MAH MCA CET Computer Concepts

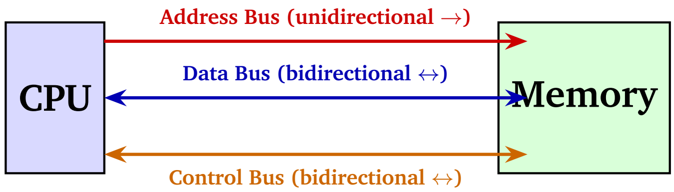
Sample Paper – 4

Duration: 23 Minutes **Maximum Marks: 50**

Instructions

- This paper contains **25** Multiple Choice Questions (Single Correct Answer), modelled on the Computer Concepts section of **MAH MCA CET**.
- Each correct answer carries **+2 marks**. There is **no negative marking** for incorrect or unattempted answers.
- Only **one** option is correct per question.
- Use of mobile phones, calculators, or electronic gadgets is strictly prohibited. Rough work may be done in the space provided in the booklet.

Q1. The diagram below shows the three buses that connect the CPU to memory in a typical computer system. Which statement correctly describes the **address bus**?



(A) The address bus is **unidirectional**, carrying memory addresses from the CPU to memory

(B) The data bus carries memory addresses; the address bus carries actual data between CPU and memory

(C) All three buses — address, data, and control — are bidirectional

(D) The control bus determines the physical width of the address and data buses

Q2. Which of the following **input devices** is primarily used in gaming and aircraft simulation, and measures **angular tilt or displacement**?

(A) Mouse — detects relative surface movement via an optical or laser sensor

(B) Trackball — a stationary ball rotated by the hand to move the cursor

(C) Light pen — a pointing device that senses light emitted by the display screen

(D) Joystick — measures angular tilt/displacement; used primarily in gaming and aircraft flight simulators

Q3. Which of the following correctly describes the structure and a key characteristic of **Dynamic RAM (DRAM)** cells?

(A) SRAM cells use capacitors that require a periodic refresh signal every few milliseconds to retain data

(B) DRAM cells use multiple transistors (6 per bit) arranged as a flip-flop and never require any refresh

(C) DRAM cells use **one capacitor and one transistor** per bit; the capacitor leaks charge, requiring periodic **refresh** every few milliseconds

(D) DRAM is inherently faster than SRAM because it does not lose charge and never needs to be refreshed

Q4. In a multi-platter **hard disk drive**, each platter surface has concentric tracks divided into sectors. What is a **cylinder**?

(A) A cylinder is the set of all sectors residing on a single platter surface

(B) A cylinder is the set of **same-numbered tracks** across all platters, forming a vertical column through the disk stack

(C) A track is divided into cylinders, and each cylinder is further subdivided into sectors

(D) A cylinder is another name for a disk partition created by the operating system

Q5. A special type of memory is accessed by specifying the **data value** to be retrieved, rather than supplying a memory address. What is this memory called?

(A) **Associative memory** (Content-Addressable Memory / CAM) — the entire memory is searched simultaneously for a matching data value

(B) Cache memory — stores recently accessed data to speed up future accesses, indexed by memory address

(C) Virtual memory — extends physical RAM using disk space, addressed by logical addresses

(D) Flash memory — a type of non-volatile EEPROM accessed using physical block addresses, not content

Q6. In a multitasking OS, **context switching** occurs when the CPU moves from one process to another. Which statement best describes what happens?

(A) The currently running process is immediately terminated and removed from memory before the next process starts

(B) Only the program counter (PC) is saved; all other CPU registers remain unchanged during a context switch

(C) Context switching occurs only when a higher-priority process is *newly created*, not when it becomes ready to run

(D) The OS **saves** the current process state (registers, PC, etc.) in its PCB and **restores** the next process state from its PCB — this overhead reduces CPU efficiency

Q7. The table below shows the digit-by-digit conversion of $A3_{16}$ to binary. What is the **complete binary equivalent** of $A3_{16}$?

Hex Digit	4-bit Binary
A_{16}	1010_2
3_{16}	0011_2
$A3_{16}$	**?**

(A) 01011100_2

(B) 10110010_2

(C) 10100011_2

(D) 11000011_2

Q8. The **2-variable K-map** below represents Boolean function $F(A, B)$. The two cells containing 1 are placed *diagonally*. Which expression is the **minimal** form of F?

F	$B{=}0$	$B{=}1$
$A{=}0$	**0**	**1**
$A{=}1$	**1**	**0**

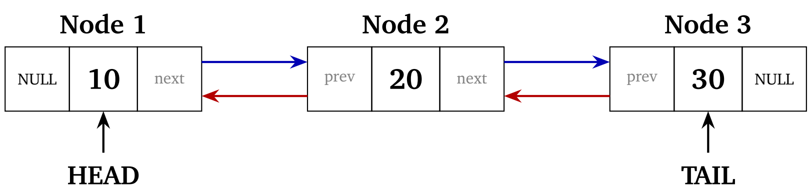
(A) $A + B$

(B) $AB' + A'B$ (the 1-cells are diagonal and *cannot* be grouped; no further simplification is possible)

(C) AB

(D) $A'B'$

Q9. The diagram below shows a **doubly linked list** with three nodes. Which statement correctly describes the structure of each node?



