HOLY INFANT HIGH SCHOOL - MELEN
MARKING GUIDE
MOCK AL CSC 2026

PAPER 1

1. **C ($ABC' + AB'C + A'BC$)**. The question requires the output to be 1 only when **exactly two** inputs are 1. Each term in option C represents exactly two 1s: $(A = 1, B = 1, C = 0), (A = 1, B = 0, C = 1), or (A = 0, B = 1, C = 1)$.
   A: $(A + B + C)'$ is a NOR gate, outputting 1 only when all inputs are 0. **B:** ABC is an AND gate, outputting 1 only when all three inputs are 1. **D:** $A + B + C$ is an OR gate, outputting 1 if any input is 1.
2. **A (X, Z)**. Operations:
   Push(X) → [X];
   Push(Y) → [X, Y];
   Pop() → [X];
   Push(Z) → [X, Z];
   Push(W) → [X, Z, W];
   Pop() → [X, Z].
   B: This reverses the order; stacks are LIFO (Last-In, First-Out), but the request asked for bottom to top. **C:** W was popped in the final operation. **D:** This shows all pushed items without accounting for the Pop() operations.
3. **C ((i) and (iv) only)**. The ER diagram shows a **1-to-M** relationship between Lecturer and Course. In a 1-to-M relationship, the primary key of the "**1**" side (NIC) becomes a foreign key in the "**M**" side (Course).
   A/D: Relation (ii) contains redundant CourseId data within the Lecturer table, violating normalization. **B:** Relation (iii) is a junction table, which is only necessary for M-to-M relationships.
4. **D (Patents)**. Patents protect new inventions, devices, or processes, granting the inventor exclusive rights for a limited period.
   A: Copyright protects original creative works like books or software code, not the physical device/invention itself. **B:** Trademarks protect brand names, logos, and slogans. **C:** Registered designs protect the aesthetic appearance or shape of a product.
5. **D (Switch)**. A switch is the standard device used within a LAN to forward data packets specifically to the destination MAC address.
   A: A modem converts digital signals to analog (and vice-versa) for internet access. **B:** A server provides services (like files or web pages) but doesn't handle the raw packet switching for the whole LAN. **C:** A firewall filters traffic based on security rules but doesn't perform basic forwarding.
6. **B (Device A: Switch, Device B: Server, Device C: Router)**. Looking at the diagram, Device A connects the local PCs (Switch), Device B connects the printer/resources (often a Server), and Device C connects the whole network to the Internet (Router).
   A/C/D: These misidentify the gateway to the internet (which must be a Router or Gateway) or the central hub for the local computers.
7. **D (Requirements cannot be easily changed...)**. The Waterfall model is linear and rigid; once a phase (like development) begins, going back to change requirements is very difficult and costly.
   A: Costs are actually easier to estimate in Waterfall due to the fixed scope. **B:** Waterfall is a methodology and does not dictate the type of libraries (open-source or proprietary) used. **C:** Waterfall is generally slower than agile methods, not too quick.
8. **C (P4)**. At time 2, P2 has 3ms left (5 − 2). P4 arrives with a burst of 2ms. Since SRTF always picks the shortest remaining time, P4 (2ms) is shorter than P2 (3ms), so P4 runs at time 3.

