MARKING GUIDE FOR HIHS AL CSC 2025

PAPER 1

1. **D. 3CD. The other options do not correctly convert the decimal number to hexadecimal.**
2. **C/D. The expression $z = x(\bar{x} + \bar{y}) = x\bar{x} + x\bar{y} = x\bar{y}(C)$ or $z = x(\bar{x} + \bar{y}) = x(\overline{\bar{x}\bar{y}})$ (D)using De Morgan's theorem. A is incorrect because it represents a different expression. B is incorrect because it represents a different expression.**
3. **D. Flip-Flop.** A flip-flop is a basic memory element that can store one bit of data. **A. Register, B. Encoder, C. Cell**: While these components relate to data storage, only a flip-flop specifically stores a single bit.
4. **C. Bubble sort.** Bubble sort works by repeatedly stepping through the list, comparing adjacent elements and swapping them if they are in the wrong order. **A. Quick sort, B. Merge sort, D. Selection sort**: These algorithms do not focus on adjacent elements; Quick sort uses partitioning, Merge sort combines sorted arrays, and Selection sort selects the smallest element iteratively.
5. **A. Divide-and-conquer approach.** Recursive algorithms often break a problem down into smaller subproblems, which is a divide-and-conquer strategy. **B. Modular approach, C. Structural approach, D. Hierarchical approach**: These do not accurately describe the nature of recursion, which inherently involves breaking down problems.
6. **A. O(n)**. This function performs a linear number of additions, which leads to a time complexity of O(n). **B. O(log n), C. O(n^2), D. O(n!)**: These complexities do not reflect the linear recursive nature of the function, which sums up n steps.
7. **B. DRAM.** Dynamic RAM (DRAM) requires periodic refreshing to maintain the stored data. **A. SRAM, C. ROM, D. Flash memory**: SRAM is static and does not require refreshing, ROM is non-volatile and retains data without power, while Flash memory is also non-volatile and doesn't need refreshing.
8. **B. Shortest Job First.** The Shortest Job First (SJF) scheduling algorithm minimizes the average waiting time. **A. First Come First Served, C. Round Robin, D. Priority Job First**: These algorithms can lead to longer average wait times compared to SJF, especially in the presence of varying job lengths.
9. **B. More than one program in memory.** A time sharing system implies that more than one program is in memory and multiple programs are executed concurrently by sharing the CPU's time. The other options are incorrect because they do not accurately describe the concept of a time sharing system.
10. **C. Reduce costs.** Banks deploy web self-services mainly to cut operational expenses (e.g., staffing, infrastructure) tied to traditional support channels. Reducing costs directly improves profitability. A is incorrect because self-service enhances customer value. B (risk management) is unrelated. D (opportunities) is secondary.
11. **B. Documentation.** Documentation includes comments, guides, and explanations to clarify code functionality for developers. A (commentary) is a subset of documentation. C (implementation) refers to coding itself, not explanations. D (programming) is the act of writing code.
12. **B. Systems programs.** Systems software (e.g., OS, drivers) sets up the foundational environment for machines to operate. A (applications) run on top of systems. C (libraries) are code modules. D (utilities) perform specific tasks but aren't foundational.
13. **C. Object code.** Object code is machine-readable output from a compiler. It requires linking to libraries to form executables. A (source) is human-readable. B (executable) is final linked code. D (byte code) is intermediate (e.g., Java).
14. **C. UPDATE USER SET City = 'DC' WHERE City = 'Yde';** SQL uses UPDATE (not MODIFY) to alter records. A/B use invalid syntax. D swaps 'Yde' and 'DC', reversing the intent.
15. **D. 0 – iii, 1 – I, 2 – ii, 3 – iv**

- 0NF: Repeating values (iii) are allowed.
- 1NF: Single-valued attributes (i), no repeating groups.
- 2NF: Full functional dependency on the entire primary key (ii).
- 3NF: No transitive dependencies (iv).

Other options misalign these definitions with their respective normal forms.