(A) Each node has a **prev** pointer (to the previous node) and a **next** pointer (to the next node), enabling traversal in both directions

(B) Each node has only a **next** pointer; backward traversal requires reversing the entire list first

(C) A doubly linked list uses two separate singly linked lists — one for forward and one for backward traversal

(D) The **prev** pointer always points to the head and the **next** pointer always points to the tail of the list

Q10. A **deque** (double-ended queue) is a linear data structure. Which statement correctly describes its operations?

(A) A deque allows insertion only at the rear and deletion only at the front, just like a standard FIFO queue

(B) A deque supports only LIFO (stack) operations and cannot support FIFO (queue) operations

(C) A deque requires all elements to be sorted by priority before any insertion or deletion can occur

(D) A deque allows **insertion and deletion at both the front and rear** ends, supporting both LIFO and FIFO operations

Q11. Which of the following correctly states the **port numbers** used by the **File Transfer Protocol (FTP)**?

(A) FTP uses port 80 for control connections and port 443 for data transfer

(B) FTP uses port 23 for control and port 25 for data transfer

(C) FTP uses **port 21** for the control connection and **port 20** for data transfer

(D) FTP uses a single port (port 21) for both control commands and data transfer

Q12. A host is assigned the IPv4 address $192.168.10.5/24$. How many **usable host addresses** does a $/24$ subnet provide?

(A) A $/24$ mask reserves 24 bits for host addresses, giving $16{,}777{,}214$ usable hosts per subnet

(B) A $/24$ mask reserves the first 24 bits for the network, leaving 8 bits for hosts: $2^8 - 2 =$ **254** usable host addresses

(C) A $/24$ mask divides the network into exactly 24 equal subnets with equal host capacity

(D) A $/24$ mask reserves only 8 bits for the network and 24 bits for host addresses

Q13. Given the table `Students(id, name, marks)`, which SQL query correctly retrieves the **name** of the student with the **highest marks**?

(A) `SELECT name FROM Students WHERE marks = (SELECT MAX(marks) FROM Students);`

(B) `SELECT name FROM Students WHERE marks = MAX(marks);`

(C) `SELECT MAX(marks), name FROM Students GROUP BY name;`

(D) `SELECT name FROM Students HAVING marks = MAX(marks);`

Q14. Which of the following correctly defines **Boyce-Codd Normal Form (BCNF)**?

(A) BCNF requires that every non-prime attribute is fully functionally dependent on the primary key — identical to the 3NF requirement

(B) BCNF is achieved by eliminating all multi-valued dependencies from a relation schema

(C) BCNF allows functional dependencies where the determinant is a candidate key but *not* a superkey of the relation

(D) For every non-trivial functional dependency $X \to Y$ in a relation, X must be a **superkey** of that relation

Q15. Database transactions must satisfy the **ACID** properties to ensure reliability. What does the acronym **ACID** stand for?

(A) Accuracy, Concurrency, Isolation, and Dependability

(B) Atomicity, Compression, Integrity, and Durability

(C) **Atomicity**, **Consistency**, **Isolation**, and **Durability** — ensuring all-or-nothing, correct, isolated, and permanent transactions

(D) Availability, Consistency, Isolation, and Dependability

Q16. In the C programming language, which statement correctly describes a **pointer** and the **dereference** operator?

(A) A pointer stores the actual value of a variable; it is declared using the `&` operator

(B) A pointer stores the **memory address** of another variable; the `*` operator dereferences it to access the value at that address

(C) Pointers in C can only point to `int` variables; pointing to other data types requires an explicit type cast

(D) The `%` operator is used to dereference a pointer to obtain the value stored at the pointed-to address

Q17. Which statement correctly compares **linear search** and **binary search** in terms of prerequisites and time complexity?

(A) Linear search works on **unsorted** arrays with $O(n)$ time complexity; binary search requires a **sorted** array and runs in $O(\log n)$ time

(B) Binary search can be applied to any unsorted array; a pre-sorted array is *not* required

(C) Linear search has $O(\log n)$ complexity because it skips elements that do not match the search key

(D) Both linear search and binary search have the same $O(n)$ worst-case time complexity

Q18. A software analyst uses the structured table below to model access-control logic for a web application. What is this type of representation called?

Condition / Action	Rule 1	Rule 2	Rule 3
C1: Logged in?	Y	Y	N
C2: Has permission?	Y	N	—
A1: Allow access	X		
A2: Deny access		X	X

(A) A Gantt chart

(B) A flowchart

(C) A **decision table** — a tabular representation of conditions and corresponding actions used to model complex business logic

(D) A state transition diagram

Q19. In operating systems, a **semaphore** controls access to shared resources. Which statement correctly distinguishes a **binary semaphore** from a **counting semaphore**?

(A) Semaphores can be used only in single-threaded applications to protect global variables

(B) A binary semaphore allows multiple processes to access a shared resource simultaneously without restriction

(C) A counting semaphore has only two states (0 and 1) and is always used as a mutex lock for critical sections

(D) A binary semaphore (values 0 or 1) acts as a **mutex**; a counting semaphore allows up to N concurrent accesses to a resource pool of size N

Q20. Which of the following correctly describes **RAID 1** (disk mirroring)?