0 P2 2	**2 P4 4**	**4 P2 7**	**7 P3 15**	**15 P1 25**

   A/B/D: P1 and P3 have longer remaining times (10 and 8 respectively) than P4 (2).
9. **B (The variable val is updated to 20)**. *ptr is the dereference operator. By assigning *ptr = 20, you are changing the value stored at the address ptr points to, which is the variable val.
   A: No new variable is created; the existing memory for val is used. **C:** ptr still holds the address of val; only the value at that address changed. **D:** This is standard, valid C syntax.
10. **B (Provide reports... to support structured decision-making)**. The primary goal of an MIS is to process data into meaningful reports that help managers make informed, structured decisions.
    A: Automation of manufacturing is typically handled by Process Control Systems or Robotics. **C:** This is the role of facilities management or security systems. **D:** This refers to signal processing or telecommunications.
11. **B (To ensure unique representation and maximize precision)**. Normalization ensures that there is only one way to represent a specific number (by having a non-zero digit after the binary point) and uses the available bits for the mantissa most efficiently.
    A: Normalization affects the mantissa/precision, not the range of the exponent. **C:** Normalization is unrelated to the conversion between complement formats. **D:** Storing decimals as integers is a different technique (like fixed-point or scaling).
12. **D (*)**. In the expression $((A + B) * (C/D))$, the multiplication is the last operation performed because it joins the two parenthesized sub-expressions. Therefore, it is the root of the tree.
    A/B: These are operators for the sub-expressions and would be child nodes of the root. **C:** 'A' is an operand and would be a leaf node.
13. **C (storing data online)**. Storing data online (cloud storage) depends on internet connectivity and the service provider, not the local RAM or HDD/SSD specs.
    A: Regular freezing is often caused by insufficient RAM (8GB) or a failing HDD. **B:** Running old software on new hardware can cause compatibility issues (hardware specifications/drivers). **D:** Having no space is directly related to the 256GB HDD capacity.
14. **A (It focuses on the internal logic...)**. White-box testing requires knowledge of the code's internal structure and paths, whereas black-box testing only focuses on inputs and outputs.
    B: End-user testing is Beta testing, which is a form of black-box testing. **C:** Functional requirements are the focus of black-box testing. **D:** This describes black-box testing.
15. **C (Compiler)**. A compiler translates the entire source code into machine code as a single batch before execution.
    A: An interpreter translates and executes code line-by-line. **B:** An assembler translates assembly language (low-level) into machine code, not high-level languages. **D:** A linker combines object files into a single executable.
16. **B (A blueprint or template for creating objects)**. In OOP, a class defines the properties (attributes) and behaviors (methods) that the objects created from it will have.
    A: A specific instance is called an **Object**. **C:** A function that returns an object is usually a constructor or a factory method, not the class itself. **D:** A group of related variables is a "struct" or data record.
17. **A (When the loop body should potentially never execute...)**. A 'While' loop tests the condition **before** entering the loop, so if the condition is false initially, the body never runs. A 'Repeat-Until' loop tests at the end, so it always runs at least once.
    B: This describes a 'Repeat-Until' loop. **C:** This describes a 'For' loop. **D:** Both loops can handle complex logical operators.
18. **C (7)**. Table size 10, $h(k) = k \backslash pmod 10$:
    Key 25: $25\ mod\ 10 = 5$. Index 5 is filled.
    Key 35: $35\ mod\ 10 = 5$. Index 5 is full; linear probing moves to index 6.

Key 45: $45 \ mod \ 10 \ = \ 5$. Index 5 and 6 are full; linear probing moves to index 7.
A/B/D: These indices are either already occupied or do not follow the linear probing sequence.