16. **D. All of the above**. In the design phase of program development, the term "module" refers to functions, procedures, and subprograms. All of these options are correct as they represent different ways of organizing and structuring code in a modular manner.
17. **A. 00 101 000.** Correct binary subtraction. B: Incorrect due to wrong result and sign bit.; C: Incorrect value after subtraction.; D: Incorrect result and sign bit.
18. **D. All. NAND & NOR gates** will resolve to $C = A.B$ which gives C = 0 when A = 0 & B = 1. XOR will resolves to $C = 0$.
19. **D. 613**. Final value via indirect addressing. A: Initial address, not value; B: Intermediate address; C: Intermediate address.
20. **C. 0101110100 000101**. The normalized floating point representation of 23.25 is 0101110100 000101. Option A has incorrect bits in the exponent. Option B has incorrect bits in both the mantissa and the exponent. Option D has incorrect bits in the mantissa.
21. **C. 64K memory locations.** With a 16-bit program counter, the CPU can address $2^{16} = 64K$ memory locations. A is incorrect because it represents a smaller memory capacity. B is incorrect because it also represents a smaller memory capacity. D is incorrect because it represents a larger memory capacity.
22. **C. Memory parity error** has the highest priority as it indicates a critical error in memory integrity. A is a software exception, B is a lower priority interrupt, and D is also a lower priority interrupt.
23. **B. Optical fiber provides the highest data rate**. A has a lower data rate, C has a lower data rate, and D have a lower data rate compared to optical fiber.
24. **B. 110110.** To find the 2's complement of 1010 in 6 bits, we invert the bits and add 1. Options A, C, and D do not correctly represent the calculation of the 2's complement.
25. **A.** $\frac{n(n-1)}{2}$. B. Incorrect because $\frac{(n+1)(n-1)}{2}$ is not the formula for calculating connections in a full mesh topology. C. Incorrect because $\frac{n(n+1)}{2}$ is not the formula for calculating connections in a full mesh topology. D. Incorrect because $\frac{(n-1)(n-2)}{2}$ is not the formula for calculating connections in a full mesh topology.
26. **D: traversal**. Traversal refers to the operation of processing each element in a list or data structure. Sorting (A) arranges elements in a specific order, searching (B) finds a specific element, and indexing (C) refers to accessing elements by their position.
27. **B. Central point of failure**. A star topology's central hub or switch is crucial; if it fails, the entire network goes down, affecting all connected devices. A. High costs of cabling is a financial issue but not as immediate as a single point of failure. C. Difficulty in troubleshooting is incorrect because star topology makes fault detection easier due to its centralized structure D. Limited scalability, while true that performance can degrade with many devices, isn't the primary disadvantage compared to a central failure risk.

28. **B. To filter and forward packets between different networks.** Routers primarily connect multiple networks, forwarding data packets between them while filtering traffic for security and efficiency. A describes a switch's role within a LAN. C is more related to repeaters or amplifiers. D refers to physical connections, which routers do not provide directly.
29. **B. 2.** The query counts employees with salaries between $60,000 and $100,000 whose names start with 'M'. "Mahas" & "Mary" fits this criterion (Salary between $60,000 and $100,000). Thus, the output is 2. Other options are incorrect because they miscount the matching records based on the given conditions.
30. **D. Derived attribute**. A derived attribute is calculated from other related attributes in a database. For example, age can be derived from the date of birth. A. Composite attribute combines multiple simple attributes into one. B. Simple attribute is a basic, indivisible piece of data. C. Multivalued attribute can have multiple values for a single entity.
31. **B. I and II only**. I is correct because one farmer can have many tasks (one-to-many). II is correct as one adviser can advise many tasks (one-to-many). III is incorrect because the Advises relation implies that for each task, there's typically one adviser assigned at a time; having many advisers per task isn't directly supported by the given schema structure without additional relationships or assumptions.
32. **A. System testing.** System testing evaluates the application in a setup that mirrors real-world conditions. B. Integration testing: Tests interactions between components, not the entire system. C. Acceptance testing: Validates software against requirements, usually by end-users. D. Unit testing: Tests individual components in isolation.
33. **C. New -> Ready -> Blocked -> Running -> Terminated.** This sequence represents the correct process lifecycle. A. New -> Ready -> Running -> Terminated: Omits the blocked state. B. New -> Blocked -> Terminated: Skips ready and running states. D. New -> Running -> Ready -> Blocked -> Terminated: Incorrect order; running cannot jump directly to ready.
34. **D. Half duplex**. In half-duplex communication, transmission can occur in only one direction at a time. A is incorrect because Simplex communication is one-way only. B is incorrect because Full duplex communication allows simultaneous transmission in both directions. C is incorrect because half simplex communication is one-way but in a different direction during communication.
35. **B. 12**. p is then multiplied by q, resulting in p = 2 * 6 = 12; The loop iterates until p is less than q; After the first iteration, p = 5 and q = 4; After the second iteration, p = 2 and q = 6; The loop terminates after the second iteration.
36. **C. By value**. Functions can return values by reference, but this is not the default behavior. By default, functions return values by value, meaning that the actual value of the variable is copied into the calling function; Passing parameters by reference allows the function to modify the original variable in the calling function.
37. **D. 21.** After the while loop, x = 16. Which implies x = 16+5 = 21.
38. **B. 3.** Bubble sort requires 3 passes to sort the array | 5 | 1 | 4 | 2 | in ascending order.
    1. First Pass: | 1 | 4 | 2 | 5 | (after swaps).
    2. Second Pass: | 1 | 2 | 4 | 5 | (after one swap).
    3. Third Pass: No swaps needed; the array is sorted.