(A) RAID 1 stripes data across multiple disks in blocks to maximise read/write throughput without any redundancy

(B) RAID 1 **mirrors** data on two disks; if one disk fails, the other retains a complete copy — providing fault tolerance but no write-performance improvement

(C) RAID 1 uses distributed parity bits across disks, allowing single-disk failure recovery without storing a full data copy

(D) RAID 1 requires at least four disks and combines both striping and mirroring for maximum performance and redundancy

Q21. Which statement correctly describes the **operating frequencies** and effective range of **Bluetooth** and **NFC** (Near Field Communication)?

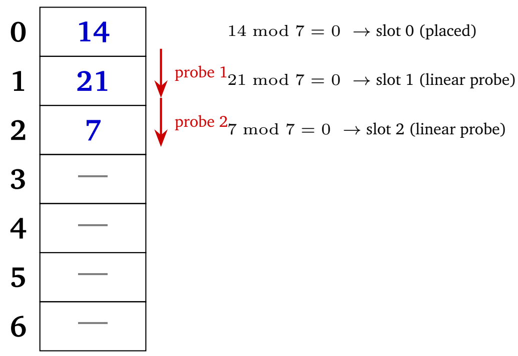
(A) Bluetooth operates in the **2.4 GHz** ISM band; NFC operates at **13.56 MHz** and functions within approximately 4 cm

(B) Bluetooth operates at 5 GHz; NFC operates at 2.4 GHz for short-range communication up to 1 metre

(C) Both Bluetooth and NFC operate in the same 2.4 GHz ISM band, differing only in communication range

(D) NFC operates at 2.4 GHz with a range up to 10 metres; Bluetooth operates at 13.56 MHz

Q22. The diagram below shows a hash table of size 7 after inserting keys 14, 21, and 7 using $h(k) = k \bmod 7$. All three keys hash to slot 0, causing successive collisions. What **collision resolution** strategy is illustrated?



(A) Separate chaining — each slot holds a linked list of all keys that hash to it

(B) Double hashing — a second hash function determines the step size for probing

(C) **Linear probing** — when a collision occurs, the algorithm searches *sequentially* ($+1$ each step) until an empty slot is found

(D) Quadratic probing — the probe sequence uses squared offsets $(+1^2, +2^2, \ldots)$

Q23. In **IEEE 754 single-precision** floating-point representation, how are the **32 bits** allocated across the three fields?

(A) 1 sign bit $+$ 7 exponent bits $+$ 24 mantissa bits $= 32$ bits

(B) 2 sign bits $+$ 8 exponent bits $+$ 22 mantissa bits $= 32$ bits

(C) 1 sign bit + 11 exponent bits + 52 mantissa bits = 64 bits (this is *double* precision)

(D) **1 sign bit** + **8 exponent bits** + **23 mantissa bits** = 32 bits total

Q24. Which of the following correctly describes the **Internet of Things (IoT)**?

(A) IoT refers exclusively to industrial robots on private manufacturing networks that do not connect to the public internet

(B) IoT refers to physical devices with embedded **sensors and actuators** connected to the internet to collect and exchange data; examples include smart thermostats and wearables

(C) IoT is a programming paradigm in which software objects communicate directly with hardware through object-oriented interfaces

(D) IoT refers to the set of TCP/IP protocols that enable computers to communicate over the internet

Q25. Which of the following correctly describes **proprietary software**?

(A) Proprietary software has freely available source code that anyone can modify and redistribute without restriction

(B) Proprietary software is identical to open-source software but requires payment for any commercial use

(C) Proprietary software has **closed source code**; users cannot modify or redistribute it and must use it under a license agreement; examples include Microsoft Windows and Adobe Photoshop

(D) Proprietary software can be obtained for free but restricts the total number of concurrent users who may run it simultaneously

Detailed Solutions

Q1.

Solution

Concept — System Bus Architecture: A system bus has three sub-buses: address, data, and control. Each has a distinct direction and purpose.

Step 1 — Address Bus: The CPU generates a memory address and sends it to memory (or I/O) to specify the location to read from or write to. Information flows only *from* the CPU, so the address bus is **unidirectional** (CPU $\rightarrow$ Memory).

Step 2 — Data and Control Buses: The data bus is **bidirectional**: it carries data to memory on a write and back to the CPU on a read. The control bus is also bidirectional: it carries read/write enable, interrupt, and clock signals in both directions.

Why other options are wrong:

- Option B: Incorrect — the address bus carries addresses, not data.
- Option C: Incorrect — the address bus is strictly unidirectional, not bidirectional.
- Option D: Incorrect — bus width is a design parameter; the control bus carries control signals.

Final Answer: The address bus is unidirectional (CPU $\rightarrow$ Memory) $\Rightarrow$ $\boxed{\text{A}}$

Answer: (A) Go Back to Q 1

Q2.

Solution

Concept — Input Devices: Input devices translate human actions or physical signals into digital data for the computer.