19. **C (Data Link Layer)**. The Data Link Layer is responsible for node-to-node data transfer and basic error detection (using frames) on the physical link. **A:** The Network Layer handles routing and logical addressing (IP). **B:** The Transport Layer handles end-to-end communication and reliability. **D:** The Session Layer manages sessions between applications.
20. **C (Analysis Phase)**. The User Requirement Specification (URS) is the primary output of the Analysis phase, where the needs of the users are documented.
A: The Design phase creates technical blueprints based on the URS. **B:** The Implementation phase involves actual coding. **D:** Maintenance happens after the system is live.
21. **B (Save the CPU register state... into its PCB)**. During a context switch, the OS must save the current state (registers, program counter) of the running process into its Process Control Block (PCB) so it can be resumed later.
A: Clearing RAM would destroy all other active processes. **C:** A context switch doesn't necessarily mean a process is terminated; it might just be paused. **D:** This is the job of the Memory Management Unit (MMU), not a specific task unique to context switching.
22. **A ($O(n^2)$)**. The outer loop runs n times. The inner loop runs $n/2$ times. Total operations are $n \cdot (n/2) = (1/2) \cdot n^2$. In Big O notation, constants are dropped, leaving $O(n^2)$.
B/C/D: These represent logarithmic, constant, or linear growth, which do not match a nested loop structure.
23. **B (A join based on all columns with the same name...)**. A Natural Join automatically links tables based on all columns that share the same name and data type, removing duplicate columns in the result.
A: This describes a Cross Join or Union. **C:** Natural joins do not require a WHERE clause; the join condition is implicit. **D:** While usually used with keys, it technically works on any matching column names.
24. **B (send an email with the document attached and a digital signature)**. A digital signature provides **authentication** (trusted source) and **integrity** (has not been changed). Encryption (implied in the strategy) ensures confidentiality.
A/C: Password protection offers some confidentiality but does not verify the source or integrity as strongly as a digital signature. **D:** An electronic signature (like a scanned image) is easily forged and does not provide technical integrity/verification.
25. **D (Gateway)**: A Gateway operates at higher OSI layers to translate between different network architectures and protocols.
A and B work within the same protocol. C connects two LAN segments using the same protocol.
26. **A (Risks are localized to a small part...)**: Pilot changeover involves implementing the system in one branch or department first. If it fails, only that "pilot" group is affected.
B describes "Parallel" changeover. C describes "Direct" changeover. D is false; pilot still requires training.
27. **B (The maximum theoretical RAM capacity increases...)**: A 32-bit bus addresses 2^{32} bytes (4GB). A 64-bit bus addresses 2^{64} bytes (16 Exabytes), allowing the system to use much more RAM.
A describes the data bus, not the address bus. C is irrelevant to transfer rates. D is handled by the FPU, not the address bus.
28. **C (It halves the remaining search space...)**: Binary search uses a "divide and conquer" approach, reducing the work by 50% at each step, making it much faster ($O(\log n)$) than checking every item ($O(n)$).
A is false. B is false; Binary search requires sorted data. D is false; it works on any comparable data.
29. **A (anti-virus software)**: Malicious code refers to viruses, worms, and trojans. Anti-virus software is specifically designed to detect and remove these threats.
B prevents unauthorized access. C protects data privacy. D is for recovering after an incident.
30. **C (i7 dual core at 2.8GHz, 8GB primary memory...)**: Complex research and 3D rendering require a powerful processor (i7) and sufficient RAM (8GB). 1TB storage is better for large media files than 128GB.
A has only 4GB RAM. B has a slow CPU and tiny storage. D has a slow CPU and only 4GB RAM.
31. **A (20, 40, 30, 60, 80, 70, 50)**: Inserting in order: 50 is root; 30 is left, 70 is right; 20/40 are children of 30; 60/80 are children of 70. Post-order (Left, Right, Root) gives this sequence.
B is In-order traversal. C is Pre-order traversal. D is a reverse In-order traversal.
32. **C (Negates at least one of the four Coffman conditions)**: Deadlock requires four conditions: Mutual Exclusion, Hold and Wait, No Pre-emption, and Circular Wait. Breaking any one of these prevents deadlock.
A might help but doesn't guarantee avoidance. B is only for printers/output. D does not solve logic-based resource contention.
33. **C (HAVING)**: The HAVING clause is specifically designed to filter groups created by GROUP BY based on aggregate values like SUM() or COUNT().
A filters individual rows before grouping. B creates the groups. D sorts the final output.
34. **A (Host Identifier)**: An IP address consists of a network part and a host part. The host identifier refers to the specific interface or device on that network.
B identifies the network itself. C defines the split between network and host. D is the router's address for external traffic.
35. **C (Act as a software interface...)**: A Device Driver allows the Operating System to communicate with hardware without needing to know the specific technical details of the hardware.
A is a utility or control panel. B describes a compiler/assembler. D is power management.
36. **B (The limited throughput between the CPU and Main Memory)**: Because the CPU and memory are separate, the speed of the system is limited by how fast data can travel across the bus between them.
A refers to internal component speeds. C is a design limitation but not the "bottleneck". D is secondary storage latency.
37. **A (Relational databases use multiple tables...)**: Relational databases split data into tables to ensure each fact is stored once (reducing redundancy), whereas flat-files store everything in one list.
B is false; flat-files can't handle these. C describes a flat-file. D is reversed; relational needs a DBMS.
38. **A (11110011)**: +13 is 00001101. To get -13 in Two's Complement: Flip bits (11110010) and add 1, resulting in 11110011.
B is the result before adding 1. C is +13. D is Sign-Magnitude representation for -13.
39. **A (RISC uses a small set... CISC uses many complex...)**: RISC (Reduced Instruction Set Computer) focuses on efficiency through simplicity; CISC (Complex) uses varied instructions to perform tasks in fewer lines.
B is false; RISC is common in phones. C is false; RISC needs more registers. D is the opposite of reality.
40. **C ((ii), (iii) only)**: Batch processing handles tasks that can be grouped and run without user interaction (backups and sorting daily orders).
(i) is a real-time system, as it must provide immediate updates on parking availability to be useful.
41. D ($\bar{X} + Y + Z$): Initially, $F = \overline{(\bar{Y}.X)} + Z = (Y + \bar{X}) + Z = \bar{X} + Y + Z$. Without simplifying using De Morgan theorem, we wouldn't have gotten to the final answer.
A represents an inverter on Y and an **AND** on (X and Y) instead of **NAND**. B represents a **NAND** and an **AND** on instead of an **OR**. C represents a **NOR**, **NOT**, & **OR** instead of a **NAND**, **NOT** and **OR** gates.

42. **D (I, II, III)**: Signal A is continuous (analog). Signal B is discrete/step-like (digital). B follows the shape of A, making it a digitized version.
A, B, C are incomplete because all three statements provided in the prompt are technically accurate.
43. **C (Technical feasibility)**: Technical feasibility assesses whether the organization has the hardware, software, and expertise (skills) to build the proposed system.
A is about money/ROI. B is about how it fits daily workflows. D is about laws and regulations.
44. **C (UPDATE USER SET City = 'Douala' WHERE City = 'Yaounde';)**: The UPDATE command is used to modify existing records. You set the new value ('Douala') for the records that match the condition ('Yaounde').
A and B use the keyword MODIFY, which is incorrect SQL for updating data. D swaps the values, setting them to 'Yaounde'.
45. **B (1, 5, 9, 7, 25)**:

i	0	1	2	3	4
nlist[i]	1	2	3	4	5

Length	i	i mod 2	nlist
5	0	0	1
	1	1	1, 5
	2	0	1, 5, 9
	3	1	1, 5, 9, 7
	4	0	1, 5, 9, 7, 25