    A. 2: Not enough passes. C. 4: Extra pass is unnecessary. D. None: Incorrect, as 3 passes are needed.
39. **B. -9**; f(-3, -3) = -3 + f(-3, -2) (since b < 0); f(-3, -2) = -3 + f(-3, -1) (since b < 0); f(-3, -1) = -3 + f(-3, 0) (since b < 0); f(-3, 0) = 0 (since b = 0). Therefore, f(-3, -3) = -3 + -3 + -3 + 0 = -9.
40. **B. In order.** Pre order: 9, 5, 3, 7, 13, 11, 15; In order: 3, 5, 7, 9, 11, 13, 15; Post order: 3, 7, 5, 11, 15, 13, 9; Mixed order is not a valid tree traversal method.
41. **A. It represents the recursive definition of the Fibonacci series, where each number is the sum of the two preceding numbers.** B is incorrect because it uses a different formula. C is incorrect because it calculates the factorial. D is incorrect because A is the correct answer.
42. **D. 16.**

r		product
10		
7		
4		16
1		

    Incorrect: A. 0 – r is never 0. B. 6 – this is the product after third test. C. 14 - this is the product after the first test.
43. **B. Test Plan** - Outlines the procedures to verify if a system functions as intended, making it ideal for identifying conflicts with existing applications. A. Design Specification - focuses on design, not compatibility. C. Technical Manual - provides technical info, not specifically about compatibility. D. Installation Manual – guides installation, not about compatibility.
44. **A. Priority scheduling allocates CPU based on highest priority**. B, C, D do not reflect priority-based allocation.
45. **D. Messages.** Messages are not a primary object-oriented feature that facilitates software reuse.
46. **B. MS Windows GUI;** Linux CLI – MS Windows GUI offers user-friendly interface for Google Earth, while Linux CLI provides a foundational learning experience for new system administrators. A. MS Windows GUI; MS Windows GUI - not suitable for learning, B. MS Windows GUI; Linux CLI - not consistent for both users, C. Linux CLI; Linux CLI - not suitable for navigation.
47. **A. Computer Games satisfy expert users and can be recreational.** B, C, and D are incorrect because they do not involve creativity or exploiting expertise.
48. **B. Computer hardware is organised into functional units and subsystems.** This highlights the structural organization of hardware. A: Too general, does not specify units and subsystems; C: Focuses on tasks, not hardware organization; D: Mentions design, not organization into units.
49. **D. The address in a program requires an offset from its page address only**. Requires segment and offset. A: True for paging; B: True for segmented memory; C: True, pages vary in size.
50. **D. Software quality assurance.** These activities ensure that the software meets the specified requirements and is free of defects. Design (A), analysis (B), and maintenance (C) are stages of the software development life cycle (SDLC) and are not specific to verification and validation.

QUESTION 1:

i)

a) Let's determine the decimal equivalent of the mantissa and exponent first:
Mantissa: 1.0110000
This implies it's a negative number since the MSB is 1.
Unsigned: $01001111 + 1 = -0.1010000$
$\rightarrow -0.1010000 = 2^{-1} + 2^{-3} = -\frac{5}{8} = -0.625$
$Exponent: 0011 = 2^1 + 2^0 = 2 + 1 = 3_{10}$
$answer = mantissa \times 2^{exponent} = -0.625 \times 2^3$
$= -5$

b) Convert the decimal 58.5 to binary first:
- Convert 58 to binary:

2	58	R
2	29	0
2	14	1
2	7	0
2	3	1
2	1	1
2	0	1

58 = 111010
- Convert 0.5 to binary
$0.5 \times 2 = 1.0$
$\rightarrow 0.5 = 0.1$
$\therefore 58.5 = 111010.1$
Shifting the binary point to 6 places:
$\rightarrow mantissa = 0.1110101$ *and exponent* = 0110

ii)

a) The identity is true because no matter the value of A (0 or 1), A or its negation ($\bar{A}$) must be 0. And 0 times anything will result to 0.