Step 1 — Joystick operation: A joystick contains potentiometers or Hall-effect sensors that measure the **angular tilt or displacement** of a stick relative to its base. The degree of tilt and its direction are reported as coordinate values (X, Y, and sometimes Z axis).

Step 2 — Primary use: Joysticks are the standard input device for flight simulators, arcade-style games, and RC/UAV control because angular displacement maps naturally to directional control.

Why other options are wrong:

- Option A: A mouse detects relative surface movement via optical/laser sensing — not tilt.
- Option B: A trackball is a stationary ball rotated by the hand; it does not measure angular tilt.
- Option C: A light pen detects the light emitted by specific pixels on a CRT screen; not used in simulation.

Final Answer: The joystick measures angular tilt/displacement; used in gaming and simulation $\Rightarrow \boxed{\text{D}}$

Answer: (D) Go Back to Q 2

Q3.

Solution

Concept — DRAM vs. SRAM: The two main RAM technologies differ in cell structure, speed, density, and refresh requirements.

Step 1 — DRAM cell structure: Each DRAM cell stores one bit using **one capacitor** (holds charge = 1, no charge = 0) and **one transistor** (acts as a switch for read/write access). This 1T-1C design gives DRAM very high density.

Step 2 — Refresh requirement: Capacitors are leaky — they lose charge over time (typically within 64 ms). The memory controller must **periodically refresh** all rows (read and rewrite the charge) to prevent data loss.

Why other options are wrong:

- Option A: SRAM uses 6 transistors per bit (a flip-flop) — no capacitor, no refresh needed.
- Option B: DRAM uses only 1 transistor, not multiple; and it definitely requires refresh.
- Option D: DRAM is actually *slower* than SRAM and always requires refresh.

Final Answer: DRAM: 1 capacitor + 1 transistor per bit; capacitor leaks $\rightarrow$ periodic refresh needed $\Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 3

Q4.

Solution

Concept — Hard Disk Geometry: A hard disk consists of one or more platters, each with two magnetic surfaces. Data is organised into tracks, sectors, and cylinders.

Step 1 — Track and sector: Each platter surface has concentric circular **tracks**. Each track is subdivided into fixed-size **sectors** (typically 512 B or 4 KB), which are the smallest addressable unit.

Step 2 — Cylinder: When all platters are stacked, the tracks at the **same radial position** on every surface form a vertical column called a **cylinder**. Reading an entire cylinder requires no head movement, reducing seek time.

Why other options are wrong:

- Option A: All sectors on one surface form a platter face, not a cylinder.
- Option C: Tracks are divided into sectors, not cylinders.
- Option D: Disk partitions are OS-level concepts, unrelated to physical cylinder geometry.

Final Answer: A cylinder = same-numbered tracks across all platters $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 4

Q5.

Solution

Concept — Content-Addressable Memory (CAM): Unlike conventional RAM (accessed by address), CAM is accessed by **content** — the hardware searches all locations simultaneously for a matching value.

Step 1 — CAM operation: A search key (tag) is broadcast to all cells in parallel. Each cell compares the key with its stored value. Cells that match assert a *match line*; the hardware returns the matching data or its address.

Step 2 — Typical applications: CAM is used in CPU TLBs (Translation Lookaside Buffers), network routing tables, and cache tag arrays — wherever fast associative lookup is critical.

Why other options are wrong:

- Option B: Cache memory is accessed using a physical or virtual address, not by content.

- Option C: Virtual memory extends RAM using disk space; it is address-mapped, not content-searched.
- Option D: Flash memory (EEPROM) is a storage technology accessed by block address.

Final Answer: Associative memory (CAM) — searched by data content, not by address ⇒ $\boxed{\text{A}}$

Answer: (A) Go Back to Q 5

Q6.

Solution

Concept — Context Switching: When the OS decides to switch the CPU from one process to another (due to a timer interrupt, I/O wait, or higher-priority process becoming ready), it performs a context switch.

Step 1 — Save current context: The OS saves the running process's CPU state — including all general-purpose registers, program counter (PC), stack pointer, and process status — into the process's **Process Control Block (PCB)**.

Step 2 — Restore new context and overhead: The OS loads the PCB of the next scheduled process, restoring its saved CPU state. This save-and-restore operation takes non-trivial CPU time (plus cache/TLB flush costs), reducing the effective CPU time available for user processes.

Why other options are wrong:

- Option A: The switched-out process is *not* terminated; it becomes Ready/Waiting and resumes later.
- Option B: Saving only the PC is insufficient — all registers must be saved to resume correctly.
- Option C: Context switches also occur on I/O blocks, timer expiry, and voluntary yields — not just at process creation.

Final Answer: Context switch saves current PCB and restores next process's PCB — adds overhead ⇒ $\boxed{\text{D}}$

Answer: (D) Go Back to Q 6

Q7.

Solution

Concept — Hexadecimal to Binary Conversion: Each hexadecimal digit maps directly to a unique 4-bit binary group, enabling digit-by-digit conversion.