For even indices (0, 2, 4): $1^2 =, 3^2 = 9, 5^2 = 25$. For odd indices (1, 3): $2 + 3 = 5, 4 + 3 = 7$. Result: {1, 5, 9, 7, 25}
A has considered index starting from 1 to 5 instead of 0 to 4. D has incorrect math. C is similar but has a digit swap.
46. **D (0-c, 1-a, 2-b, 3-d)**: 0NF has repeated data (c). 1NF single valued attributes/atomic values (a). 2NF ensures full functional dependency (b). 3NF removes transitive dependencies (d).
A, B, C incorrectly match the definitions to the normal form levels.
47. **A (00110010):** A logical right shift moves all bits to the right and fills the leftmost gaps with zeros. 11001011 shifted twice becomes 00110010.
B is an arithmetic shift (preserves sign bit). C is a left shift. D is a logical AND operation result.
48. **B (It provides a human-readable shorthand for binary)**: One hex digit represents exactly 4 binary bits (a nibble). This makes long binary strings much easier for humans to read and debug.
A is false; computers process binary. C is false; hex doesn't change signed logic. D is false; BIOS uses machine code.
49. **B (The completion of a time slice...)**: In multitasking, when a process's allotted time (time slice) ends, the OS moves it from **Running** back to **Ready** to let another process take the CPU.
A moves a process to **Blocked/Waiting**. C and D usually lead to the **Terminated** state.
50. **D (11)**: First, `func2(2, 4)` returns $2 \times 4 = 8$. Then, `func1(3, 8)` is called, which returns $3 + 8 = 11$.
A (12) is 3×4. B (14) is $(3 + 4) \times 2$. C (15) is $(3 \times 4) + 3$.

PAPER 2

QUESTION 1:

i)

a) Fetch stage description:
- The address in the Program Counter (PC) is copied Memory Address Register (MAR).
- A read signal is sent via the control bus to main memor
- The instruction at that address is fetched and placed in the memory Buffer Register (MBR)
- Then transferred to the Current Instruction Register (CIR).
- The PC is incremented.

b) The MBR is a temporary buffer for memory data and may be overwritten during the execute stage when operands are fetched. The CIR is dedicated to holding the current instruction for decoding and execution.

c) It uses separate buses and memory for instructions and data, allowing the CPU to fetch an instruction and read/write data simultaneously, avoiding the "Von Neumann bottleneck".

ii)

a) Exponent: 1101
b) Mantissa: 011100000000
c) Positive: The MSB (Most Significant Bit) of the mantissa is 0.
d) Mantissa: $0.11100000000_2 = 0.5 + 0.25 + 0.125 = 0.875$
Exponent: $1101_2 = -8 + 4 + 0 + 1 = -3$
$Mantissa \times 2^{Exponent} = 0.875 \times 2^{-3}$
$= 0.875 \times 0.125 = 0.109375$

e)
- Increasing exponents bits increases the range of numbers
- Decreasing mantissa bits decreases the precision (accuracy).

QUESTION 2

(i)

a) These are the steps:
- Feasibility study (technical, economic, legal).
- Requirements gathering (interviews, questionnaires).
- Documentation (data flow diagrams).
- Problem identification.
- Requirements specification.

b) Any two:
- Passenger's full name
- Gender
- Place of birth
- Date of birth
- Telephone number
- Email
- National Identify Number (NIC)
- etc.

c) Reasons (any one) :
- To ensure data accuracy and integrity
- To comply with data protection regulations
- To maintain system reliability and efficiency

d)

Situation	Method
Identifying spelling errors	Proof-reading
Using a range check	Validation
Typing in data twice	Verification
Visually comparing data on screen with the original source document	Proof-reading/Verification

(ii)

a) Any two methods:
- **Observation**: Watch users perform actual tasks in real environment.
- **Questionnaire** : Distribute surveys for quantitative usage data.

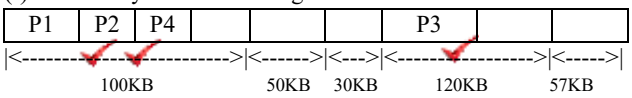
- **Document Analysis**: Reviews manuals, forms, reports, existing code.

b) (1 mark stating, 2 marks explanation)

Parallel implementation: Old and new systems run simultaneously, allowing comparison and fallback.

QUESTION 3

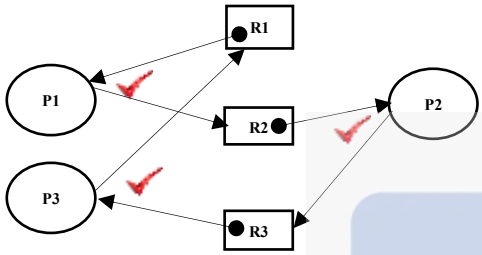
(i) Memory Allocation Diagram



(ii)

a) When processes are blocked, each waiting for a resource held by another.

b) Resource Allocation Graph (RAG)



c) **Yes**, a circular wait exists: **P1→R2→P2→R3→R1→P1**

Methods used to avoid deadlock (any one):

- Resource ordering: Allow resource order to request resources.
- Avoid hold-and-wait: Acquire all resources at once.
- Deadlock detection & recovery (if avoidance fails)
- Safe state detection (Banker's algorithm)