b)
$\overline{\overline{\bar{B}.A.\bar{B}} + A.B} = \overline{(B + \bar{A}).\bar{B}} + A.B$
$= \overline{B + \bar{A}} + \overline{B} + A.B$
$= \bar{B}.A + B + A.B$
$= A(\bar{B} + B) + B = A + B$

c)

iii)

a) **Stored program concept:** It refers to the idea that Program instructions and data are stored in the same memory unit and the CPU executes instructions sequentially from memory.

b) **Instruction Fetch and Load using Register Transfer Notation (RTN):**
$MAR \leftarrow [PC]\ //MemoryAddressRegister$
$\leftarrow ProgramCounter$
$MDR \leftarrow MEM[MAR]\ //MemoryDataRegister$
$\leftarrow Memory[AddressinMAR]$
$PC \leftarrow [PC] + 1\ //ProgramCounter$
$\leftarrow ProgramCounter + 1$
$IR \leftarrow [MDR]\ //InstructionRegister$
$\leftarrow MemoryDataRegister$

c) **Explanation:**
- ✓ The Program Counter (PC) contains the address of the next instruction to be fetched.
- ✓ The MAR is loaded with the address from the PC.
- ✓ The Memory Data Register (MDR) is loaded with the instruction from the memory location specified by the MAR.
- ✓ The PC is incremented to point to the next instruction.

QUESTION 2

(i)

a)

Statement	Assembler	Interpreter	Compiler
Translates and executes each line of source code one line at a time		✓	
Translates low-level source code into machine code	✓		
Must be present in memory to execute the code		✓	
Translates high-level source code into low-level code		✓	✓

b) Benefits of using compiler:
- ✓ Compiled code can run faster than interpreted code because it is translated into machine code whereas interpreted code is translated and executed line by line, making it slow.
- ✓ Once code is compiled, the Compiler is no longer needed to run the program. Thereby, reducing memory usage compared to interpreters.
- ✓ Identifies errors before execution. Making it easier to debug and improve code quality.

(ii)

a) Maximum Storage Space = 50 x 2.5 MB = 125 MB. A suitable medium for sending copies to guests would be a USB drive because it can easily hold the 125 MB of data and is portable and convenient for mailing.

b)
- ✓ **Primary Storage**: This is the main memory used by a computer to store data temporarily while it is in use. Example: RAM (Random Access Memory).
- ✓ **Secondary Storage**: This refers to storage devices that hold data permanently or for long-term use. Example: Hard drives or SSDs (Solid State Drives).
- ✓ **Off-line Storage**: This is storage that is not immediately accessible by a computer and requires manual intervention to access. Example: External hard drives or USB drives that are not connected to the computer.

(iii)

a) There are 2 addressing modes (direct and Indirect)
$\therefore number\ bits = \log_2 2 = 1\ bit$

b) There are 256 registers
$\therefore number\ bits = \log_2 256 = \log_2 2^8 = 8\ bits$

c) There are 512K words. i.e. 512 * 2^{10} = 524288 = 2^{19} words.
$\therefore number\ bits = \log_2 2^{19} = 19\ bits$

d) The remaining bits of 32 bits are used to identify the opcode.

$\therefore number\ bits = 32 - (1 + 8 + 19) = 32 - 28 = 4\ bits$

QUESTION 3

(i) $Number\ of\ possible\ schedules: n! = {}^{n}_{r}P$

(ii) **Preemptive**:

- ✓ Allows processes to be interrupted and resumed later.
- ✓ Ensures fairer resource allocation.

Non-preemptive:

- ✓ Once a process starts running, it cannot be interrupted.
- ✓ Simpler to implement, but may lead to starvation.

(iii)

a)

- ✓ RR

Process	AT	BT	CT	TAT(CT-AT)	WT
A	0	3	3	3	0
B	2	6	17	15	9
C	4	4	11	7	3
D	6	5	20	14	9
E	8	2	19	11	9

0 A 3	**3 B 7**	**7 C 11**	**11 D 15**	**15 B 17**	**17 E 19**	**19 D 20**

- ✓ SRTF

Process	AT	BT	CT	TAT(CT-AT)	WT (TAT-BT)
A	0	3	3	3	0
B	2	6	15	13	7
C	4	4	8	4	0
D	6	5	20	14	9
E	8	2	10	2	0

0 A 3	**3 B 4**	**4 C 8**	**8 E 10**	**10 B 15**	**15 D 20**

b)

- ✓ RR: ATAT = $\frac{(3+15+7+14+11)}{5} = \frac{50}{5} = 10\ units$
- ✓ SRJF: ATAT = $\frac{(3+13+4+14+2)}{5} = \frac{}{5} = 7.2\ units$

QUESTION 4

(i)

a) The interconnection of two or more computers and network devices that communicate and share resources.

