer: (C) Box plot

Solution:

Concept:

- Outliers are data points that differ significantly from other observations.
- Effective outlier detection requires a visual summary of the data's distribution, including the median, quartiles, and extremes.

Step 1: Analyze the purpose of a Box Plot

A box plot (also known as a box-and-whisker plot) displays the five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum.

Most importantly, it defines "whiskers" based on the Interquartile Range (IQR).

Any data point falling beyond 1.5 × IQR from the quartiles is explicitly plotted as an individual point (an outlier).

Step 2: Compare with other methods

- **Line graph:** Used for showing trends over time. Outliers might look like spikes but aren't mathematically isolated.
- **Pie chart:** Shows proportions of a whole. Useless for outlier detection in numerical distributions.
- **Heat map:** Uses colors to show the magnitude of values. While it can show patterns, it is less precise than a box plot for identifying specific numerical outliers across multiple dimensions.

Step 3: Conclusion

The Box Plot is the industry standard for visualizing univariate and multivariate outliers because of its built-in statistical thresholding.

Quick Tip: Box plots use the "Whiskers" rule to find outliers. Points outside the whiskers are outliers. Points inside are typical data. This provides an objective visual test.

---

26. Which clustering algorithm can detect clusters of arbitrary shape and handle noise effectively ?

(A) K-Means

(B) Mean shift

(C) DBSCAN

(D) Agglomerating hierarchical clustering

Correct Answer: (C) DBSCAN

Solution:

Concept:

- Clustering is an unsupervised learning task that groups similar data points together.

- Traditional algorithms like K-Means assume that clusters are spherical and of similar size, which is often not true for real-world data.
- Density-based clustering identifies regions of high point density separated by regions of low point density.

Step 1: Analyze the limitations of K-Means and Hierarchical clustering

K-Means (Option A) uses a distance-based approach to minimize the sum of squared distances to centroids. This forces clusters to be convex/spherical. It is also highly sensitive to outliers (noise), as noise points can significantly pull the centroids away from the true cluster center. Hierarchical clustering (Option D) is also primarily distance-based and lacks a built-in mechanism to ignore noise.

Step 2: Evaluate DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN defines clusters based on two parameters: ϵ (epsilon - radius) and $MinPts$ (minimum points).

- **Core Points:** Points that have at least $MinPts$ within their ϵ-neighborhood.
- **Border Points:** Points that are within the neighborhood of a core point but don't have enough neighbors themselves.
- **Noise Points:** Points that are neither core nor border points.

Because it connects adjacent high-density regions, it can follow any "trail" of data, allowing it to find clusters of arbitrary shapes (like "moons" or "donuts").

Step 3: Conclusion on Noise Handling

Unlike other algorithms that force every point into a cluster, DBSCAN explicitly labels isolated points as "Noise." This makes it exceptionally robust in datasets where outliers are prevalent.

Quick Tip: Use K-Means if clusters are circular/spherical. Use DBSCAN if the data has irregular shapes or a lot of "stray" points that shouldn't belong to any group.

---

27. Which of the following is primitive data type in C ?

(A) long int

(B) struct

(C) union

(D) enum

Correct Answer: (A) long int

Solution:

Concept:

- Data types in C are categorized into:
  - **Primitive/Primary/Basic types:** Built-in types that represent single values (e.g., int, char, float, double).
  - **Derived types:** Types derived from primitives (e.g., arrays, pointers).
  - **User-Defined/Constructed types:** Types created by the programmer (e.g., struct, union, enum).

Step 1: Analyze the fundamental types in C

Primitive types are the atomic building blocks of the language. "int" is a primitive type. "long" is a type modifier that, when applied to "int", creates "long int". In C, the set of integer types (including short, long, signed, unsigned variations) are considered primary or primitive data types because they are supported directly by the hardware and the compiler's basic logic.

Step 2: Evaluate User-defined types

- **struct (Option B):** A structure is a container that holds variables of different types. It is defined by the user.
- **union (Option C):** Similar to a struct but all members share the same memory location. Defined by the user.
- **enum (Option D):** An enumeration is a user-defined type consisting of a set of named integer constants.

Step 3: Conclusion

Since "long int" is a built-in numeric type provided by the C standard to handle larger integer values, it is the only primitive type among the choices.

Quick Tip: Primitive = Built-in (comes with the compiler). User-defined = You define it using keywords like 'struct' or 'typedef'. Modifiers like 'short', 'long', 'signed', and 'unsigned' don't change the primitive nature of the underlying 'int' or 'double'.

---

28. Which of the following is not a non linear data structure ?

(A) Binary tree

(B) Heap

(C) Graph

(D) char

Correct Answer: (D) char

Solution:

Concept:

- Data structures are classified based on how data elements are organized.
- **Linear Data Structures:** Elements are arranged in a sequence (e.g., Array, Stack, Queue, Linked List).
- **Non-Linear Data Structures:** Elements are not arranged in a sequence; they form hierarchical or interconnected relationships (e.g., Trees, Graphs).

Step 1: Identify the non-linear structures among the options

- **Binary Tree (Option A):** A hierarchical structure where each node has at most two children. Clearly non-linear.
- **Heap (Option B):** A specialized tree-based data structure (often represented in an array, but logically a complete binary tree). Non-linear.
- **Graph (Option C):** A set of vertices connected by edges in a non-sequential web. Non-linear.

Step 2: Analyze the "char" data type

A "char" (character) is a basic data type in programming. While it is not a "structure" in the same way a tree is, it represents a single scalar value. If we consider it in the context of

data organization, characters are treated linearly (like in a string/array). More importantly, it definitely does not possess the hierarchical or networked properties required for a non-linear classification.

Step 3: Final conclusion

Since the question asks for what is *not* non-linear, "char" is the only correct fit as it is a primitive/scalar type, and the others are explicitly non-linear structures.

Quick Tip: Non-linear = Parent-Child (Trees) or Neighbor-Neighbor (Graphs). Linear = One-after-another (Arrays, Lists). Primitive types like 'char' or 'int' are the values stored inside these structures.

---

29. Which of the following is not a keyword in C ?

(A) do

(B) break

(C) join

(D) for

Correct Answer: (C) join

Solution:

Concept:

- Keywords are reserved words in a programming language that have a predefined meaning to the compiler.
- They cannot be used as identifiers (variable names, function names, etc.).
- ANSI C has 32 reserved keywords.

Step 1: Evaluate common C control flow keywords

- **do (Option A):** Used in the "do-while" loop construct. It is a keyword.
- **break (Option B):** Used to exit from a loop or switch statement prematurely. It is a keyword.
- **for (Option D):** Used for implementing the "for" loop. It is a keyword.

Step 2: Evaluate the word "join"

The word "join" is common in database languages (SQL) or in multithreading libraries (like POSIX threads `pthread_join`). However, it is **not** a reserved keyword in the C programming language itself. You are free to use "join" as a variable name or function name in a C program.

Step 3: Reference full keyword list

Standard C keywords include: auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while. "join" is absent from this list.

Quick Tip: C is a small language with few keywords. If a word sounds like it belongs to SQL (JOIN) or high-level threading, it is likely a function name in a library, not a core C keyword.

---

30. Which of the following is not correct IP address ?

(A) 20.110.254.160

(B) 25.256.230.10

(C) 230.110.230.230

(D) 10.11.234.234

Correct Answer: (B) 25.256.230.10

Solution:

Concept:

- An IPv4 (Internet Protocol version 4) address is a 32-bit numeric address written as four decimal numbers separated by dots (dotted-decimal notation).
- Each of the four numbers is called an "octet" because it represents 8 bits.
- An 8-bit binary number can represent values from 00000000_2 to 11111111_2.

Step 1: Determine the valid range for an octet

The minimum value is 0 and the maximum value is $2^8 - 1 = 255$.

Therefore, any decimal number in a valid IPv4 address must be in the range $[0, 255]$.

Step 2: Examine each option against the range

- **Option (A):** 20, 110, 254, 160. All values are ≤ 255. **Valid.**
- **Option (B):** 25, **256**, 230, 10. The second octet is 256, which exceeds the maximum limit of 255. **Invalid.**
- **Option (C):** 230, 110, 230, 230. All values are ≤ 255. **Valid.**
- **Option (D):** 10, 11, 234, 234. All values are ≤ 255. **Valid.**

Step 3: Conclusion

Since an octet cannot exceed 255, the address in Option (B) is not a correct IP address.

Quick Tip: To quickly validate an IP address, look for numbers greater than 255 or addresses with more/fewer than 4 dots. 255 is the "magic number" because 256 is the first value that requires a 9th bit.

---

31. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : Preemptive scheduling can lead to race conditions in critical section.

Reason (R) : Preemption allows multiple processes to access shared data simultaneously.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)
(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)
(C) (A) is correct but (R) is not correct
(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- A race condition occurs when multiple processes access and manipulate the same data concurrently.
- The outcome of the execution depends on the particular order in which the access takes place.

- Preemptive scheduling allows the operating system to interrupt a process at any time to assign the CPU to another process.

Step 1: Evaluate the Assertion (A)

Preemptive scheduling can interrupt a process while it is in its critical section.

If a process is modifying shared data and is preempted, the data may be in an inconsistent state.

The next process scheduled might then access this inconsistent data, causing a race condition.

Thus, Assertion (A) is correct.

Step 2: Evaluate the Reason (R)

Preemption facilitates "concurrency" by allowing multiple processes to be in a state of execution at the same time.

In a broad sense, this allows processes to access shared data "simultaneously" (overlapping in time).

Without preemption, a process would finish its critical section before another could start.

Thus, Reason (R) is correct in the context of concurrent system theory.

Step 3: Determine the relationship

The reason why race conditions happen under preemption is exactly because processes can be interleaved.

This interleaving creates the situation where multiple processes attempt to access the same resource at once.

Therefore, (R) explains (A).

Quick Tip: To prevent race conditions in preemptive systems, we use synchronization primitives like mutexes or semaphores. In non-preemptive kernels, race conditions are much rarer because a process won't be interrupted mid-task.