(iii) Gantt charts:

a)

SRTF

0 P1 1	1 P2 5	5 P4 10	10 P1 17	17 P3 26

RR

0 P1 4	4 P2 8	8 P3 12	12 P4 16	16 P1 20	20 P3 24	24 P4 25	25 P3 26

b)

RR

Process	AT	BT	FT	TAT (FT-AT)
P1	0	8	20	20
P2	1	4	8	7
P3	2	9	26	24
P4	3	5	25	22
			Average	$\frac{73}{4} = 18.25$

SRTF

Process	AT	BT	FT	TAT (FT-AT)
P1	0	8	17	17
P2	1	4	5	4
P3	2	9	26	24
P4	3	5	10	7
			Average	$\frac{52}{4} = 13$

QUESTION 4

(i)

a) Truth Table

INPUTS				INTERMEDIATE POINTS					OUTPUT
X3	X2	X1	X0	A	B	C	D	E	Q
0	0	0	0	1	0	1	0	0	0
0	0	0	1	1	0	1	0	0	0
0	0	1	0	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1
0	1	1	0	0	0	1	1	1	1
0	1	1	1	0	1	0	0	1	1
1	0	0	0	1	0	1	0	0	1
1	0	0	1	1	0	1	0	0	1

b) $Q = \overline{X3}\,\overline{X2}\,X1\,\overline{X0} + \overline{X3}\,\overline{X2}\,X1\,X0 + \overline{X3}\,X2\,X1\,\overline{X0} + \overline{X3}\,X2\,X1\,X0 + X3\,\overline{X2}\,\overline{X1}\,\overline{X0} + X3\,\overline{X2}\,\overline{X1}\,X0$

c) $Q = \overline{X3}\,\overline{X2}\,X1(\overline{X0} + X0) + \overline{X3}\,X2\,X1\,(\overline{X0} + X0) + X3\,\overline{X2}\,\overline{X1}(\overline{X0} + X0)$

$= \overline{X3}\,\overline{X2}\,X1 + \overline{X3}\,X2\,X1 + X3\,\overline{X2}\,\overline{X1}$

$= \overline{X3}\,X1(\overline{X2} + X2) + X3\,\overline{X2}\,\overline{X1}$

$= \overline{X3}\,X1 + X3\,\overline{X2}\,\overline{X1}$

(ii)

a) Devices (Any three):

- Webcam
- Microphone
- Speakers
- Headphones

b) Potential Problems (Any two):

- **Hardware & Connectivity**: Technical lag or latency causes synchronization issues, while equipment failure (mics/webcam) disrupts flow.
- **Communication & Wellness**: Psychological fatigue sets in from screen time, and the lack of physical presence leads to a loss of non-verbal cues and body language.
- **Logistics & Security**: Time zone differences make scheduling difficult, and poor encryption creates security vulnerabilities.
- **Financial**: High initial setup costs for quality professional hardware.

c) Process of scrambling readable information (plaintext) into a secret code, making it unreadable to anyone who doesn't have a special key to unlock it. It's used to security during transmission.

QUESTION 5

(i)

a) A flat file database is a type of database that stores data in a single table or multiple files without relationships.

b) Drawbacks of file-based approach (Any three):

- **Data redundancy**: Same data repeated across files or a table.
- **Data inconsistency**: Updates may not sync across all files or table.
- **Poor data integrity**: No constraints or validation rules.
- **Limited querying**: Complex searches are slow or impossible.
- **Security weaknesses**: Access control is difficult to enforce.

c) Reasons:

- No primary key
- Redundancy (Repeating groups like Paul Schloss/Jay Patel)
- Non-atomic data (Customer name contains both first name and last names)

d) DDL to create table GUIDE_TABLE:

```
CREATE TABLE GUIDE_TABLE
(
        GuideID INT PRIMARY KEY,
        Guide VARCHAR(50),
        DateOfBirth DATE,
        Gender CHAR(1)
);
```

(ii)

a) Measure:
- User authentication (Passwords)
- Installation of Firewall

b) Causes of data loss (Any one):
- Hardware failure
- Floods
- Human error
- Software corruption

c) Preventive methods (Any one):
- Regular backups
- Store off-site
- Installing antivirus
- Keeping operating system updated
- Encrypting sensitive data

QUESTION 6

(i)

a) Bubble sort in descending order:

✓ Pass 1:

Anna 56	Abba 47	Baker 24	Emma 87	Aaron 63	Ambe 90

Swap Anna 56 and Abba 47

Abba 47	Anna 56	Baker 24	Emma 87	Aaron 63	Ambe 90

Swap Anna 56 and Baker 24

Abba 47	Baker 24	Anna 56	Emma 87	Aaron 63	Ambe 90

No swap between Anna 56 and Emma 87

Abba 47	Baker 24	Anna 56	Emma 87	Aaron 63	Ambe 90

Swap Emma 87 and Aaron 63

Abba 47	Baker 24	Anna 56	Aaron 63	Emma 87	Ambe 90

No swap between 87 and 90

Abba 47	Baker 24	Anna 56	Aaron 63	Emma 87	Ambe 90

✓ Pass 2:

Swap Abba 47 and Baker 24

Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

No swap between Abba 47 and Anna 56

Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

No swap between Anna 56 and Aaron 63

Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

No swap between Aaron 63 and Emma 87

Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

No swap between Emma 87 and Ambe 90

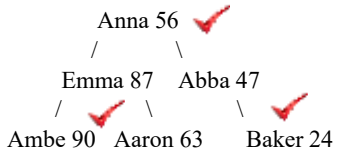
Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

✓ Sorted

Baker 24	Abba 47	Anna 56	Aaron 63	Emma 87	Ambe 90

b) Alternative sorting algorithm (Any one):
- Selection sort
- Insertion sort
- Quick sort
- Merge sort

c) Binary tree for scores and names:



Inorder: Ambe 90, Emma 87, Aaron 63, Anna 56, Abba 47, Baker 24

d) **No**, because the node Abba 47 does not have a left child.

e) Anna 56, Emma 87, Abba 47, Ambe 90, Aaron 63, Null, Baker 24.

(ii)

a) Logic expression:

```
IF Y MOD X AND X <> Y THEN
  RETURN true
ELSE
  RETURN false
ENDIF
```

b)

```
FUNCTION sumPD(N)≡
  DECLARE SUM, i AS INTEGER
  SET SUM = 0
  FOR i FROM 1 TO N-1 DO
     IF i MOD N == 0 THEN
          SET SUM = i + SUM
     ENDIF
  NEXT i
  RETURN SUM
ENDFUNCTION
```

QUESTION 7

(i)

a) A central computer that manages resources and provides services to other computers (clients) on the network.

b) Computers and the printer.

c) External threats (Any two):
- Malicious software like worms, viruses, spyware, trojan
- Physical threats like floods, earthquakes, theft, hardware damage, etc.
- Denial Of Service Attacks
- Phishing
- Hacking

d) Preventive measures (Any one):
- Install firewall software and keep it updated every time.
- Install antivirus/anti-spyware and keep it updated every time.
- Always keep your system updated.
- Do regular scans of your entire system.
- Make use of strong and difficult to crack passwords (having uppercase, lowercase, numbers and special characters)
- Do open suspicious mails or links.
- User education on threats and preventive measures.

(ii)

a) Network terms:
- **Switch/Bridge**: Receives data packets from a network and forwards them onto a similar network.
- **Server**: Manages access to a centralised resource.
- **Gateway/Router**: Joins networks that use different sets of rules to transmit data.
- **Firewall**: Monitors and controls incoming and outgoing network traffic based on set criteria.

b) Advantages of serial data transmission over parallel data transmission (Any one):
- Superior long-distance reliability
- Lower costs
- Reduced hardware complexity

c) They allow the internet to function as a cohesive global network rather than a collection of isolated, incompatible systems.

QUESTION 8

(i)

a) Example of errors:

Type	Error	Correction
Syntax	esle	else
Logic	If r < 0 then	If r >= 0 then

b) Interpreter flags syntax errors at runtime while compiler catches it during compilation.

(ii)

a) Answers:

the memory location of the value that has been on the stack for the longest time	**400**
the number of consecutive push operations that will result in the TopOfStack variable containing 409	**6**

b) Operations of the stack:

409	
408	'Y' ←TopOfStack
407	'Z'
406	'C'
405	'F'
404	'K'
403	'B'
402	'S'
401	'R'
400	'D'

(iii)

a) *Line 07: RETURN NumberPattern(Value1, Value2, EndValue) + 1*
because a recursive function is a function that is being called by itself.

b) Trace table:

Value1	Value2	Temp	EndValue	OUTPUT	RETURN value
1	1	1	12	1	5
2	1	1		2	4
3	2	2		3	3
5	3	3		5	2
8	5	5		8	1
13	8			13	0

Final returned value: 5

c) To output/find a value that is the addition of the two previous values // (output) Fibonacci sequence