b) Similarities and differences:

Peer-to-Peer	Client-Server
Each device can act both as a server and client	One computer acts as server which provides resources to the clients
Decentralized allowing direct communication with other devices	Centralized relying on a server for resource management
Lower costs due to fewer resources required	Higher Cost due to more investment in Hardware and infrastructure

(ii)

a) Characteristics of RAM

- ✓ Loses its stored data when powered off.
- ✓ Fast access speed allowing quick data retrieval for active apps.
- ✓ Temporary storage for active processes

b) Persistent storage on the computers is to retain data even when the computer is turned off, allowing retrieval of files and applications

c) Types of general application software:

- ✓ Word processors (e.g., Microsoft Word).
- ✓ Spreadsheets (e.g., Microsoft Excel).
- ✓ Presentation software (e.g., Microsoft PowerPoint).

d)

- ✓ Advantages:
  1. Reduces Cost by allowing personal devices to be used.
  2. Increased flexibility for teachers since they work with devices they are familiar with.
- ✓ Disadvantages
  1. Potential security risks from unknown devices connecting to network.
  2. Face challenges for providing support for diverse devices and software.

e) Methods of security for the wireless network:

- ✓ WPA3 encryption to secure data transmission over the network.
- ✓ MAC address filtering to control which devices can connect.

f) The speed of data transmission on the wireless network may vary due to:

- ✓ interference from other devices,
- ✓ physical obstructions (walls), and
- ✓ the distance between devices and the access point.

QUESTION 5

(i)

a) A flat file database is a type of database that stores data in a single table

b)

- *Referential integrity* refers to the accuracy and consistency of data within a relationship. That is, **whenever a foreign key value is used it must reference a valid, existing primary key in the parent table.**

Example:

User

Userid	Username	Password
1	Jeskovic	grandista
2	Manasta	Resqui12

Review

Userid	Restid
1	2
1	3

We can see that the foreign key value of Userid 1 is found in the parent table User.

Database Administrators responsibilities:

- restricting access to certain users,
- controlling what each user can do and
- running anti-virus software.

(ii) a)

- Use a text field for the Supp_Phone with a length of 15 characters (*length check*) to ensure that an entered phone number is not shorter or longer than 15 characters. Entering fewer or more characters makes a phone number invalid.
- Use a validation rule (format check) to ensure the first six characters can only be "(+237)"
- Use an input mask such as LLLLLL000000000 (L represents any letter and 0 represents any digit) to ensure the last 9 entries are only numbers. For example (+237) 6542 98001 will be accepted as valid.

b)

Prod_ID	Prod_Name	Prod_Price	Supp_Name	Supp_Contact	Supp_Phone
P211	Apples	2625	Fruit and Veggie	Ines Bond	(+237) 6542 98001
P121	Onions	1575	Fruit and Veggie	Ines Bond	(+237) 6542 98001

c)

– When inserting a tuple supplied by an existing supplier, it may lead to *inconsistency* if the data entered is not consistent with the data about this supplier in all other tuples;
For example, inserting:

P189	Potatoes	2700	Veggy Co.	Mia Abiss	(+237) 6234 76547

– Deleting a tuple could remove data which is not intended to be lost;
For example; deleting the following tuple.

P265	Tomatoes	4000	New Fruits	John Smith	(+237) 677 789521

Which is the only tuple in the relation which holds data about New Fruits suppliers (all data about this supplier will be lost); **[1]**

- Modifying/changing the phone number for Fruit and Veggie. would require that each and every tuple containing item supplied by Fruit and Veggie should be modified (unnecessary work);

Prod_ID	Prod_Name	Prod_Price	Supp_Name	Supp_Contact	Supp_Phone
P211	Apples	2625	Fruit and Veggie	Ines Bond	(+237) 6542 98001
P121	Onions	1575	Fruit and Veggie	Ines Bond	(+237) 6542 98001

d)

3NF
SupplierContact(Supp_Phone, Supp_Contact, Supp_ID*) Supplier(Supp_ID, Supp_Name) Product(Prod_ID, Prod_Name, Prod_Price, Supp_ID*)

QUESTION 6

(i) Bubble sort in descending order:

✓ Pass 1:

5	1	4	2	8

Swap 5 and 1

1	5	4	2	8

Swap 5 and 4

1	4	5	2	8

Swap 5 and 2

1	4	2	5	8

No swap between 5 and 8

1	4	2	5	8

✓ Pass 2:

No swap between 1 and 4

1	4	2	5	8

Swap 4 and 2

1	2	4	5	8

No swap between 4 and 5

1	2	4	5	8

No swap between 5 and 8

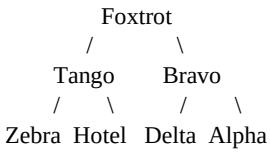
1	2	4	5	8

✓ Sorted

1	2	4	5	8

(ii) To create a balanced binary Search tree that yields the inorder traversal output of ***Zebra, Tango, Hotel, Foxtrot, Delta, Bravo, Alpha.*** Since the middle element is Foxtrot, it is therefore the root node.