---

32. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : In a ripple carry adder, carry propagation delay increases linearly with the number of bits.

Reason (R) : Each bit addition must wait for the carry output from the previous stage.

In the light of the above statements, choose the most appropriate answer from the options

given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)

(C) (A) is correct but (R) is not correct

(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- A Ripple Carry Adder (RCA) is a digital circuit that produces the arithmetic sum of two binary numbers.
- It is constructed by cascading several full adders in series.

Step 1: Analyze the structure of an RCA

In an n-bit RCA, the first full adder computes the sum of the LSBs.

The carry generated by the i^{th} bit addition is fed as the input carry (C_{in}) to the $(i+1)^{th}$ bit addition.

This creates a chain where each stage depends on the completion of the previous one.

Step 2: Calculate the propagation delay

If Δt is the delay of a single full adder stage to produce a carry:

Total delay $T \approx n \times \Delta t$.

Since T is directly proportional to n, the delay increases linearly.

Assertion (A) is correct.

Step 3: Verify the causal link

The reason the delay is linear is exactly because of the serial dependency described in Step 1.

Reason (R) correctly describes this "wait" condition.

Thus, (R) is the correct explanation for (A).

Quick Tip: To overcome the linear delay of RCA, designers use Carry Look-ahead Adders (CLA). CLAs generate all carries in parallel, making the delay logarithmic or constant relative to bits.

---

33. Given below are two statements : one is labelled as Assertion (A) and the other is labelled

as Reason (R).

Assertion (A) : TCP uses sequence numbers to ensure reliable, in-order delivery of data segment.

Reason (R) : Sequence number helps detect packet duplication and reordering in transmission.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)
(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)
(C) (A) is correct but (R) is not correct
(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- TCP (Transmission Control Protocol) is a connection-oriented protocol that provides reliable byte-stream delivery.
- IP (the underlying layer) is best-effort and does not guarantee order or delivery.

Step 1: Explain the role of Sequence Numbers

TCP breaks the application data into segments.
Each byte of data is assigned a unique sequence number.
The sequence number in a segment header represents the number of the first byte in that segment.

Step 2: Analyze how sequence numbers solve network issues

- **Reordering:** If packets arrive as [Seq 200, Seq 100], the receiver uses the numbers to put 100 before 200.
- **Duplication:** If two packets with Seq 100 arrive, the receiver keeps one and discards the duplicate.
- **Loss:** If Seq 100 and 300 arrive, the receiver knows Seq 200 is missing and requests retransmission.

Step 3: Conclusion

Assertion (A) is true as this is the primary mechanism of TCP reliability.

Reason (R) is true as it explains the specific sub-tasks sequence numbers perform.
Since performing these sub-tasks is exactly how reliability is achieved, (R) is the correct explanation.

Quick Tip: While UDP is faster because it lacks these checks, TCP is essential for applications like File Transfer (FTP) or Web Browsing (HTTP) where data integrity is critical.

---

34. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : TCP provides reliable, in-order delivery of data segments using sequence numbers.

Reason (R) : TCP ensures security and confidentiality of transmitted data.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)
(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)
(C) (A) is correct but (R) is not correct
(D) (A) is not correct but (R) is correct

Correct Answer: (C) (A) is correct but (R) is not correct

Solution:

Concept:

- TCP is a Transport Layer protocol designed for reliable communication.
- Security functions like encryption and authentication are typically handled by different protocols or layers.

Step 1: Verify Assertion (A)
As established in previous questions, TCP is famous for its "Reliable Byte Stream" service.
It uses sequence numbers, acknowledgments, and retransmissions to ensure data is received in order.
Thus, Assertion (A) is correct.

Step 2: Verify Reason (R)

Standard TCP sends data in plain text. It does not provide any encryption.

It does not ensure confidentiality (preventing eavesdropping) or data integrity against malicious tampering.

Security is provided by the **TLS (Transport Layer Security)** protocol, which runs on top of TCP (e.g., HTTPS = HTTP over TLS over TCP).

Thus, Reason (R) is incorrect.

Step 3: Final Choice

Since (A) is true but (R) is false, Option (C) is the correct answer.

Quick Tip: Reliability $\neq$ Security. A reliable connection ensures the message arrives perfectly. A secure connection ensures the message remains private. TCP provides the former, not the latter.

---

35. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : In a synchronous counter, all flip flops toggle simultaneously.

Reason (R) : All flip flops share the same clock signal.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)

(C) (A) is correct but (R) is not correct

(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- Counters are sequential circuits used to count pulses.
- **Asynchronous (Ripple) Counters:** The output of one flip-flop serves as the clock for the next.

- **Synchronous Counters:** All flip-flops are connected to a common global clock signal.

Step 1: Analyze the Assertion (A)

Because every flip-flop in a synchronous counter is triggered by the exact same clock edge, they all change their states at the same time.

There is no "ripple" effect or accumulated delay between stages.

Assertion (A) is correct.

Step 2: Analyze the Reason (R)

The definition of a synchronous system is one where all elements are synchronized to a single clock source.

By wiring the clock input of every flip-flop to the same external clock signal, we achieve simultaneous triggering.

Reason (R) is correct.

Step 3: Determine if (R) explains (A)

The simultaneous toggling is a direct physical result of sharing the same clock signal.

Therefore, (R) is the fundamental explanation for why (A) happens.

Quick Tip: Synchronous counters are faster because they don't suffer from cumulative propagation delays. However, they require more complex combinational logic at the inputs to determine the next state.

---

36. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : During memory interfacing, address decoding is required to select the desired memory chip.

Reason (R) : Each memory chip has unique address range that must be enabled using chip select signals.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)

(C) (A) is correct but (R) is not correct

(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- A computer system typically contains multiple memory chips (RAM, ROM) and I/O devices.
- The processor has a single address bus that is shared by all these components.
- Address decoding is the process of generating a unique 'Chip Select' (CS) or 'Device Enable' signal for each individual component so they don't clash on the bus.

Step 1: Evaluate the Assertion (A)

When a processor wants to read or write data, it puts an address on the bus. To ensure that only the intended memory chip responds, we must "decode" that address to activate the specific chip's enable pin. Without this, multiple chips might attempt to drive the bus simultaneously, causing a bus contention. Thus, (A) is true.

Step 2: Evaluate the Reason (R)

In a memory map, different physical chips are assigned different ranges (e.g., Chip 1 handles $0000H - 3FFFH$, Chip 2 handles $4000H - 7FFFH$). The address decoder looks at the higher-order bits to see which range is being accessed and then pulls the corresponding chip's CS pin low (or high). Thus, (R) is true.

Step 3: Determine the relationship

The physical requirement of having unique address ranges (R) is exactly why we need a decoder (A) to translate those logical addresses into physical selection signals. Hence, (R) explains (A).

Quick Tip: Address decoding is like a postman: The address is on the envelope, but the 'Decoder' is what decides which specific house door (Chip Select) to knock on based on the house number.

---

37. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : Prim's and Kruskal's algorithms always produce the same minimum total weight spanning tree for a given connected, weighted graph.

Reason (R) : Both algorithms use same greedy properties of choosing the minimum weight edge at each step.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)
(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)
(C) (A) is correct but (R) is not correct
(D) (A) is not correct but (R) is correct

Correct Answer: (D) (A) is not correct but (R) is correct

Solution:

Concept:

- A Minimum Spanning Tree (MST) is a subset of edges that connects all vertices without cycles and with the minimum possible total edge weight.
- Prim's algorithm builds the tree one vertex at a time.
- Kruskal's algorithm builds the tree by processing edges in increasing order of weight.

Step 1: Evaluate Assertion (A)

While Prim's and Kruskal's will always find **a** spanning tree with the same minimum total weight, they do not necessarily produce the **same set of edges** (the same tree). If a graph has multiple edges with the same weight, there can be multiple distinct MSTs. Prim's might choose one edge while Kruskal's chooses a different one with an identical weight. Thus, (A) is false.

Step 2: Evaluate Reason (R)

Both algorithms are indeed greedy. Prim's greedily picks the minimum weight edge connected to the current growing tree. Kruskal's greedily picks the absolute minimum weight edge from the entire remaining set (that doesn't form a cycle). This shared greedy philosophy is correct. Thus, (R) is true.

Quick Tip: An MST is unique IF AND ONLY IF all edge weights in the graph are distinct. If weights are repeated, expect multiple valid MSTs, and different algorithms may find different ones.

---

38. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : Every function is a relation.

Reason (R) : A function can be one to many.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)

(C) (A) is correct but (R) is not correct

(D) (A) is not correct but (R) is correct

Correct Answer: (C) (A) is correct but (R) is not correct

Solution:

Concept:

- A **Relation** from set A to B is any subset of the Cartesian product $A \times B$.
- A **Function** is a special relation where every element in the domain (A) is mapped to **exactly one** element in the codomain (B).

Step 1: Evaluate Assertion (A)

By definition, a function is defined as a subset of a relation that satisfies the uniqueness condition. Therefore, mathematically, the set of all functions is a subset of the set of all relations. This makes (A) true.

Step 2: Evaluate Reason (R)

One of the primary rules of a function is that one input cannot have multiple different outputs. A relation can be "one-to-many", but if it is, it **fails** to be a function. Functions must be either "one-to-one" or "many-to-one". Thus, (R) is false.

Quick Tip: Function Test: 1. Every element in Domain must have an image. 2. No element in Domain can have more than one image (No One-to-Many).

---

39. Given below are two statements : one is labelled as Assertion (A) and the other is labelled as Reason (R).

Assertion (A) : $++*ptr$ **in C increments the value pointed by ptr.**

Reason (R) : ptr is a pointer.

In the light of the above statements, choose the most appropriate answer from the options given below :

(A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

(B) Both (A) and (R) are correct but (R) is not the correct explanation of (A)

(C) (A) is correct but (R) is not correct

(D) (A) is not correct but (R) is correct

Correct Answer: (A) Both (A) and (R) are correct and (R) is the correct explanation of (A)

Solution:

Concept:

- Pointers store the memory address of another variable.
- The dereference operator ($*$) accesses the value stored at that address.
- The pre-increment operator (++) increases the value of its operand.