Step 1 — Convert each hex digit separately:

$$\mathrm{A}_{16} = 10_{10} = 1010_2 \qquad 3_{16} = 3_{10} = 0011_2$$

Step 2 — Concatenate the groups:

$$\mathrm{A}3_{16} = \underbrace{1010}_{A}\,\underbrace{0011}_{3} = 10100011_2$$

The most-significant hex digit (A) becomes the most-significant 4 bits.

Why other options are wrong:

- Option A (01011100_2): equals $5\mathrm{C}_{16}$, not $\mathrm{A}3_{16}$.
- Option B (10110010_2): equals $\mathrm{B}2_{16}$, not $\mathrm{A}3_{16}$.
- Option D (11000011_2): equals $\mathrm{C}3_{16}$, not $\mathrm{A}3_{16}$.

Final Answer: $\mathrm{A}3_{16} = 1010\,0011_2 = 10100011_2 \Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 7

Q8.

Solution

Concept — Karnaugh Map Minimisation: K-map minimisation groups adjacent 1-cells to eliminate variables. Diagonal 1-cells (non-adjacent) *cannot* be grouped.

Step 1 — Read the K-map:

$$F(A,B): \quad F(0,0)=0,\ F(0,1)=1,\ F(1,0)=1,\ F(1,1)=0$$

The 1s occur at $(A=0, B=1)$ and $(A=1, B=0)$ — diagonal positions in the map.

Step 2 — Attempt grouping: In a 2-variable K-map, adjacency requires sharing an edge. The diagonal cells share only a corner — they are **not adjacent** and cannot be combined. The minimal SOP is therefore:

$$F = A'B + AB' \quad \text{(XOR function — irreducible)}$$

Why other options are wrong:

- Option A ($A + B$): This is OR — true when at least one is 1; gives $F(1, 1) = 1 \neq 0$.
- Option C (AB): True only when both are 1; gives $F(1, 1) = 1 \neq 0$ and $F(0, 1) = 0 \neq 1$.
- Option D ($A'B'$): True only when both are 0; gives $F(0, 0) = 1 \neq 0$.

Final Answer: $F = AB' + A'B$ (XOR; the diagonal 1-cells cannot be grouped) $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 8

Q9.

Solution

Concept — Doubly Linked List: A doubly linked list extends the singly linked list by adding a backward pointer to each node, enabling efficient bidirectional traversal.

Step 1 — Node structure: Each node contains three fields:

- **prev** — pointer to the previous node (NULL at the head)
- **data** — the stored value
- **next** — pointer to the next node (NULL at the tail)

Step 2 — Traversal: Forward traversal follows `next` pointers from HEAD to TAIL. Backward traversal follows `prev` pointers from TAIL to HEAD. No reversal is needed.

Why other options are wrong:

- Option B: Having only a `next` pointer describes a *singly* linked list, not a doubly linked list.
- Option C: A doubly linked list is *one* list with two pointers per node, not two separate lists.
- Option D: `prev` points to the *preceding node*, not always to the head.

Final Answer: Each node has `prev` (to previous node) and `next` (to next node) — bidirectional $\Rightarrow \boxed{\text{A}}$

Answer: (A) Go Back to Q 9

Q10.

Solution

Concept — Deque (Double-Ended Queue): A deque generalises both stacks and queues by allowing insertion and deletion at *both* ends.

Step 1 — Operations supported:

- `insertFront`, `insertRear` — insert at either end
- `deleteFront`, `deleteRear` — delete from either end

Step 2 — Generalisation: Using only rear-insert and front-delete gives FIFO (queue). Using only one end for both operations gives LIFO (stack). The deque supports all four combinations.

Why other options are wrong:

- Option A: Rear-insert + front-delete only describes a standard FIFO queue, not a deque.
- Option B: A stack restricts both insert and delete to one end; a deque is not so restricted.
- Option C: Priority queues require ordering by priority; a deque has no priority constraint.

Final Answer: Deque = insert/delete at both front and rear; supports LIFO and FIFO $\Rightarrow \boxed{\text{D}}$

Answer: (D) Go Back to Q 10

Q11.

Solution

Concept — FTP Port Numbers: FTP uses a dual-channel model with separate TCP connections for commands and data.

Step 1 — Control connection (port 21): The client establishes a persistent TCP connection to port 21 on the server. All FTP commands (USER, PASS, LIST, RETR, STOR, ...) and server responses travel over this channel.

Step 2 — Data connection (port 20, active mode): For each file transfer or directory listing, a separate TCP data connection is opened. In *active* mode, the server initiates this connection from its port 20 to a client-specified port.

Why other options are wrong:

- Option A: Port 80 = HTTP; port 443 = HTTPS — neither is FTP.
- Option B: Port 23 = Telnet; port 25 = SMTP — neither is FTP.
- Option D: FTP uses *two* separate ports, not one port for everything.

Final Answer: FTP uses port 21 (control) and port 20 (data) $\Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 11

Q12.

Solution

Concept — IPv4 Subnetting: A subnet mask specifies how many bits of a 32-bit IPv4 address are the *network* portion versus the *host* portion.