The Balanced Search Tree:



(iii) To determine if the name "Tango" is in the binary search tree, you would follow these steps:

1. **Start at the Root**: Begin at the root node.
2. **Compare Values**: Compare "Tango" with the current node's value.
   - If they match, "Tango" is found.
   - If "Tango" is less, move to the left child; if greater, move to the right.
3. **Repeat**: Continue until you find "Tango" or reach a null reference, indicating it is not in the tree.

(iv)

a) The value of fun(1216)

Call	N	N mod 10	N div 10	Result
fun(1216)	1216	6	121	6 + fun(121)
fun(121)	121	1	12	6 + 1 + fun(12)
fun(12)	12	2	1	6 + 1 + 2 + fun(1)
fun(1)	1	1	0	6 + 1 + 2 + 1 + fun(0)
fun(0)	0	0	0	6 + 1 + 2 + 1 + 0 = 10

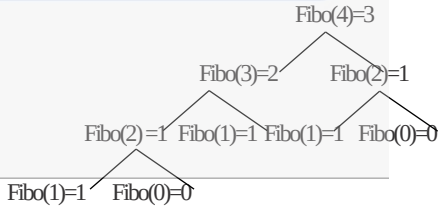
b) To calculate the sum of the digits of a non-negative integer.
Example fun(1234) = 1+2+3+4 = 10

QUESTION 7

(i)

a) 8^{th} term = 21.

b) Fibo(4) = 3



c)

```
Function fibo (Integer n)
    Declare Integer Array f[n+2]
    Declare Integer i, f0, f1, fn

    Assign f0 = 0
    Assign f1 = 1
    For i = 2 to n
        Assign fn = f0 + f1
        Assign f0 = f1
        Assign f1 = fn
    End
Return Integer f1
```

d)

```
FUNCTION Main
    counter = 0
    Output "Enter the value of n:"
    Input n
    Output "Fibonacci Series:"
    FOR i = 1 to n
        Output fibo(counter)
        counter = counter+1
    ENDFOR
ENDFUNCTION

FUNCTION fibo (Integer n)
    IF (n==0 OR n==1) THEN
        Return n
    ELSE
        Return fibo(n-1) + fibo(n-2)
    ENDIF
ENDFUNCTION
```

(ii)

a)

PIN	C	x	Message
51020	0	51020	
	1	5102	
	2	510.2	
	3	51.02	
	4	5.102	
	5	0.5102	"PIN OK"
5120	0	5120	
	1	512	
	2	51.2	
	3	5.12	
	4	0.512	"error in PIN entered"

- **51020**:
  value of c: 5

  message: PIN OK

- **5120**:
  value of c: 4

  message: error in PIN entered

b) Length check

(iii)

line 6: this should read **h = x**
line 8: **PRINT h** should come after the end of the repeat loop
line 9: this should read **UNTIL c = 20 or UNTIL c >= 20 or UNTIL c > 19**

QUESTION 8

(i)

a) Advantages of Surveys:

1. Reach Many People: Surveys can easily get opinions from a large group of stakeholders.
2. Anonymity: People may share honest feedback without fear of judgment.
3. Easy to Analyze: The data collected can be measured and evaluated easily.

Disadvantages of Surveys:

1. Shallow Insights: Surveys may miss deeper details about needs.
2. Bias: Answers may be influenced by how questions are phrased.
3. Low Participation: Some people may not want to fill out surveys.

b)

1. Interviews
2. Focus Groups
3. Workshops

c) Advantages of Prototypes:

1. Get Feedback: Prototypes let users share their thoughts early.
2. Clearer Ideas: They help everyone understand how the system will work.
3. Find Problems Early: Issues can be spotted before full development.
4. Better Communication: Prototypes help discussions among users and developers.

d) More cycles of analysis and design may be needed to refine requirements and adapt to changes. This helps ensure the system meets user needs and improves the final product by catching issues early.

e) Thorough testing is essential to make sure the system works correctly and safely. It helps find problems, increases user satisfaction, and reduces risks, ensuring a smooth launch and good user experience.