Step 1: Analyze the expression $++*ptr$ using operator precedence

In C, both $*$ and prefix ++ have the same precedence and associate from right to left.

1. $*ptr$ is evaluated first: It fetches the value at the address held by `ptr`.
2. ++ is then applied to that fetched value: It increments the actual data stored in that memory location.

So, if `ptr` points to an integer 5, after this operation, that integer becomes 6. Thus, (A) is true.

Step 2: Evaluate the Reason (R)

For the expression $*ptr$ to be valid and to allow dereferencing, the variable `ptr` must indeed be declared as a pointer type. If it were a regular integer, the compiler would throw an error. Thus, (R) is true.

Step 3: Determine the relationship

The fact that `ptr` is a pointer is the reason we can apply the dereference operator, which in turn allows the increment operator to reach and modify the target value. Thus, (R) is the underlying reason for (A).

Quick Tip: Precedence matters! $++*ptr$ increments the value. $*ptr++$ increments the pointer address (after the line is executed). (*ptr)++ increments the value (after the line is executed).

40. The attributes in IPsec for Authentication Header protocol in transport mode from left to right are :

A. IPsec header

B. Padding

C. Authentication header

D. Rest of original packet

Choose the correct answer from the options given below :

(A) D, C, A, B

(B) A, C, D, B

(C) D, C, B, A

(D) A, C, B, D

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

Solution:

Concept:

- IPsec (Internet Protocol Security) provides security services at the IP layer.
- Authentication Header (AH) provides data integrity and authentication.
- Transport Mode is used for end-to-end communication between two hosts.

Step 1: Analyze the AH Transport Mode packet structure

In transport mode, the IPsec header is inserted immediately after the original IP header and before the upper-layer protocol header (TCP/UDP). The sequence from the start of the packet (left) to the end (right) is:

1. **Original IP Header:** This is often referred to as the initial "IPsec header" in packet diagrams because it is the outer header of the secured packet.
2. **Authentication Header (AH):** Inserted to protect the rest of the packet.
3. **Upper Layer Protocol / Data:** The "Rest of the original packet" (Payload).
4. **Padding:** Added at the end to satisfy block size requirements or alignment.

Step 2: Map the attributes to the sequence

Matching with the given letters: A (IP header) → C (AH) → D (Payload) → B (Padding).

Step 3: Conclusion

The correct sequence is A, C, D, B.

Quick Tip: Transport Mode: [IP][AH][Data][Padding] (Protects the payload). Tunnel Mode: [New IP][AH][Original IP][Data][Padding] (Protects the entire original packet).

---

41. Consider the following automata in increasing processing power :

A. Linear bounded automata

B. Turing machine

C. Finite automata

D. Pushdown automata

Choose the correct answer from the options given below :

(A) A, B, D, C

(B) C, D, A, B

(C) B, A, C, D

(D) C, D, B, A

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

Solution:

Concept:

- The Chomsky hierarchy classifies grammars and their corresponding automata based on their generative and computational power.
- Computational power refers to the complexity of languages the machine can recognize and the flexibility of its memory/storage.
- The hierarchy from least powerful to most powerful is: Regular ⊂ Context-Free ⊂ Context-Sensitive ⊂ Recursively Enumerable.

Step 1: Identify the power of Finite Automata (C)

Finite Automata have the least processing power as they have no external memory (only states). They recognize only Regular Languages.

Step 2: Identify the power of Pushdown Automata (D)

Pushdown Automata are more powerful than FA because they have an infinite stack for memory. They recognize Context-Free Languages.

Step 3: Identify the power of Linear Bounded Automata (A)

LBAs are more powerful than PDAs as they can use the input tape itself for restricted read/write operations. They recognize Context-Sensitive Languages.

Step 4: Identify the power of Turing Machines (B)

Turing Machines have the highest processing power as they have an infinite read/write tape. They can simulate any algorithmic process and recognize Recursively Enumerable Languages.

Step 5: Arrange in increasing order

The order of increasing power is C (Finite) → D (Pushdown) → A (Linear Bounded) → B (Turing Machine).

Quick Tip: Remember the hierarchy using the acronym: FA < PDA < LBA < TM. Computational power is strictly related to the type of memory used (None < Stack < Limited Tape < Infinite Tape).

---

42. Consider the historic perspective of AI in terms of year of their development in increasing year :

A. Turing test

B. Expert systems

C. Perceptron

D. Analytical Engine

Choose the correct answer from the options given below :

(A) D, A, C, B

(B) A, C, B, D

(C) D, A, B, C

(D) A, C, D, B

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

Solution:

Concept:

- Artificial Intelligence has evolved from mechanical computing theories to complex sym-

bolic and neural network systems.

- Understanding the timeline helps identify how computational capacity paved the way for advanced AI models.

Step 1: Identify the era of the Analytical Engine (D)

Proposed by Charles Babbage in 1837, the Analytical Engine was the first design for a general-purpose mechanical computer. It is the earliest item in the list.

Step 2: Identify the era of the Turing Test (A)

Alan Turing published his seminal paper "Computing Machinery and Intelligence" in 1950, which introduced the Turing Test as a benchmark for machine intelligence.

Step 3: Identify the era of the Perceptron (C)

The Perceptron, an early artificial neural network, was developed by Frank Rosenblatt in 1957.

Step 4: Identify the era of Expert Systems (B)

Expert systems (like DENDRAL and MYCIN) were developed and became commercially successful primarily during the 1970s and 1980s.

Step 5: Determine the increasing chronological order

D (1837) → A (1950) → C (1957) → B (1970s/80s).

Quick Tip: Babbage (Mechanical) → Turing (Theoretical/Turing Test) → Rosenblatt (Connectionism/Perceptron) → Expert Systems (Symbolic AI). Timeline: Mid 19th Century → Mid 20th Century → Late 20th Century.

---

43. A process is organised in the memory from bottom to top ordering the following in correct order :

A. stack

B. text

C. data

D. heap

Choose the correct answer from the options given below :

(A) B, C, D, A

(B) B, D, C, A

(C) A, C, D, B

(D) A, D, C, B

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

Solution:

Concept:

- When a program is loaded into memory as a process, it is divided into logical segments.
- In a standard UNIX process layout, memory addresses increase from "bottom" (low addresses) to "top" (high addresses).

Step 1: Identify the bottom-most segment (B)

The **Text segment** (code) is placed at the lowest memory addresses to prevent it from being overwritten by heap or stack growth.

Step 2: Identify the segment above code (C)

The **Data segment** (containing initialized and uninitialized global/static variables) is placed immediately above the text segment.

Step 3: Identify the dynamically growing segment (D)

The **Heap** is used for dynamic memory allocation (e.g., `malloc` in C). It starts above the data segment and grows "upwards" toward higher addresses.

Step 4: Identify the top-most segment (A)

The **Stack** is used for local variables and function calls. It is placed at the very top of the available address space and grows "downwards" toward the heap.

Step 5: Arrange the sequence from bottom to top

The correct order is Text (B) → Data (C) → Heap (D) → Stack (A).

Quick Tip: Bottom (Low Addr): Text, Data. Middle: Heap (grows up). Top (High Addr): Stack (grows down). Unused space remains between the Heap and the Stack.

---

44. Arrange 5 stage instruction pipeline operation in correct order :

A. Instruction Fetch

B. Instruction Decode / Register Fetch

C. Execute / ALU operation

D. Memory access

E. Write Back

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Pipelining is a technique used in computer architecture to overlap the execution of multiple instructions.
- The classic RISC pipeline consists of five stages, where each instruction moves through these stages in a fixed sequential order.

Step 1: Identify the first stage (A)

IF (Instruction Fetch): The instruction is fetched from memory using the address in the Program Counter (PC).

Step 2: Identify the second stage (B)

ID (Instruction Decode): The instruction is decoded to identify the operation and the registers involved. Register values are also read.

Step 3: Identify the third stage (C)

EX (Execute): The Arithmetic Logic Unit (ALU) performs the actual computation (e.g., addition) or calculates an effective memory address.

Step 4: Identify the fourth stage (D)

MEM (Memory Access): If the instruction is a Load or Store, the processor accesses the data memory. Otherwise, the result bypasses this stage.

Step 5: Identify the final stage (E)

WB (Write Back): The final result (from ALU or Memory) is written back into the register file.

Quick Tip: The standard sequence is IF → ID → EX → MEM → WB. Memory access always happens after execution because we need the ALU to calculate the target address first.

---

45. Consider the number of symbols in data representation in increasing order :

A. Binary

B. Hexadecimal

C. Octal

D. Penta (radix 5)

Choose the correct answer from the options given below :

(A) A, D, C, B

(B) B, C, A, D

(C) A, D, B, C

(D) B, C, D, A

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

Solution:

Concept:

- In any positional number system, the "radix" or "base" defines the total number of unique symbols used to represent values.
- For a base r, the symbols used are the integers from 0 to $r-1$.

Step 1: Determine the number of symbols for each system

- **Binary (A):** Base is 2. Symbols are $\{0, 1\}$. Count $= 2$.
- **Penta (D):** Base is 5. Symbols are $\{0, 1, 2, 3, 4\}$. Count $= 5$.

- **Octal (C):** Base is 8. Symbols are {0, 1, 2, 3, 4, 5, 6, 7}. Count = 8.
- **Hexadecimal (B):** Base is 16. Symbols are {0-9, A-F}. Count = 16.

Step 2: Arrange in increasing order of symbols

Comparing the counts: $2 < 5 < 8 < 16$.

The corresponding labels are A $\rightarrow$ D $\rightarrow$ C $\rightarrow$ B.

Quick Tip: Radix = Number of Symbols. Higher radix systems allow representing large numbers with fewer digits but require a larger set of unique characters.