PAPER 3

PROGRAMMING: A COMPUTER PROGRAM FOR YOUNGER STUDENTS

```
(1) //inputInRange()
    int N;//global variable
    int inputInRange()
    {
       while(N<1 || N>25)
       {
              printf("The number you entered
    is not in the range 1-25. Please try
    again\n");
              printf("Enter a number\n");
              scanf("%d",&N);
       }
       return N;
    }
(2) //insertValues(MAT)
    void InsertValues(int MAT[][5])
    {
       int i, j;

       for(i=0;i<5;i++)
       {
              for(j=0;j<5;j++)
              {
                     printf("Enter a
    number\n");
                     scanf("%d",&N);
                     MAT[i][j] =
    inputInRange();
              }
       }
    }
```

```
(3) //Initialise()
    int FLAGS[25];//global variable
    void Initialise()
    {
       int i;
       for(i=0;i<25;i++)
       {
              FLAGS[i] = 0;
       }
    }

       OR

    void Initialise()
    {
       int FLAGS[25] ={0};
    }

(4) // isCorrectlyCompleted
    void isCorrectlyCompleted(int MAT[][5])
    {
       int i, j, Num;

       Initialise();
       //Visit all elements of the MAT array
       for(i=0;i<5;i++)
       {
              for(j=0;j<5;j++)
              {
                     Num = MAT[i][j];
                     //Assign one (1) to the
    FLAGS array at that index
```

```
                FLAGS[Num-1] = 1;
            }
        }
        //Inspect elements of the FLAGS array
        for(i=0;i<25;i++)
        {
            if (FLAGS[i] == 0)
            {
                isComp = false;
                break;
            }
        }
    }
(5) //Missing Numbers
    void MissingNumbers()
    {
        int i;
        printf("\nThe missing numbers
    are:\n");
        for(i=0;i<25;i++)
        {
            if (FLAGS[i] == 0)
                printf("%d\n", i+1);
        }
    }

(6) //Display matrix MAT
    void displayGrid(int MAT[][5])
    {
        int i, j;
        printf("\n\nThe Table\n");
        for(i=0;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                printf("%d\t", MAT[i][j]);
            }
            printf("\n");
        }
    }

(7) Main program
    int main()
    {
        int MAT[5][5];

        //Prompt the user to insert values of
    the grid MAT
        InsertValues(MAT);
        //Display the array MAT
        displayGrid(MAT);
        //Verify if array MAT is correctly
    completed.
        isCorrectlyCompleted(MAT);
        //verify if the matrix is correctly
    completed
        if (isComp == true)
            printf("\nThe Table has been
    correctly completed\n");
        else
        {
            printf("\nThe Table has not been
    correctly completed\n");
            MissingNumbers();
        }
        return 0;
    }
```

(8) Screenshot2

```
The Table
2       24      1       8       15
23      5       7       14      16
4       6       13      20      22
10      12      19      21      3
11      18      25      17      9

The Table has been correctly completed
```

Completed correctly

```
The Table
11      24      1       8       15
23      5       7       14      16
4       6       13      20      22
10      2       19      21      3
11      18      25      2       9

The Table has not been correctly completed

The missing numbers are:
12
17
```

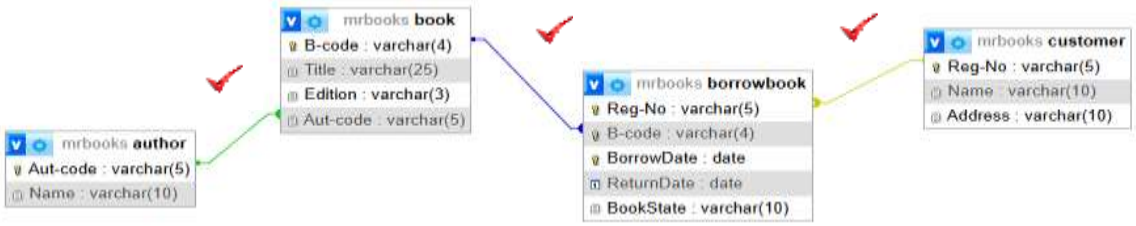
Not Completed correctly

DATABASE: HOTEL MANAGEMENT SYSTEM

(1) Relational schema

- *Author(Aut-code, Name)*
- *Book(B-code, Title, Edition, Aut-code)*
- *BorrowBook(Reg-No, B-code, BorrowDate, BookState)*
- *Customer(Reg-No, Name, Address)*

(2) Entity Relationship Diagram



(3) CREATE DATABASE MrBooks;

(4) Creating all relations (Any three):

- Author Table

```
CREATE TABLE `Author`
 (
 `Aut-code` varchar(5) NOT NULL,
 `Name` varchar(10) NOT NULL,
 PRIMARY KEY (`Aut-code`)
);
```

- Book Table

```
CREATE TABLE `Book`
 (
 `B-code` varchar(4) NOT NULL,
 `Title` varchar(25) NOT NULL,
 `Edition` varchar(3) NOT NULL,
 `Aut-code` varchar(5) NOT NULL,
 PRIMARY KEY (`B-code`),
 FOREIGN KEY (`Aut-code`) REFERENCES
`author` (`Aut-code`)
);
```

- Customer Table

```
CREATE TABLE `Customer`
(
 `Reg-No` varchar(5) NOT NULL,
 `Name` varchar(10) NOT NULL,
 `Address` varchar(15) NOT NULL,
 PRIMARY KEY (`Reg-No`)
);
```

- BorrowBook Table

```
CREATE TABLE `BorrowBook`
(
 `Reg-No` varchar(5) NOT NULL,
 `B-code` varchar(4) NOT NULL,
 `BorrowDate` date NOT NULL,
 `BookState` varchar(10) NOT NULL,
 PRIMARY KEY (`Reg-No`,`B-
code`, `BorrowDate`),
 FOREIGN KEY (`B-code`) REFERENCES
`book` (`B-code`),
 FOREIGN KEY (`Reg-No`) REFERENCES
`customer` (`Reg-No`)
);
```

Any prove of three tables
Created 3 marks

(5) Populating tables:

- Author

```
INSERT INTO `Author` (`Aut-code`, `Name`)
VALUES ('AUT01', 'Jane A.');
INSERT INTO `Author` (`Aut-code`, `Name`)
VALUES ('AUT02', 'Charles D.');
```

Aut-code	Name
AUT01	Jane A.
AUT02	Charles D.