Step 1 — Bit allocation for /24:

$$\text{Network bits} = 24 \qquad \text{Host bits} = 32 - 24 = 8$$

Step 2 — Usable host count:

$$\text{Total addresses} = 2^8 = 256 \qquad \text{Usable hosts} = 256 - 2 = \mathbf{254}$$

Subtract 2: the all-zeros address (network address) and the all-ones address (broadcast address) are reserved and cannot be assigned to hosts.

Why other options are wrong:

- Option A: $16{,}777{,}214$ usable hosts corresponds to a /8 mask (8 network bits, 24 host bits).
- Option C: A subnet mask does not directly equal the number of subnets; the number of subnets depends on borrowed bits.
- Option D: /24 means 24 *network* bits, not 8 network bits.

Final Answer: $/24 \Rightarrow$ 8 host bits $\Rightarrow 2^8 - 2 = 254$ usable host addresses $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 12

Q13.

Solution

Concept — SQL Subquery with Aggregate Function: A subquery is a SELECT statement nested inside another query. Using `MAX()` in a subquery retrieves the maximum aggregate value, which is then used as a filter in the outer query.

Step 1 — Inner query: `SELECT MAX(marks) FROM Students` returns the single highest marks value in the table.

Step 2 — Outer query: `SELECT name FROM Students WHERE marks = (max_value)` returns the name(s) of students whose marks equal that maximum.

Why other options are wrong:

- Option B: `MAX()` is an aggregate function and *cannot* be used directly in a `WHERE` clause without a subquery; this would cause a SQL error.
- Option C: `GROUP BY name` computes each student's maximum separately — meaningless for a single-row table and does not isolate the overall top scorer.
- Option D: `HAVING` is used with `GROUP BY` to filter groups; it cannot be used standalone without a `GROUP BY` clause and an aggregate.

Final Answer: Use a correlated subquery with `MAX(marks)` in the `WHERE` clause $\Rightarrow$ $\boxed{\text{A}}$

Answer: (A) Go Back to Q 13

Q14.

Solution

Concept — Boyce-Codd Normal Form (BCNF): BCNF is a stricter version of Third Normal Form (3NF). It eliminates all functional dependency anomalies by requiring every determinant to be a superkey.

Step 1 — Formal definition: A relation R is in BCNF if, for every non-trivial functional dependency $X \to Y$ in R, X is a **superkey** of R (i.e., X functionally determines all attributes of R).

Step 2 — Difference from 3NF: 3NF additionally allows $X \to Y$ if Y is a prime attribute (part of a candidate key), even when X is not a superkey. BCNF removes this exception — every determinant *must* be a superkey, making BCNF strictly stronger than 3NF.

Why other options are wrong:

- Option A: That describes 3NF's full functional dependency requirement — not uniquely BCNF.
- Option B: Eliminating multi-valued dependencies is the requirement for 4NF, not BCNF.
- Option C: Allowing a non-superkey candidate key as a determinant is exactly what 3NF (not BCNF) permits.

Final Answer: BCNF: for every $X \to Y$, X must be a superkey $\Rightarrow \boxed{\text{D}}$

Answer: (D) Go Back to Q 14

Q15.

Solution

Concept — ACID Transaction Properties: These four properties guarantee that database transactions are processed reliably.

Step 1 — Definitions:

- **Atomicity:** A transaction is all-or-nothing. If any step fails, all changes are rolled back.
- **Consistency:** A transaction brings the database from one valid state to another, preserving all integrity constraints.
- **Isolation:** Concurrent transactions execute as if they were serial — intermediate states are not visible to other transactions.
- **Durability:** Once a transaction commits, its changes are permanent, surviving crashes and power failures (ensured by write-ahead logging and checkpoints).

Why other options are wrong:

- Option A: "Accuracy" and "Dependability" are not standard ACID terms.
- Option B: "Compression" replaces Consistency — incorrect.
- Option D: "Availability" is a CAP theorem term, not part of ACID.

Final Answer: ACID $=$ Atomicity, Consistency, Isolation, Durability $\Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 15

Q16.

Solution

Concept — Pointers in C: A pointer is a variable whose value is the **memory address** of another variable.

Step 1 — Declaration and assignment:

- `int x = 42;` — integer variable
- `int *ptr;` — pointer to int, declared with `*`
- `ptr = &x;` — `&` gives the address of `x`; now `ptr` holds that address

Step 2 — Dereferencing: `*ptr` applies the dereference operator `*` to retrieve the value stored at the address held in `ptr`. Here, `*ptr` returns 42.

Why other options are wrong:

- Option A: A pointer stores an *address*, not the variable's value; `&` is the *address-of* operator, not the declaration keyword.
- Option C: C pointers can point to any data type (`int*`, `float*`, `char*`, `void*`, etc.).
- Option D: `%` is the modulo operator; it does not dereference pointers.

Final Answer: A pointer stores a memory address; `*ptr` dereferences it $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 16

Q17.

Solution

Concept — Search Algorithm Comparison: Linear search and binary search differ in their input requirements and performance characteristics.

Step 1 — Linear search: Scans each element sequentially from index 0. Works on any array (sorted or unsorted). Worst case: n comparisons $\Rightarrow O(n)$ time.