---

46. Consider the following recurrence relations. Their solutions have complexity in increasing order :

A. $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 1$

B. $T(n) = T(n-1) + n$

C. $T(n) = T(\lceil \frac{n}{2} \rceil) + 1$

D. $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$

Choose the correct answer from the options given below :

(A) C, D, A, B

(B) B, A, D, C

(C) C, A, D, B

(D) B, D, C, A

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

Solution:

Concept:

- Recurrence relations define the time complexity of recursive algorithms.
- We solve these using the Master Theorem or recursive tree methods to find their asymptotic bounds (Big-O notation).

Step 1: Solve Relation C

$T(n) = T(n/2) + 1$.

This is characteristic of Binary Search.

By Master Theorem ($a = 1, b = 2, d = 0$), since $a = b^d$ ($1 = 2^0$), the complexity is $O(\log n)$.

Step 2: Solve Relation D

$T(n) = 2T(n/2) + n$.

This is characteristic of Merge Sort.

By Master Theorem ($a = 2, b = 2, d = 1$), since $a = b^d$ ($2 = 2^1$), the complexity is $O(n \log n)$.

Step 3: Solve Relation A

$T(n) \approx 3T(n/2) + 1$.

By Master Theorem ($a = 3, b = 2, d = 0$), since $a > b^d$ ($3 > 1$), the complexity is $O(n^{\log_2 3}) \approx O(n^{1.58})$.

Step 4: Solve Relation B

$T(n) = T(n-1) + n$.

This is the sum of the first n integers: $n + (n-1) + (n-2) + \cdots + 1 = \frac{n(n+1)}{2}$.

The complexity is $O(n^2)$.

Step 5: Compare and Order

Comparing the growth rates: $\log n < n \log n < n^{1.58} < n^2$.

Order: C $\rightarrow$ D $\rightarrow$ A $\rightarrow$ B.

Quick Tip: Logarithmic < Linearithmic < Polynomial < Quadratic. When comparing recursive functions, check how many subproblems are created (the 'a' factor). More subproblems lead to higher complexity.

---

47. Let G be finite set, which is a group under some operation and H be a subgroup and $a \in H$. Considering their order / cardinality in increasing order :

A. $O(G)$

B. $O(a)$

C. $O(H)$

D. $Card(2^G)$

Choose the correct answer from the options given below :

(A) B, C, A, D

(B) D, B, C, A

(C) D, B, A, C

(D) B, C, D, A

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

Solution:

Concept:

- Group Theory properties define the relationships between orders of elements, subgroups, and parent groups.
- Lagrange's Theorem: For any finite group G, the order of every subgroup H of G divides the order of G.
- The order of an element a is the order of the cyclic subgroup generated by a.

Step 1: Relate the element order and subgroup order

Since $a \in H$, the cyclic subgroup $\langle a \rangle$ generated by a is a subgroup of H.

By Lagrange's theorem applied to H: $O(a)$ divides $O(H)$.

Therefore, $O(a) \leq O(H)$.

Step 2: Relate the subgroup order and group order

Since H is a subgroup of G, by Lagrange's theorem applied to G: $O(H)$ divides $O(G)$.

Therefore, $O(H) \leq O(G)$.

Step 3: Evaluate the cardinality of the power set

2^G represents the power set of G.

The cardinality is given by $Card(2^G) = 2^{O(G)}$.

For any finite set of size $n > 0$, $2^n > n$. Thus, $O(G) < Card(2^G)$.

Step 4: Arrange the sequence

The strictly increasing order is $O(a) \leq O(H) \leq O(G) < Card(2^G)$.

Labels: B $\rightarrow$ C $\rightarrow$ A $\rightarrow$ D.

Quick Tip: Sub-components always have a smaller or equal order compared to the whole. Element order $\leq$ Subgroup order $\leq$ Group order. Power set size always grows exponentially compared to the base set.

---

48. Consider the following steps of a data science workflow to be arranged in correct order :

A. Data preprocessing

B. Model building

C. Data collection

D. Data visualization and interpretation

E. Model evaluation

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- The Data Science Lifecycle (similar to CRISP-DM) is a standard process followed in data projects.
- It moves from data acquisition to preparation, modeling, and finally to communication of results.

Step 1: First Step - Data Collection (C)

You cannot perform any analysis without raw data. The process starts by gathering data from various sources (Databases, APIs, etc.).

Step 2: Second Step - Data Preprocessing (A)

Raw data is usually "dirty." This stage involves cleaning, handling missing values, and transforming data into a format suitable for algorithms.

Step 3: Third Step - Model Building (B)

Once the data is ready, machine learning models are trained using chosen algorithms and features.

Step 4: Fourth Step - Model Evaluation (E)

The trained model is tested against unseen data using metrics like accuracy, precision, or RMSE to ensure it performs well.

Step 5: Fifth Step - Visualization and Interpretation (D)

The final findings and model outputs are visualized for stakeholders to interpret and make data-driven decisions.

Quick Tip: "Collect, Clean, Create, Check, Communicate." Collection always starts the flow, and Interpretation/Visualization (reporting) usually concludes it.

49. Consider the following statements in C to write a program in their order of occurrence :

A. $Printf(\text{"\%d"}, i);$

B. $void\ main\ (\ )$

C. $int\ i = 5;$

D. $\#include\ <stdio.h>$

Choose the correct answer from the options given below :

(A) D, B, C, A

(B) B, D, C, A

(C) B, C, A, D

(D) D, B, A, C

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

Solution:

Concept:

- A C program follows a strict structural hierarchy required by the compiler.
- Components must be declared or defined before they are used in statements.

Step 1: Preprocessor Directives (D)

The `#include` statement must come first so that the compiler can load definitions for standard functions like `printf`.

Step 2: Function Definition (B)

The entry point of every C program is the `main()` function. It follows the include section.

Step 3: Variable Declaration and Initialization (C)

Inside the main function, variables must be declared (and optionally initialized) before any logic uses them.

Step 4: Executable Statements (A)

The actual logic, such as printing the value of the variable, comes after variables are initialized.

Step 5: Synthesize the code

```
#include <stdio.h>
void main() {
    int i = 5;
```



```
    printf("%d", i);
}
```

Sequence: D → B → C → A.

Quick Tip: Pre-processor → Header → Body. Declaration before use is a fundamental rule in C. You cannot print 'i' if the compiler hasn't seen the 'int i' line yet.

---

50. Consider the flags in flag Register from left to right :

A. S

B. Z

C. AC

D. P

E. CY

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- In the 8085 microprocessor, the Flag Register is an 8-bit register where specific bits represent the status of the ALU after an operation.
- "Left to right" in register notation means from the Most Significant Bit (MSB) to the Least Significant Bit (LSB).

Step 1: Identify the 8085 Flag Register Architecture

The bits are arranged as follows:

- Bit 7: **S** (Sign Flag)
- Bit 6: **Z** (Zero Flag)

- Bit 5: X (Undefined)
- Bit 4: **AC** (Auxiliary Carry Flag)
- Bit 3: X (Undefined)
- Bit 2: **P** (Parity Flag)
- Bit 1: X (Undefined)
- Bit 0: **CY** (Carry Flag)

Step 2: Arrange the given flags from Bit 7 to Bit 0

Scanning from left (Bit 7) to right (Bit 0), we find:

S (A) → Z (B) → AC (C) → P (D) → CY (E).

Step 3: Verify labels

The sequence A, B, C, D, E correctly maps to the physical layout of the status bits in the register.

Quick Tip: Mnemonic to remember the 8085 flags: "Some Zeros Are Pretty Cool" (S, Z, AC, P, CY). Skip bits 5, 3, and 1 as they are not assigned to any specific condition.

---

51. Consider the 802.3 MAC frame fields from left to right :

A. Preamble

B. Source address

C. SFD

D. Destination address

Choose the correct answer from the options given below :

(A) A, C, D, B

(B) A, C, B, D

(C) D, B, C, A

(D) B, D, C, A

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

Solution:

Concept:

- The IEEE 802.3 standard defines the MAC (Media Access Control) frame format for Ethernet networks.
- A frame is the unit of data transmitted at the Data Link layer.
- The fields must appear in a specific sequence to allow synchronization, addressing, and error checking between the sender and receiver.

Step 1: Identify the synchronization components (A and C)

The frame begins with a 7-byte **Preamble** (A) consisting of alternating 1s and 0s to synchronize the receiver's clock. This is immediately followed by the 1-byte **SFD** (C - Start Frame Delimiter), which signifies the beginning of the actual frame data.

Step 2: Identify the addressing components (D and B)

The network needs to know where the packet is going first so that non-target devices can stop processing the frame early. Therefore, the **Destination Address** (D) comes before the **Source Address** (B). Both are 6-byte MAC addresses.

Step 3: Arrange the sequence from left to right

Preamble (A) → SFD (C) → Destination address (D) → Source address (B).

This matches the sequence in Option (A).

Quick Tip: "Destination comes first." In Ethernet frames, the receiver needs to see the Destination MAC immediately after the synchronization bytes to decide whether to accept or discard the frame, saving processing power.

---

52. Consider the following statements :

A. Membership problem is decidable for regular and CFL languages.

B. Emptiness problem is undecidable for RE languages.

C. Finiteness problem is decidable for regular languages.

D. Equivalence problem is decidable for context-free and RE languages.

Choose the correct answer from the options given below :

(A) A, B, C only

(B) B, C, D only

(C) C, D, A only

(D) A, B only

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

Solution:

Concept:

- Decidability refers to whether a problem can be solved by a Turing machine in a finite amount of time.
- Different classes of languages (Regular, CFL, CSL, RE) have different decidability properties for common questions like Membership, Emptiness, and Equivalence.

Step 1: Evaluate Statement A (Membership)

For Regular languages, we can simply run the string through a DFA. For CFLs, we can use the CYK algorithm or a PDA. Both are guaranteed to terminate. Thus, A is **correct**.