(ii) **Suitable test data**

Type	Example test value	Explanation	Expected Results
Normal	12	Positive integers from 1 to 20	Passed
	8		Failed
Abnormal	Ten 12.4	Any different data type	Error message or Runtime error
Boundary	20, 10	The minimum Pass mark is 10, hence the boundary values are 10 and 20	10: Failed 20: Passed

DATABASE: HOTEL MANAGEMENT SYSTEM

1) CREATE DATABASE HDB;

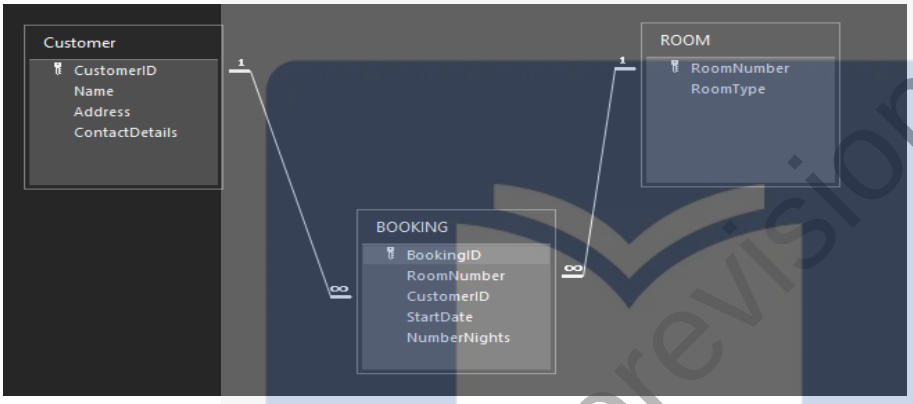
2) CREATION OF TABLES USING SQL:

a) CREATE TABLE CUSTOMER
(
CustomerID VARCHAR(5) PRIMARY KEY,
Name VARCHAR(20) NOT NULL,
Address VARCHAR(20) NOT NULL,
ContactDetails INT NOT NULL
);

b) CREATE TABLE ROOM
(
RoomNumber SMALLINT PRIMARY KEY,
RoomType VARCHAR(6) NOT NULL
);

c) CREATE TABLE BOOKING
(
BookingID VARCHAR(5) PRIMARY KEY,
RoomNumber SMALLINT NOT NULL,
CustomerID VARCHAR(5) NOT NULL,
StartDate DATE NOT NULL,
NumberNights SMALLINT NOT NULL,
FOREIGN KEY (RoomNumber) REFERENCES ROOM(RoomNumber),
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

3) Relational Diagram



4) Records for table ROOM:

```
INSERT INTO ROOM(RoomNumber,RoomType)
VALUES(1001, 'Family'),(1002, 'Family'),(1003, 'Double');
```

5) Table contents:

ROOM

RoomNumb	RoomType
1001	Family
1002	Family
1003	Double

CUSTOMER

CustomerID	Name	Address	ContactDetails
AB101	Jean Claude	Parliament, Bonas	679457812
XY102	Marie Ngome	Carriere Bar, Mewoul	651457894
MA103	Yvan Catanou	ASSEC, Centre	681245348

BOOKING

BookingID	RoomNumber	CustomerID	StartDate	NumberNights
001XY	1001	AB101	12-Jan-25	3
002XY	1001	AB101	15-Feb-25	3
003XY	1002	MA103	15-Jan-25	5

6) SQL QUERY

```
SELECT CUSTOMER.Name, ROOM.RoomNumber, ROOM.RoomType, BOOKING.StartDate
FROM ROOM, CUSTOMER, BOOKING
WHERE BOOKING.StartDate=#12/12/2024# AND CUSTOMER.CustomerID = BOOKING.CustomerID AND
ROOM.RoomNumber = BOOKING.RoomNumber;
```

7) SQL QUERY:

```
SELECT COUNT(CUSTOMER.CustomerID) AS [Number of Customers]
FROM CUSTOMER, BOOKING, ROOM
WHERE ROOM.RoomType = 'Double' AND CUSTOMER.CustomerID = BOOKING.CustomerID AND
ROOM.RoomNumber = BOOKING.RoomNumber;
```

PROGRAMMING: ROLL OF DICE GAME

1) Return values:
   - DuplicateNum (DICEDIAL,6): 0
   - DuplicateNum (DICEDIAL,5): 1
   - DuplicateNum (DICEDIAL,1): 4