- Customer

```
INSERT INTO `Customer` (`Reg-No`, `Name`,
`Address`) VALUES ('CUS01', 'John P.', '456
Ave Rd');
INSERT INTO `Customer` (`Reg-No`, `Name`,
`Address`) VALUES ('CUS02', 'Epie E.', '322
UB Jun.');
INSERT INTO `Customer` (`Reg-No`, `Name`,
`Address`) VALUES ('CUS03', 'Marie D.', '796
Bonas.');
INSERT INTO `Customer` (`Reg-No`, `Name`,
`Address`) VALUES ('CUS04', 'Ekuh A.', '232
Cnt Hand.');
```

Reg-No	Name	Address
CUS01	John P.	456 Ave Rd
CUS02	Epie E.	322 UB Jun
CUS03	Marie D.	796 Bonas.
CUS04	Ekuh A.	232 Cnt Hand

- Book

```
INSERT INTO `Book` (`B-code`, `Title`,
`Edition`, `Aut-code`) VALUES ('BK01', 'Pride
and Prejudice', '3rd', 'AUT01');
INSERT INTO `Book` (`B-code`, `Title`,
`Edition`, `Aut-code`) VALUES ('BK02', 'Sense
and Sensibility', '2nd', 'AUT01');
INSERT INTO `Book` (`B-code`, `Title`,
`Edition`, `Aut-code`) VALUES ('BK03', 'Great
Expectations', '1st', 'AUT02');
```

B-code	Title	Edition	Aut-code
BK01	Pride and Prejudice	3rd	AUT01
BK02	Sense and Sensibility	2nd	AUT01
BK03	Great Expectations	1st	AUT02

- BorrowBook

```
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS01',
'BK01', '2026-01-10', 'Fair');
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS01',
'BK02', '2026-01-12', 'Excellent');
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS02',
'BK03', '2026-01-15', 'Good');
```

```
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS02',
'BK01', '2026-01-08', 'Excellent');
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS03',
'BK03', '2026-01-14', 'Good');
INSERT INTO `BorrowBook` (`Reg-No`, `B-code`,
`BorrowDate`, `BookState`) VALUES ('CUS04',
'BK03', '2026-01-05', 'Good');
```

Reg-No	B-code	BorrowDate	ReturnDate	BookState
CUS01	BK01	2026-01-10	0000-00-00	Fair
CUS01	BK02	2026-01-12	0000-00-00	Excellent
CUS02	BK01	2026-01-08	0000-00-00	Excellent
CUS02	BK03	2026-01-15	0000-00-00	Good
CUS03	BK03	2026-01-14	0000-00-00	Good
CUS04	BK03	2026-01-05	0000-00-00	Good

(6) Add ReturnDate

```
ALTER TABLE `BorrowBook`
ADD `ReturnDate` DATE NOT NULL;
```

(7) Customers who borrowed 'Pride and Prejudice'

```
SELECT `Customer`.`Name`, `Book`.`Title`,
`BorrowBook`.`BorrowDate`
FROM `Bustomer`, `Book`, `BorrowBook`
WHERE `Book`.`Title` = 'Pride and Prejudice'
AND `BorrowBook`.`B-code` = `Book`.`B-code`
AND `BorrowBook`.`Reg-No` = `Customer`.`Reg-
No`;

OR

SELECT Customer.Name, Book.Title,
BorrowBook.BorrowDate
FROM Customer INNER JOIN (Book INNER JOIN
BorrowBook ON Book.[B-code] = BorrowBook.[B-
code]) ON Customer.[Reg-No] =
BorrowBook.[Reg-No]
WHERE (((Book.Title)="Pride and Prejudice"));
```

Name	Title	BorrowDate
John P.	Pride and Prejudice	2026-01-10
Epie E.	Pride and Prejudice	2026-01-08

(8) Books most borrowed frequently

```
SELECT Book.`Title`, COUNT(*) AS `Number of
Times`
FROM BorrowBook, Book
WHERE Book.`B-code` = BorrowBook.`B-code`
GROUP BY Book.Title
HAVING COUNT(*) > 1;

OR

SELECT Book.Title, Count(*) AS [Number of
Times]
FROM Book INNER JOIN BorrowBook ON Book.[B-
code] = BorrowBook.[B-code]
GROUP BY Book.Title
HAVING COUNT(*)>1;
```

Title	Number of Times
Pride and Prejudice	2
Great Expectations	3

for more past questions and solutions download kawlo kawlo app or visit http://www.gcerevision.com