Step 2: Evaluate Statement B (Emptiness for RE)

Determining if a Recursively Enumerable (RE) language is empty is equivalent to determining if a Turing Machine accepts no strings. This is a variation of the Halting Problem and is undecidable. Thus, B is **correct**.

Step 3: Evaluate Statement C (Finiteness for Regular)

For a Regular language (DFA), we can check if there is a cycle in the state transition graph that is reachable from the start state and can reach a final state. This is an algorithmic check. Thus, C is **correct**.

Step 4: Evaluate Statement D (Equivalence for CFL/RE)

Equivalence is undecidable for Context-Free Languages (one cannot even decide if two CFGs generate the same language). It is also undecidable for RE languages. Thus, D is **incorrect**.

Quick Tip: Regular languages are "Decidable everything." Context-Free languages are "Decidable Membership/Emptiness" but "Undecidable Equivalence." RE languages are "Undecidable everything" (except sometimes membership for Recursive sub-class).

---

53. Some Scientists, well known in AI domain have given AI definitions :

A. The Science and Engineering of making intelligent machines...

B. The study of mental faculties through the use of computational models.

C. The field of study that seeks to explain and emulate intelligent behaviour...

D. The study of how to make computer do things at which... people are better.

E. The study and analyses of computing theory.

Choose the correct answer from the options given below :

(A) A, B, C, D only

(B) B, C, D, E only

(C) C, E, A only

(D) E, A, B only

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

Solution:

Concept:

- AI definitions usually fall into four categories: Thinking Humanly, Acting Humanly, Thinking Rationally, and Acting Rationally.
- Famous computer scientists have defined AI based on these perspectives over the last 70 years.

Step 1: Validate AI-specific definitions (A-D)

- **A:** This is John McCarthy's (the father of AI) famous definition focusing on engineering.
- **B:** This reflects the cognitive modeling perspective (Charniak and McDermott).
- **C:** This reflects the behavior emulation and rational agent perspective (Schalkoff).
- **D:** This is Elaine Rich's widely used definition about computers surpassing human performance in specific tasks.

Step 2: Evaluate Statement E

"The study and analyses of computing theory" is a general definition of **Computer Science** as a whole. While AI is a subset of this, it is not a definition of AI itself.

Step 3: Conclusion

Since A, B, C, and D are specific, recognized definitions of Artificial Intelligence, the correct choice is Option (A).

Quick Tip: AI definitions often mention "Intelligence", "Human-like", or "Rationality". General statements about "Computing theory" or "Hardware" usually refer to Computer Science or Engineering, not AI specifically.

---

54. Consider the following statements about CPU scheduling :

A. First come, first serve can lead to the "convoy effect".

B. Shortest job first minimizes average waiting time.

C. Round robin does not use time quantum for fairness.

D. Priority scheduling may cause starvation.

E. Multilevel feedback queue combines preemptive and non preemptive scheduling.

Choose the correct answer from the options given below :

(A) A, B, C, E only

(B) B, C, D, E only

(C) A, B, D, E only

(D) A, C, D only

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

Solution:

Concept:

- CPU scheduling algorithms determine which process in the ready queue is allocated to the CPU.
- Each algorithm has specific pros (like fairness or low wait time) and cons (like starvation or the convoy effect).

Step 1: Evaluate FCFS and SJF (A and B)

- **Convoy Effect:** In FCFS, if a long CPU-bound process starts first, all short I/O-bound processes wait behind it. Statement A is **correct**.
- **Optimality of SJF:** SJF is provably optimal for minimizing the average waiting time for a given set of processes. Statement B is **correct**.

Step 2: Evaluate Round Robin and Priority (C and D)

- **Round Robin:** This algorithm relies **entirely** on a "time quantum" to ensure no process hogs the CPU. Statement C is **incorrect**.
- **Starvation:** In Priority scheduling, a low-priority process may never run if high-priority processes are constantly added to the queue. Statement D is **correct**.

Step 3: Evaluate Multilevel Feedback Queue (E)

MLFQ uses multiple queues with different priorities and different scheduling algorithms (some might be preemptive like RR, some non-preemptive). It moves processes between queues based on behavior. Statement E is **correct**.

Quick Tip: SJF = Best for Average Wait Time. Round Robin = Best for Response Time (needs Time Quantum). Priority = Risks Starvation (needs Aging).

---

55. Consider the following statements about bus arbitration schemes :

A. Daisy chaining is simple but slow.

B. Parallel priority arbitration provides faster response than serial methods.

C. Distributed arbitration allows multiple masters to share bus control.

D. Centralized arbitration increases fairness.

Choose the correct answer from the options given below :

(A) A, C, D only

(B) A, B, C only

(C) B, C, D only

(D) A, B, D only

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

Solution:

Concept:

- Bus arbitration is the process by which the next "bus master" is selected when multiple devices request access to the shared bus.
- It can be categorized as Centralized (one arbiter) or Distributed (shared logic).

Step 1: Analyze Daisy Chaining (A)

In Daisy Chaining (a serial centralized scheme), the grant signal passes through each device in sequence. It is hardware-simple but the delay is proportional to the number of devices. Thus, A is **correct**.

Step 2: Analyze Parallel Arbitration (B)

Parallel schemes use independent request and grant lines for each device. All requests are evaluated simultaneously by the arbiter, leading to a much faster response than the serial daisy chain. Thus, B is **correct**.

Step 3: Analyze Distributed Arbitration (C)

In distributed schemes, each device has its own arbitration logic and they "compete" or "negotiate" for the bus. There is no single master controller. Thus, C is **correct**.

Step 4: Evaluate Fairness (D)

Centralized arbitration does not inherently increase fairness. In fact, basic centralized schemes like static priority daisy chains are notoriously unfair to devices far down the line. Fairness depends on the specific logic (like Round Robin) used, not whether the hardware is centralized or distributed. Thus, D is generally considered **incorrect** in a technical comparison.

Quick Tip: Daisy Chain = Serial, Simple, Slow. Parallel = Fast, Complex, More lines. Distributed = No single point of failure (no central arbiter).

---

56. Consider the following statements about flow and error control :

A. Go-Back-N retransmits only the frame that is lost

B. Selective Repeat improves the bandwidth utilization over Go-Back-N

C. Parity bits can detect all single-bit errors

D. Stop and wait protocol achieves high channel utilization on long-delay networks

E. Cyclic Redundancy check provides stronger error detection than single parity

Choose the correct answer from the options given below :

(A) A, B, E only

(B) A, C, D only

(C) B, C, D only

(D) B, C, E only

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

Solution:

Concept:

- **Flow Control:** Techniques like Stop-and-Wait, Go-Back-N, and Selective Repeat manage the rate of data transmission between two nodes.
- **Error Control:** Mechanisms like Parity bits and CRC detect and correct errors introduced during transmission.

Step 1: Evaluate Sliding Window protocols (A and B)

In **Go-Back-N**, when a frame is lost, the sender retransmits that frame and all subsequent frames in the current window. In **Selective Repeat**, only the specifically lost frame is retransmitted. Therefore, Selective Repeat is more efficient and uses bandwidth better. Statement A is **Incorrect**, and Statement B is **Correct**.

Step 2: Evaluate Stop-and-Wait utilization (D)

The utilization of Stop-and-Wait is $U = 1/(1+2a)$, where a is the ratio of propagation delay to transmission delay. On long-delay networks, a is very large, making U very small. Thus, it is inefficient. Statement D is **Incorrect**.

Step 3: Evaluate Error Detection techniques (C and E)

A single **Parity bit** can detect any odd number of bit errors, including all single-bit errors. **CRC** (Cyclic Redundancy Check) uses polynomial division to detect a wide variety of error patterns, including burst errors, making it much stronger than simple parity. Both Statements C and E are **Correct**.

Step 4: Conclusion

The correct statements are B, C, and E.

Quick Tip: Go-Back-N = Cumulative ACK, simpler logic, lower utilization on loss.
Selective Repeat = Independent ACK, complex buffer, maximum utilization.
CRC is the industry standard for hardware-level error detection.

---

57. Consider the following statements about symmetric encryption :

A. Same key for encryption and decryption

B. Faster than asymmetric encryption

C. Key distribution is simpler

D. Stream ciphers encrypt one bit / byte at a time

E. Block ciphers operate on groups of bits

Choose the correct answer from the options given below :

(A) A, B, C, E only

(B) B, C, D only

(C) A, C, D, E only

(D) A, B, D, E only

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

Solution:

Concept:

- **Symmetric Encryption:** A cryptographic system that uses a single shared secret key for both the encryption of plaintext and the decryption of ciphertext.
- Common types include DES, AES (block ciphers), and RC4 (stream ciphers).

Step 1: Evaluate the fundamental characteristics (A and B)

Symmetric encryption is defined by using a single shared key. Because the mathematical operations involved (like XOR and substitution) are less complex than the modular exponentiation used in Asymmetric systems (like RSA), symmetric encryption is significantly faster. Statements A and B are **Correct**.

Step 2: Evaluate the Key Distribution problem (C)

One of the major disadvantages of symmetric encryption is that the secret key must be shared securely between parties. As the number of users n grows, the number of keys required grows as $O(n^2)$. This is much more complex than asymmetric systems. Statement C is **Incorrect**.

Step 3: Evaluate Ciphers types (D and E)

Stream ciphers convert plaintext to ciphertext by processing one bit or byte at a time using a keystream. **Block ciphers** divide the plaintext into fixed-size blocks (e.g., 64 or 128 bits) and process them as a single unit. Statements D and E are **Correct**.

Step 4: Conclusion

The correct set of statements is A, B, D, and E.

Quick Tip: Symmetric = High Speed + High Security + Hard Key Management.

Use Symmetric for data transfer and Asymmetric (like RSA) for the initial key exchange.

58. Consider following numbers :

A. $(1101.01)_2 = (13.25)_{10}$

B. $(13.25)_{10} = (1101.011)_2$

C. $(2A)_{16} = (42)_{10}$

D. $(37)_8 = (11010)_2$

Choose the correct answer from the options given below :

(A) A, B, C only

(B) B, D only

(C) C, D only

(D) A, C only

Correct Answer: (D) A, C only

Solution:

Concept:

- Positional number systems allow values to be converted between bases by expanding the digits with their corresponding powers of the base.