Step 2 — Binary search: Requires a **sorted** array. Compares the target with the middle element, then discards half the remaining elements. Each step halves the search space $\Rightarrow O(\log n)$ time.

Why other options are wrong:

- Option B: Binary search *requires* a sorted array; applying it to unsorted data yields incorrect results.

- Option C: Linear search does not skip elements intelligently — it checks each one, giving $O(n)$ not $O(\log n)$.
- Option D: Binary search is $O(\log n)$, which is asymptotically far better than $O(n)$ for large n.

Final Answer: Linear $O(n)$ on unsorted; binary $O(\log n)$ requires sorted array $\Rightarrow$ $\boxed{\text{A}}$

Answer: (A) Go Back to Q 17

Q18.

Solution

Concept — Decision Table: A decision table is a structured document used in requirements engineering and software testing to represent complex combinatorial business logic.

Step 1 — Structure: A decision table has four quadrants: (1) condition stubs (which conditions are evaluated), (2) condition entries (Y/N/— for each rule), (3) action stubs (which actions can be taken), and (4) action entries (X marks the actions to perform for each rule).

Step 2 — Reading the table: Rule 1: logged in & has permission $\rightarrow$ allow access. Rule 2: logged in but no permission $\rightarrow$ deny. Rule 3: not logged in $\rightarrow$ deny regardless of permission. This covers all meaningful combinations exhaustively.

Why other options are wrong:

- Option A: A Gantt chart shows project tasks on a timeline — it has no condition/action structure.
- Option B: A flowchart uses shapes (diamonds, rectangles) to show sequential and branching control flow — not tabular rules.
- Option D: A state transition diagram shows system states and the events that trigger transitions between them.

Final Answer: The table is a **decision table** — conditions mapped to actions per rule $\Rightarrow$ $\boxed{\text{C}}$

Answer: (C) Go Back to Q 18

Q19.

Solution

Concept — Semaphore: A semaphore is an integer synchronisation primitive with two atomic operations: **wait** (P) and **signal** (V).

Step 1 — Binary semaphore: Restricted to values $\{0, 1\}$. Used as a **mutex** (mutual exclusion lock): value 1 means the resource is free; value 0 means it is held. Only one process can be in the critical section at a time.

Step 2 — Counting semaphore: Can take any non-negative integer value up to N. Initialised to N, it allows up to N processes to access the resource pool simultaneously. Common use: controlling access to a pool of N database connections.

Why other options are wrong:

- Option A: Semaphores are designed for *multi-threaded/multi-process* environments, not single-threaded.
- Option B: A binary semaphore allows only *one* process at a time, not multiple without restriction.
- Option C: That describes a binary semaphore — confusing the two types.

Final Answer: Binary semaphore = mutex (0 or 1); counting semaphore allows N concurrent accesses $\Rightarrow \boxed{\text{D}}$

Answer: (D) **Go Back to Q 19**

Q20.

Solution

Concept — RAID Levels: RAID (Redundant Array of Independent Disks) combines multiple drives for performance, redundancy, or both.

Step 1 — RAID 1 (Mirroring): Every write is duplicated to two disks simultaneously. Both disks always hold identical copies of all data.

Step 2 — Properties of RAID 1:

- **Fault tolerance:** If one disk fails, the system continues reading from the other disk — no data loss.
- **No write performance gain:** Every write must complete on both disks; write throughput equals that of a single disk.
- **Read performance:** Can potentially be improved (reads can be distributed), but the primary benefit is redundancy.

Why other options are wrong:

- Option A: Striping without redundancy describes RAID 0.
- Option C: Distributed parity describes RAID 5.
- Option D: Minimum four disks with striping + mirroring describes RAID 10 (RAID 1+0).

Final Answer: RAID 1 = disk mirroring; fault tolerant, no write performance gain $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 20

Q21.

Solution

Concept — Bluetooth and NFC Frequencies: Both are short-range wireless technologies but operate at very different frequencies and ranges.

Step 1 — Bluetooth: Operates in the **2.4 GHz** ISM (Industrial, Scientific, Medical) band (2.400–2.4835 GHz). Uses frequency-hopping spread spectrum (FHSS) with 79 channels. Typical range: 10 m (Class 2) to 100 m (Class 1).

Step 2 — NFC (Near Field Communication): Operates at **13.56 MHz** — a much lower frequency. Based on RFID technology. Effective operating range: up to approximately **4 cm**. Used for contactless payments, transit cards, and device pairing.

Why other options are wrong:

- Option B: 5 GHz is used by IEEE 802.11a/ac Wi-Fi — not Bluetooth.
- Option C: Bluetooth and NFC operate at different frequencies (2.4 GHz vs. 13.56 MHz).
- Option D: The frequencies are swapped; NFC does not have a 10-metre range.

Final Answer: Bluetooth = 2.4 GHz ISM; NFC = 13.56 MHz, range $\approx$ 4 cm $\Rightarrow \boxed{\text{A}}$

Answer: (A) Go Back to Q 21

Q22.

Solution

Concept — Hash Table Collision Resolution: When two keys hash to the same slot, the collision must be resolved. Linear probing is an open-addressing strategy.