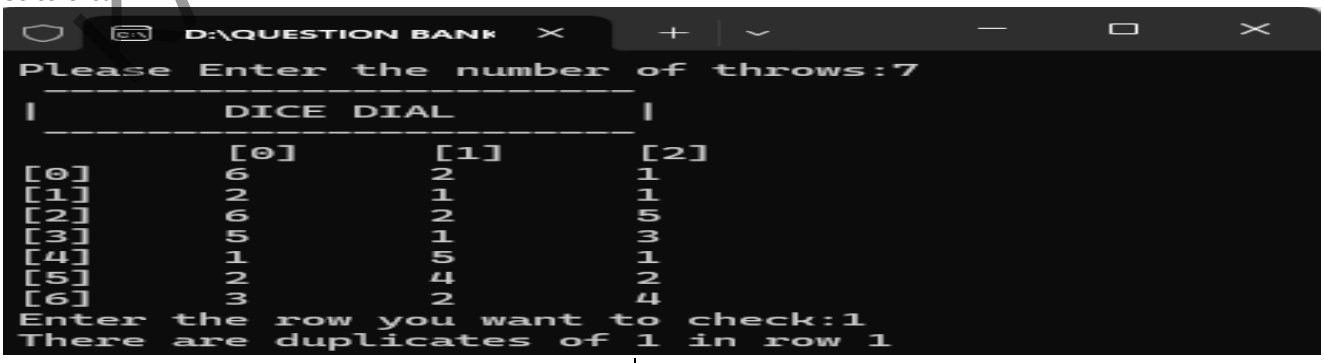
2)
```
void RandomDial(int DICEDIAL[][3], int N)
{
	int i, j, face;

	//throws
	for(i=0;i<N;i++)
	{
		//dice
		for(j=0;j<3;j++)
		{
			//face of dice
			face = rand() % 6 +
1;
			DICEDIAL[i][j] =
face;
		}
	}
}
```

3)
```
void displayDiceDial(int DICEDIAL[][3],
int N)
{
	int i, j;
	printf(" ------------------------
\n");
	printf("|\tDICE DIAL\t|\n");
	printf(" ------------------------
\n");
	printf("\t[0]\t[1]\t[2]\n");
	//throws
	for(i=0;i<N;i++)
	{
		//dice
		printf("[%d]\t",i);
		for(j=0;j<3;j++)
		{

	printf("%d\t",DICEDIAL[i][j]);
		}
		printf("\n");
	}
}
```

4)
```
void DuplicateNum (int DICEDIAL[][3], int
R)
{
	int VAL = 0;
	if (DICEDIAL[R][0] ==
DICEDIAL[R][1])
		VAL = DICEDIAL[R][0]; // or
VAL = DICEDIAL[R][1]
	else
	if (DICEDIAL[R][0] ==
DICEDIAL[R][2] )
			VAL = DICEDIAL[R][0];
	else
		if (DICEDIAL[R][1] ==
DICEDIAL[R][2])
			VAL = DICEDIAL[R][1];

	//check if found
	if (VAL == 0)
		printf("No duplicate in row
%d\n",R);
	else
		printf("There are duplicates
of %d in row %d\n",VAL, R);
}
```

5)
```
int main()
{
	int DICEDIAL[100][3], N, R;

	srand(time(NULL));
	printf("Please Enter the number of
throws:");
	scanf("%d",&N);
	RandomDial(DICEDIAL, N);
	displayDiceDial(DICEDIAL, N);
	printf("Enter the row you want to
check:");
	scanf("%d",&R);
	DuplicateNum(DICEDIAL, R);

	getch();
	return 0;
}
```

6) Screenshot1



7)
```
void addThrowDice(int DICEDIAL[][3], int N)
{
	int i, j, sum;
	//summing the faces of each throw
	for(i=0;i<N;i++)
	{
		sum = 0;
		for(j=0;j<3;j++)
		{
			sum = sum + DICEDIAL[i][j];
		}
		sumDice[i] = sum;
	}
	//printing the content of array sumDice
	printf("\nThrows\t");
	for(i=0;i<N;i++)
	{
		printf("%d\t",i+1);
	}
	printf("\nSum\t");
	for(i=0;i<N;i++)
	{
		printf("%d\t",sumDice[i]);
	}
}
```

8)
```
void lowestRT(int N)
{
	int i, lowest = 100;
	//check which is the lowest sum in sumDice
	for(i=0;i<N;i++)
	{
		if (sumDice[i]<lowest)
		{
			lowest = sumDice[i];
		}
	}
	//print the least sum
	printf("\n\nThe lowest sum is %d\n",lowest);
	//positions of the least in sumDice array
	printf("Occurs in throw:\n");
	for(i=0;i<N;i++)
	{
		if (sumDice[i] == lowest)
		{
			printf("%d\n",i+1);
		}
	}
}
```

9) screenshot2