Step 1: Verify Statement A

$(1101.01)_2 = (1 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0) + (0 \cdot 2^{-1}) + (1 \cdot 2^{-2})$

$= 8 + 4 + 0 + 1 + 0 + 0.25 = 13.25_{10}$. Statement A is **Correct**.

Step 2: Verify Statement B

From Step 1, we see that $13.25_{10} = (1101.01)_2$. The binary value $(1101.011)_2$ would equal $13.25 + (1 \cdot 2^{-3}) = 13.25 + 0.125 = 13.375_{10}$. Statement B is **Incorrect**.

Step 3: Verify Statement C

$(2A)_{16} = (2 \cdot 16^1) + (A \cdot 16^0)$. Since $A = 10$:

$= 32 + 10 = 42_{10}$. Statement C is **Correct**.

Step 4: Verify Statement D

$(37)_8 = (3 \cdot 8^1) + (7 \cdot 8^0) = 24 + 7 = 31_{10}$.

Binary $(11010)_2 = (1 \cdot 2^4) + (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0) = 16 + 8 + 0 + 2 + 0 = 26_{10}$.

Since $31 \neq 26$, Statement D is **Incorrect**.

Quick Tip: To quickly convert Octal to Binary: Convert each digit to its 3-bit binary equivalent ($3 \rightarrow 011$, $7 \rightarrow 111$). Thus, $(37)_8 = (011111)_2 = 31_{10}$.

---

59. Consider the following statements :

A. MOV A, M copies data from memory (pointed by HL) to accumulator

B. LXI H, 2025H loads 2025H into register pair HL

C. STA 2050H stores the accumulator content into memory at address 2050H

D. LDA 2050H loads HL with the contents of memory location at 2050H

Choose the correct answer from the options given below :

(A) A, B, C only

(B) B, C, D only

(C) A, D only

(D) A, C, D only

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

Solution:

Concept:

- The 8085 microprocessor uses various data transfer instructions to move 8-bit or 16-bit data between registers and memory.

Step 1: Evaluate indirect memory transfer (A)

The register 'M' in 8085 instructions refers to the memory location pointed to by the address stored in the HL register pair. `MOV A, M` correctly moves data from that memory address to the Accumulator. Statement A is **Correct**.

Step 2: Evaluate immediate register pair loading (B)

`LXI` stands for Load Register Pair Immediate. `LXI H, 2025H` loads the 16-bit constant 2025H into the HL pair. Statement B is **Correct**.

Step 3: Evaluate direct memory storage (C)

`STA` stands for Store Accumulator Direct. It copies the content of the Accumulator into the memory location specified by the 16-bit address in the instruction. Statement C is **Correct**.

Step 4: Evaluate direct memory loading (D)

`LDA` stands for Load Accumulator Direct. It copies the content of the specified memory address into the **Accumulator**, not the HL pair. To load HL from memory, one would use `LHLD`.

Statement D is **Incorrect**.

Quick Tip: STA / LDA = Direct addressing (Address is in instruction).
MOV A, M / MOV M, A = Indirect addressing (Address is in HL).
'X' in LXI or INX indicates the instruction operates on a 16-bit Register Pair.

---

60. Consider following statements about time complexity :

A. Merge sort and heap sort have O (n log n) in worst case

B. Quick sort has O (n log n) in average case and O (n^2) in worst case

C. Insertion sort is faster than merge sort for large n

D. Bubble sort is stable

E. Selection sort has fewer swap than insertion sort

Choose the correct answer from the options given below :

(A) B, C, E only

(B) A, C, D only

(C) A, B, C only

(D) A, B, D, E only

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

Solution:

Concept:

- Algorithm analysis involves evaluating the performance (Time and Space complexity) of different sorting methods under various conditions (Worst, Average, and Best case).

Step 1: Evaluate log-linear sorts (A and B)

Merge sort and Heap sort both use a divide-and-conquer strategy (logarithmic levels) and process every element (linear work), leading to a guaranteed $O(n \log n)$ complexity in all cases. Quick sort is also $O(n \log n)$ on average but degrades to $O(n^2)$ if the pivot choices are poor (e.g., sorted arrays). Statements A and B are **Correct**.

Step 2: Evaluate relative performance (C)

For large n, an $O(n \log n)$ algorithm (Merge sort) will always eventually be faster than an $O(n^2)$ algorithm (Insertion sort). Insertion sort is only faster for very small arrays. Statement C is **Incorrect**.

Step 3: Evaluate Stability (D)

A sorting algorithm is stable if it preserves the relative order of elements with equal keys. Bubble sort only swaps adjacent elements if they are strictly out of order, making it stable. Statement D is **Correct**.

Step 4: Evaluate Swap efficiency (E)

Selection sort finds the minimum in each pass and performs exactly one swap. Insertion sort may perform up to $O(n)$ swaps (or rather, shifts) for a single element to reach its destination. In terms of memory writes/swaps, Selection sort is superior. Statement E is **Correct**.

Quick Tip: Selection Sort = Minimum Swaps.

Insertion Sort = Best for almost-sorted small data.

Merge Sort = Stable $O(n \log n)$ but needs extra space.

---

61. Consider a binary operation $*$ on set Z (set of integers) defined as $a * b = a + b + 1$ then :

A. $*$ is commutative

B. $*$ is associative

C. Identity element under $*$ exists

D. Every element has an inverse under $*$

E. The structure $(Z, *)$ is not a group

Choose the correct answer from the options given below :

(A) A, B, E only

(B) B, D, E only

(C) A, C, D, E only

(D) A, B, C, D only

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

Solution:

Concept:

- A binary operation $*$ on a set G forms a group if it satisfies four properties: Closure, Associativity, Identity, and Inverse.
- If it also satisfies Commutativity, it is an Abelian group.

Step 1: Verify Commutativity (A)

We check if $a * b = b * a$ for all $a, b \in Z$.

$a * b = a + b + 1$

$b * a = b + a + 1$

Since integer addition is commutative ($a + b = b + a$), $a * b = b * a$ is true. Statement A is correct.

Step 2: Verify Associativity (B)

We check if $(a * b) * c = a * (b * c)$.

LHS: $(a * b) * c = (a + b + 1) * c = (a + b + 1) + c + 1 = a + b + c + 2$.

RHS: $a * (b * c) = a * (b + c + 1) = a + (b + c + 1) + 1 = a + b + c + 2$.

LHS = RHS. Statement B is correct.

Step 3: Find the Identity Element (C)

Let e be the identity such that $a * e = a$.

$a + e + 1 = a \implies e + 1 = 0 \implies e = -1$.

Since -1 is an integer ($-1 \in Z$), the identity element exists. Statement C is correct.

Step 4: Find the Inverse for every element (D)

Let b be the inverse of a such that $a * b = e$.

$a + b + 1 = -1 \implies b = -a - 2$.

For any integer a, the value $-a - 2$ is also an integer. Statement D is correct.

Step 5: Determine if it is a group (E)

Since properties A, B, C, and D are satisfied, $(Z, *)$ is indeed a group.

Therefore, statement E ("is not a group") is false.

Quick Tip: To find the identity, set $a * e = a$ and solve for e. To find the inverse, set $a * b = e$ and solve for b. If both results stay within the specified set (Integers here), the group properties hold.

---

62. Consider following statements :

A. Regression predicts categorical outcomes.

B. Classification models are used for continuous variable prediction.

C. Clustering is an unsupervised learning task.

D. Reinforcement learning uses rewards and penalties to improve performance.

E. Decision tree can be used for both classification and regression.

Choose the correct answer from the options given below :

(A) A, B, E only

(B) A, C, D only

(C) C, D, E only

(D) B, C, E only

Correct Answer: (C) C, D, E only

Solution:

Concept:

- Supervised learning is divided into Classification (discrete) and Regression (continuous).
- Unsupervised learning deals with unlabeled data (Clustering).
- Reinforcement learning involves an agent learning through environmental feedback.

Step 1: Evaluate Regression and Classification (A, B)

Regression models predict numerical/continuous values (e.g., house prices). Classification models predict discrete labels/categories (e.g., spam vs. not spam). Statement A and B have their definitions swapped, so both are incorrect.

Step 2: Evaluate Clustering (C)

Clustering is the task of grouping a set of objects such that objects in the same group are more similar to each other than to those in other groups. Since this is done without target labels, it is an unsupervised task. Statement C is correct.

Step 3: Evaluate Reinforcement Learning (D)

In reinforcement learning, an agent learns to behave in an environment by performing actions and seeing the results as rewards or penalties. This is the core mechanism of the paradigm. Statement D is correct.

Step 4: Evaluate Decision Trees (E)

Decision trees can handle both categorical targets (Classification trees) and continuous targets (Regression trees). This versatility is a key feature of the CART (Classification and Regression Trees) algorithm. Statement E is correct.

Quick Tip: Continuous Target → Regression.

Discrete/Categorical Target → Classification.

Unlabeled Data → Clustering.

63. Consider the following statements :

A. In a binary search tree, left sub tree has smaller value than right sub tree

B. In doubly linked list, only forward traversal is possible

C. In C, a function may not always return a value

D. Two dim array requires two index variables

E. A tree must have at least two nodes

Choose the correct answer from the options given below :

(A) $A, B, C\ only$

(B) $B, A, D\ only$

(C) $C, D, E\ only$

(D) $D, C, A\ only$

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

Solution:

Concept:

- Binary Search Trees (BST) maintain a specific ordering to facilitate efficient searching.
- Different data structures (Lists, Arrays, Trees) have unique properties regarding traversal and connectivity.
- Programming languages like C have specific rules for function definitions.

Step 1: Analyze BST properties (A)

In a BST, for any node N, all values in the left subtree are smaller than N, and all values in the right subtree are larger than N. Consequently, the left subtree values are always smaller than the right subtree values. Statement A is correct.