Step 1 — Hash function and insertions:

$$h(k) = k \bmod 7$$

- $h(14) = 14 \bmod 7 = 0$ — slot 0 empty $\to$ place 14 at slot 0.
- $h(21) = 21 \bmod 7 = 0$ — slot 0 occupied (collision) $\to$ probe slot 1 (empty) $\to$ place 21 at slot 1.
- $h(7) = 7 \bmod 7 = 0$ — slot 0 occupied $\to$ probe slot 1 (occupied) $\to$ probe slot 2 (empty) $\to$ place 7 at slot 2.

Step 2 — Linear probing rule: On collision at slot i, probe slots $(i+1) \bmod m$, $(i+2) \bmod m$, ... until an empty slot is found. The probe step is always $+1$, giving a *linear* sequence.

Why other options are wrong:

- Option A: Separate chaining stores colliding keys in linked lists attached to each slot — the array stays flat in open addressing.
- Option B: Double hashing uses a second hash function $h_2(k)$ to compute variable step sizes.
- Option D: Quadratic probing uses steps $1^2, 2^2, 3^2, \ldots$ — not the unit step shown.

Final Answer: Unit-step sequential probing on collision = **linear probing** $\Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 22

Q23.

Solution

Concept — IEEE 754 Single-Precision Format: The IEEE 754 standard defines a 32-bit binary representation for floating-point numbers with three distinct fields.

Step 1 — Field breakdown:

$$\underbrace{\text{1 bit}}_{\text{Sign}} \mid \underbrace{\text{8 bits}}_{\text{Exponent (biased-127)}} \mid \underbrace{\text{23 bits}}_{\text{Mantissa (fraction)}}$$

Total: $1 + 8 + 23 = 32$ bits.

Step 2 — Field interpretation:

- **Sign (1 bit):** 0 = positive, 1 = negative.
- **Exponent (8 bits):** Stored with a bias of 127; actual exponent = stored value − 127.
- **Mantissa (23 bits):** Fractional part of the normalised significand; an implicit leading 1 is assumed ($1.fraction$).

Why other options are wrong:

- Option A: 7 exponent bits would drastically reduce the representable range — not the IEEE 754 standard.
- Option B: Two sign bits make no mathematical sense.
- Option C: 1+11+52=64 bits describes *double-precision* (IEEE 754 binary64), not single-precision.

Final Answer: IEEE 754 single-precision: $1 + 8 + 23 = 32$ bits $\Rightarrow \boxed{\text{D}}$

Answer: (D) Go Back to Q 23

Q24.

Solution

Concept — Internet of Things (IoT): IoT refers to the network of physical objects embedded with sensors, actuators, software, and connectivity to exchange data with other devices and systems over the internet.

Step 1 — Core components:

- **Sensors:** Measure physical quantities (temperature, humidity, motion, heart rate).
- **Actuators:** Perform physical actions based on commands (adjust thermostat, open valve).
- **Connectivity:** Wi-Fi, Bluetooth, Zigbee, LTE, etc., link devices to the internet or gateways.
- **Data processing:** Cloud or edge platforms analyse collected data and trigger responses.

Step 2 — Examples: Smart thermostats (Nest), fitness wearables (Fitbit/Apple Watch), smart meters, connected vehicles, industrial sensors, and agricultural monitoring systems.

Why other options are wrong:

- Option A: IoT is not limited to industrial robots and does include internet connectivity.
- Option C: IoT is a hardware/networking paradigm, not an object-oriented programming model.
- Option D: That describes the TCP/IP internet protocol suite, not IoT.

Final Answer: IoT = physical devices with sensors and actuators connected to the internet $\Rightarrow \boxed{\text{B}}$

Answer: (B) Go Back to Q 24

Q25.

Solution

Concept — Software Licensing Types: Software licensing determines who may use, modify, and distribute software and under what conditions.

Step 1 — Proprietary software characteristics:

- **Closed source:** Source code is kept secret by the developer/vendor.
- **License agreement (EULA):** Users are granted a limited right to *use* the software; they cannot copy, modify, or redistribute it.
- **Examples:** Microsoft Windows, Adobe Photoshop, Microsoft Office, Oracle Database.

Step 2 — Contrast with open-source: Open-source software (GPL, MIT, Apache) makes source code publicly available, permitting modification and redistribution (subject to license conditions). Freeware is free-to-use but may still be proprietary with a closed source.

Why other options are wrong:

- Option A: Freely available, modifiable, redistributable source code defines *open-source* software.
- Option B: Open-source software can also require payment for commercial versions (e.g., Red Hat Enterprise Linux).
- Option D: Per-seat or concurrent-user limits are licensing models that can apply to proprietary software, but do not *define* it.

Final Answer: Proprietary = closed source, licensed, non-modifiable (e.g., Windows, Photoshop) $\Rightarrow \boxed{\text{C}}$

Answer: (C) Go Back to Q 25

Answer Key

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	A	2	D	3	C	4	B	5	A
6	D	7	C	8	B	9	A	10	D
11	C	12	B	13	A	14	D	15	C
16	B	17	A	18	C	19	D	20	B
21	A	22	C	23	D	24	B	25	C