Step 2: Analyze Doubly Linked Lists (B)

A doubly linked list contains two pointers per node: `next` and `prev`. This allows for bidirectional traversal (both forward and backward). Statement B is incorrect.

Step 3: Analyze C function returns (C)

In C, a function declared with the `void` return type does not return a value. For example, `void display() { ... }`. Therefore, it is true that a function may not always return a value. Statement C is correct.

Step 4: Analyze 2D Array indexing (D)

To access a specific element in a two-dimensional array, one must specify both the row index

and the column index (e.g., `arr[i][j]`). Thus, two index variables are required. Statement D is correct.

Step 5: Analyze the definition of a Tree (E)

A tree can be empty (a NULL tree) or consist of only a single node (the root). It does not strictly require two nodes. Statement E is incorrect.

Quick Tip: BST: Left < Root < Right.

Doubly Linked List: "Next" and "Prev" pointers.

A function with 'void' return type is used for procedures that don't need to return data.

---

64. Match List - I with List - II.

	List - I		List - II
A.	Cook's Theorem	I.	The Boolean satisfiability problem (SAT) is NP complete
B.	Pumping lemma for regular languages	II.	Used to prove non-regularity of certain languages
C.	Closure under complementation	III.	Fails for context free language
D.	Non-deterministic pushdown automata	IV.	Recognises all context - free languages

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Theory of computation deals with the complexity of problems and the characteristics of different language classes (Regular, Context-Free, etc.).

Step 1: Identify Cook's Theorem (A-I)

Cook's Theorem is a cornerstone of complexity theory. It proved that the Boolean Satisfiability Problem (SAT) is NP-complete, providing the first known member of this class.

Step 2: Identify Pumping Lemma usage (B-II)

The Pumping Lemma provides a necessary condition that all regular languages must satisfy. It is typically used as a tool for proof by contradiction to show that a specific language is *not* regular.

Step 3: Identify CFL Closure properties (C-III)

Context-Free Languages (CFL) are closed under Union, Concatenation, and Kleene Star. However, they are notably **not** closed under Intersection or Complementation. Thus, closure under complementation fails for CFL.

Step 4: Identify the power of NPDA (D-IV)

A Non-deterministic Pushdown Automaton (NPDA) is exactly the machine that accepts the class of Context-Free Languages. Note that Deterministic PDAs only accept a subset (DCFL).

Quick Tip: Regular languages = Closed under all basic operations.

Context-Free languages = NOT closed under intersection and complement.

NP-Complete = The set of hardest problems in NP.

---

65. Match List - I with List - II.

	List - I		List - II
A.	Modularity	I.	Bounded rationality
B.	Representation	II.	Offline
C.	Computational limits	III.	Flat
D.	Interaction	IV.	Features

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- These terms relate to how AI systems and cognitive agents are architected and how they process information.

Step 1: Match Modularity (A-III)

In the design of intelligent agents, "Modularity" refers to a system composed of discrete, specialized units. The opposite of a modular or hierarchical structure is a "Flat" structure, where all elements are at the same level of organization.

Step 2: Match Representation (B-IV)

In Artificial Intelligence and Machine Learning, "Representation" refers to how raw data is mapped into a form the system can use. This is primarily done through "Features" (individual independent variables).

Step 3: Match Computational limits (C-I)

"Bounded Rationality" is the idea that decision-making is limited by the information available, the cognitive limitations of the mind, and the finite time available. It describes the realistic computational limits of an agent.

Step 4: Match Interaction (D-II)

Interaction modes can be described as "Online" (interacting in real-time) or "Offline" (where learning or processing happens based on pre-collected data without live environment feedback).

Quick Tip: Bounded Rationality = Humans/Agents make the best decision possible given their limits. Modularity vs. Flat = Architecture of the system's knowledge or functions.

---

66. Match List - I with List - II.

	List - I		List - II
A.	Deadlock prevention	I.	Banker's algorithm
B.	Deadlock detection	II.	Rollbock
C.	Deadlock avoidance	III.	Circular wait
D.	Recovery from deadlock	IV.	High overheads

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Deadlock is a state in which a set of processes are blocked because each process is holding a resource and waiting for another resource held by another process.
- **Prevention:** Ensuring that at least one of the four necessary conditions (Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait) can never hold.
- **Avoidance:** Dynamically deciding whether a resource allocation is safe using algorithms like Banker's.
- **Detection and Recovery:** Periodically checking the system state for cycles and taking corrective action.

Step 1: Match A with III (Deadlock Prevention)

Deadlock prevention works by denying one of the four necessary conditions. A classic method is preventing "Circular Wait" by imposing a linear ordering of resource types. So, A matches with III.

Step 2: Match C with I (Deadlock Avoidance)

Deadlock avoidance requires the OS to know in advance the maximum resources a process will ever request. The "Banker's Algorithm" uses this to ensure the system stays in a "safe state." So, C matches with I.

Step 3: Match D with II (Recovery from Deadlock)

Once a deadlock is detected, the system must recover. One way is "Rollback," which involves returning a process to a previous "safe" checkpoint so it can release its resources. So, D matches with II.

Step 4: Match B with IV (Deadlock Detection)

Detection algorithms maintain resource graphs and check for cycles. Because these checks are computationally expensive and run frequently, they impose "High Overheads" on the system. So, B matches with IV.

Quick Tip: Prevention = Structural solution.

Avoidance = Dynamic check (Banker's).

Detection = Costly checking.

Recovery = Corrective action (Rollback).

---

67. Match List - I with List - II.

	List - I		List - II
A.	Pipeline bubble	I.	A deliberate delay or idle slot inserted in pipeline
B.	Pipeline flush	II.	Clearing all in progress instructions
C.	Data forwarding	III.	Using result before it is written back
D.	Pipeline stall	IV.	Holding stages due to dependency or resource conflict

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Pipelining improves instruction throughput by overlapping different stages of instruction execution.
- Hazards (Data, Control, or Structural) are situations that prevent the next instruction from executing in the next clock cycle.

Step 1: Match A with I (Pipeline Bubble)

A "bubble" is essentially a NOP (No-Operation) that moves through the pipeline. It is a "deliberate delay or idle slot" used to resolve hazards. So, A matches with I.

Step 2: Match B with II (Pipeline Flush)

When a branch prediction is incorrect, the instructions currently in the earlier stages are wrong. The pipeline must be "flushed" by "clearing all in-progress instructions." So, B matches with II.

Step 3: Match C with III (Data Forwarding)

Data forwarding (or bypassing) is a hardware technique where the result of an operation is routed directly to the next instruction "before it is written back" to the register file. So, C matches with III.

Step 4: Match D with IV (Pipeline Stall)

A stall occurs when the control unit "holds stages" due to a "dependency or resource conflict" that cannot be resolved by forwarding, forcing the instruction to wait. So, D matches with IV.

Quick Tip: Forwarding = Speed up (bypass registers).
Bubble/Stall = Slow down (insertion of idle time).
Flush = Clean up (after mispredicted branch).

---

68. Match List - I with List - II.

	List - I		List - II
A.	Ethernet	I.	Uses token passing and deterministic access
B.	Token ring	II.	Uses CSMA/CD for medium access
C.	Wi-Fi (802.11)	III.	Uses CSMA/CA to avoid collusion
D.	Bluetooth	IV.	IEEE 802.15.1

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Medium Access Control (MAC) protocols determine how multiple nodes share the same physical channel.
- Standard bodies like IEEE define specific protocols (802.3, 802.11, etc.) for different network technologies.

Step 1: Match A with II (Ethernet)

Standard Ethernet (IEEE 802.3) uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to manage access on a shared bus. So, A matches with II.

Step 2: Match B with I (Token Ring)

IEEE 802.5 (Token Ring) uses a "token passing" protocol. It provides "deterministic access" because a station only transmits when it holds the token. So, B matches with I.

Step 3: Match C with III (Wi-Fi)

IEEE 802.11 (Wi-Fi) uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) because collision detection is difficult to implement in wireless channels. So, C matches with III.

Step 4: Match D with IV (Bluetooth)

Bluetooth is the commercial name for the IEEE 802.15.1 standard, which focuses on Wireless Personal Area Networks (WPAN). So, D matches with IV.

Quick Tip: Ethernet = 802.3 = CD (Detection).
Wi-Fi = 802.11 = CA (Avoidance).
Bluetooth = 802.15.

---

69. Match List - I with List - II.

	List - I		List - II
A.	Caesar Cipher	I.	Substitution cipher using 5×5 matrix
B.	Playfair Cipher	II.	Polyalphabetic cipher using matrix multiplication
C.	Hill Cipher	III.	Monoalphabetic cipher using fixed shift
D.	Vigenere Cipher	IV.	Polyalphabetic cipher using keyboard repetition

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Cryptography techniques are classified based on how characters are substituted.
- **Monoalphabetic:** One fixed substitution for each character throughout the message.
- **Polyalphabetic:** Substitutions change based on the position of the character and a key.

Step 1: Match A with III (Caesar Cipher)

The Caesar cipher is the most basic "monoalphabetic" cipher where each letter is shifted by a "fixed" number of positions (e.g., $n = 3$). So, A matches with III.

Step 2: Match B with I (Playfair Cipher)

The Playfair cipher is a manual symmetric encryption technique that uses a "5 × 5 matrix" of letters to encrypt pairs of letters (digrams). So, B matches with I.

Step 3: Match C with II (Hill Cipher)

The Hill cipher is a "polyalphabetic" substitution cipher based on linear algebra. It uses "matrix multiplication" to encrypt blocks of text. So, C matches with II.

Step 4: Match D with IV (Vigenere Cipher)

The Vigenere cipher uses a keyword that is repeated across the plaintext. The "keyboard repetition" refers to the periodic shift change based on the keyword letters. So, D matches with IV.

Quick Tip: Caesar = Fixed Shift.

Playfair = Matrix Grid.

Hill = Matrix Multiplication.

Vigenere = Repeated Keyword.

---

70. Match List - I with List - II.

	List - I		List - II
A.	$x\ (x' + y)$	I.	$xy + x'z$
B.	$x'y'z + x'yz + xy'$	II.	$(x + y)\ (x' + z)$
C.	$xy + x'z + yz$	III.	xy
D.	$(x + y)\ (x' + z)\ (y + z)$	IV.	$x'z + xy'$

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Boolean algebra simplification relies on basic laws like Distributive Law ($A(B + C) = AB + AC$), Complement Law ($AA' = 0$), and the Consensus Theorem ($AB + A'C + BC = AB + A'C$).

Step 1: Simplify expression A

Using Distributive Law and Complement Law:

$$\begin{aligned} x(x' + y) &= x \cdot x' + x \cdot y \\ &= 0 + xy \\ &= xy \end{aligned}$$

This matches with III. So, A-III.

Step 2: Simplify expression B

Factoring out $x'z$ from the first two terms:

$$\begin{aligned} x'y'z + x'yz + xy' &= x'z(y' + y) + xy' \\ &= x'z(1) + xy' \\ &= x'z + xy' \end{aligned}$$

This matches with IV. So, B-IV.

Step 3: Simplify expression C using Consensus Theorem

The Consensus Theorem states: $AB + A'C + BC = AB + A'C$. Here, let $A = x, B = y, C = z$:

$$xy + x'z + yz = xy + x'z$$

The term yz is redundant. This matches with I. So, C-I.

Step 4: Simplify expression D using Dual Consensus Theorem

The Dual Consensus Theorem states: $(A + B)(A' + C)(B + C) = (A + B)(A' + C)$. Here, let $A = x, B = y, C = z$:

$$(x + y)(x' + z)(y + z) = (x + y)(x' + z)$$

This matches with II. So, D-II.

Quick Tip: Consensus Theorem trick: Look for three variables appearing twice. If one variable appears once as A and once as A', the third term without A or A' is redundant.

---

71. Match List - I with List - II.

	List - I		List - II
A.	Machine cycle	I.	Time required to complete one operation (Fetch/read/write)
B.	T-state	II.	One subdivision of a machine cycle
C.	Instruction cycle	III.	Time required to complete one instruction
D.	Opcode fetch	IV.	Machine cycle to get instruction opcode from memory

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- CPU timing terminology describes the hierarchy of time units required for a processor to

execute commands.

- **Instruction Cycle:** The total time from the start of fetching an instruction to the completion of its execution.
- **Machine Cycle:** The time required to perform a single basic operation involving the system bus, such as memory read, memory write, I/O read, or I/O write.
- **T-state:** The smallest unit of time, corresponding to one period of the system clock. A machine cycle consists of multiple T-states.

Step 1: Match A (Machine cycle)

A machine cycle is the basic unit of operation for a microprocessor bus. It is defined as the time required to complete one basic bus operation like fetching data from memory or writing to an I/O port. This matches with **I**.

Step 2: Match B (T-state)

T-states are the individual clock pulses that make up a machine cycle. For instance, in an 8085, a memory read machine cycle typically takes 3 T-states. It is the fundamental "subdivision" of the machine cycle. This matches with **II**.

Step 3: Match C (Instruction cycle)

The instruction cycle is the top-level time unit. It consists of multiple machine cycles (e.g., Opcode Fetch followed by Memory Read). It encompasses the entire process of getting and finishing an instruction. This matches with **III**.

Step 4: Match D (Opcode fetch)

Opcode fetch is the very first machine cycle of every instruction cycle. Its specific task is to provide the address to memory and retrieve the instruction's operation code (opcode). This matches with **IV**.

Quick Tip: Hierarchy: Instruction Cycle → Machine Cycles → T-states.

Opcode fetch is usually the longest machine cycle (4-6 T-states) because it includes decoding time.

---

72. Match List - I with List - II.

	List - I		List - II
A.	Binary search	I.	Queue
B.	Merge sort	II.	Stack
C.	Depth first search	III.	$T(n) = 2T(n/2) + n$
D.	Breadth first search	IV.	$T(n) = T(n/2) + 1$

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Divide and Conquer algorithms are often described by recurrence relations that show how the problem is split and combined.
- Graph traversal algorithms are distinguished by the data structures used to track the "frontier" of exploration.

Step 1: Match A (Binary search)

Binary search works by splitting a sorted array in half and discarding one half in each step. The time complexity is defined by the recurrence $T(n) = T(n/2) + 1$, leading to $O(\log n)$. This matches with **IV**.

Step 2: Match B (Merge sort)

Merge sort splits the array into two halves, recursively sorts them, and then merges them in linear time. Its recurrence is $T(n) = 2T(n/2) + n$, leading to $O(n \log n)$. This matches with **III**.

Step 3: Match C (Depth first search)

DFS explores as far as possible along each branch before backtracking. This "Last-In, First-Out" behavior is implemented using a **Stack** (either explicitly or via the recursion stack). This matches with **II**.

Step 4: Match D (Breadth first search)

BFS explores all neighbors at the present depth level before moving on to nodes at the next

depth level. This "First-In, First-Out" behavior is implemented using a **Queue**. This matches with **I**.

Quick Tip: DFS = Deep = Stack.
BFS = Broad = Queue.
Binary Search = 1 subproblem ($T(n/2)$).
Merge Sort = 2 subproblems ($2T(n/2)$).

---

73. Match List - I with List - II.

	List - I		List - II
A.	Injective function	I.	$f(x)=x^2$ on $\mathbb{R}$
B.	Surjective function	II.	$f(x)=2x+3$ on $\mathbb{R}$
C.	Bijective function	III.	$f(x)=x^3$ on $\mathbb{R}$
D.	non-injective and non-surjective	IV.	Every element of a codomain has a preimage

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- **Injective (One-to-One):** Distinct elements of the domain map to distinct elements of the codomain.
- **Surjective (Onto):** Every element in the codomain has at least one preimage in the domain.
- **Bijective:** Both injective and surjective.

Step 1: Analyze $f(x) = x^2$ on $\mathbb{R}$ (I)

- Not injective: $f(1) = 1$ and $f(-1) = 1$. Two inputs give same output.

- Not surjective: Negative numbers (e.g., -5) in $\mathbb{R}$ have no real preimage because squares are always non-negative.

So, **D-I**.

Step 2: Analyze surjective definition (IV)

By definition, a function is surjective (onto) if every element in the codomain has a corresponding preimage in the domain.

So, **B-IV**.

Step 3: Analyze f(x) = 2x + 3 on $\mathbb{R}$ (II)

- Injective: $2x_1 + 3 = 2x_2 + 3 \implies x_1 = x_2$.

- Surjective: For any y, $x = (y-3)/2$ exists in $\mathbb{R}$.

Since it is both, it is bijective. In the context of the options, let's look at III first.

Step 4: Analyze f(x) = x^3 on $\mathbb{R}$ (III)

- Injective: Cube roots are unique.

- Surjective: Range is $(-\infty, \infty)$.

This is also bijective. Looking at Option (C), it pairs A with II and C with III. Since both II and III are technically bijective, they are also injective. This mapping holds.

Quick Tip: To check surjectivity on $\mathbb{R}$, check if the range of the function is $(-\infty, \infty)$.

To check injectivity, see if horizontal lines cross the graph more than once.

---

74. Match List - I with List - II.

	List - I		List - II
A.	Structured data	I.	tweet, text documents, videos
B.	Unstructured data	II.	sensor reading, temperature logs
C.	Semi-structured data	III.	JSON, XML, web log files
D.	Time-series data	IV.	Tabular data with defined schema

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- Data Science classifies data based on its internal organization and the presence of metadata or schemas.

Step 1: Match A (Structured data)

Structured data follows a strict, predefined model and is typically organized into rows and columns in a relational database. This is "Tabular data with defined schema." This matches with **IV**.

Step 2: Match B (Unstructured data)

Unstructured data has no predefined internal structure. It includes media files and free-form text where information is not easily mapped to tables. Examples: "tweet, text documents, videos." This matches with **I**.

Step 3: Match C (Semi-structured data)

Semi-structured data contains tags or markers to separate semantic elements but does not follow a strict tabular schema. Examples: "JSON, XML, web log files." This matches with **III**.

Step 4: Match D (Time-series data)

Time-series data is a sequence of data points indexed in time order, capturing how a variable changes. Examples: "sensor reading, temperature logs." This matches with **II**.

Quick Tip: Structured = SQL Tables.

Semi-structured = NoSQL / Nested tags.

Unstructured = "Natural" data like speech or video.

Time-series = Data with a timestamp.

---

75. Match List - I with List - II.

	List - I		List - II
A.	typedef in C	I.	modifies the memory allocated to a variable
B.	enum in C	II.	stores memory address
C.	type casting	III.	is similar to struct
D.	pointer	IV.	redefines the name of existing data types

Choose the correct answer from the options given below :

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

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

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

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

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

Solution:

Concept:

- C language provides keywords and operators to manage how data types are defined, interpreted, and accessed in memory.

Step 1: Match A (typedef in C)

The `typedef` keyword is used to create an alias or a synonym for an existing data type. It "redefines the name of existing data types." This matches with **IV**.

Step 2: Match D (pointer)

A pointer is a variable whose value is the address of another variable. It "stores memory address." This matches with **II**.

Step 3: Match B (enum in C)

An `enum` is a user-defined type, similar to a `struct` or `union`, used to assign names to integral constants. In the context of categorization as a custom data structure, it "is similar to struct." This matches with **III**.

Step 4: Match C (type casting)

Type casting forces a variable of one data type to be treated as another. While "modifies memory allocated" is a slightly ambiguous phrase, it is the remaining pair. In dynamic memory allocation (e.g., `malloc`), type casting the returned `void*` pointer is essential to allocate and manage the intended memory block size. This matches with **I**.

Quick Tip: Typedef = Alias name.

Pointer = Address.

Enum = Named constants.

Type casting = Interpreting data as a different type.

