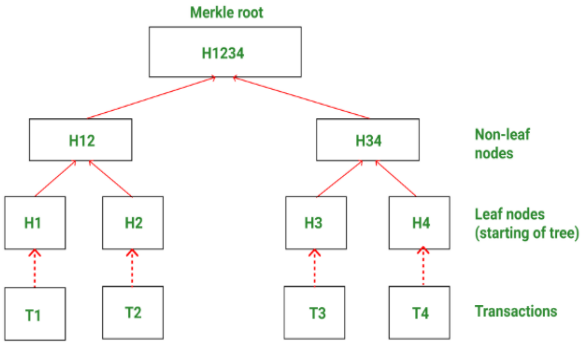
(Affiliated to University of Mumbai)	***End Semester Examination (R-24/R-25) SH 2026 Answer Key with marking scheme***
Branch:CSE (IoT &CSIBCT)	Course: Introduction to Blockchain Technology
Year/ Semester: TE/V	Course code: CSEC503
Time: 03 hours	Marks: 80

		Marks
Q. 1	All questions are compulsory.	
A.	**Block Height & Its Significance (2 Marks)** Definition (1 mark): Block height is the number of blocks preceding a particular block in a blockchain, measured from the genesis block. The genesis block has a height of 0. Significance (1 mark): - It indicates the position of a block in the blockchain. - It helps identify blocks and determine their order in the chain. **Genesis Block and its Significance (3 Marks)** Definition (2 mark): The genesis block is the first block of a blockchain. It is created when the blockchain is initialized and has no preceding block. Its block height is 0. It has no previous block hash linking it to an earlier block. Subsequent blocks are linked to it directly or indirectly through their previous block hashes. Significance (1 mark): - It serves as the starting point of the blockchain. - It establishes the foundation for linking and ordering subsequent blocks.	05
B.	**Concept of Blockchain (1Marks)** Blockchain is a peer-to-peer, distributed ledger that is cryptographically secure, append-only, immutable (extremely hard to change), and updateable only via consensus — agreement among peers. **2. Benefits of Blockchain (2 Marks)** **Any 8 points** 1. Decentralization: Data is distributed across multiple nodes without a central controlling authority. 2. Enhanced security: Cryptography and consensus mechanisms help protect records against unauthorized modification. 3. Transparency: Transactions are visible to participants according to the blockchain's access rules. 4. Immutability: Once recorded, information is extremely difficult to change or delete. 5. Traceability: Transactions are chronologically linked, allowing assets to be tracked from origin to destination. 6. Faster transactions: Peer-to-peer transactions can reduce dependence on intermediaries. 7. Lower transaction cost: Automation and reduced intermediary involvement can lower costs. 8. High availability and fault tolerance: Other nodes can continue operating when some nodes fail. 9. Automation using smart contracts: Predefined conditions can trigger automatic execution.	05

	10. Auditability: Time-stamped and chronologically ordered records support auditing. **3. Limitations of Blockchain (2 Marks)** **Any 8 points** 1. Complexity: Blockchain can be difficult to understand and implement. 2. High initial investment: Development and maintenance require specialized expertise. 3. High energy consumption: Proof of Work blockchains can consume substantial electricity. 4. High storage requirements: Full nodes must store the blockchain ledger. 5. High transaction fees: Network congestion can increase transaction fees. 6. Privacy concerns: Blockchain addresses may sometimes be linked to real identities. 7. Scalability issues: Some blockchain networks process fewer transactions per second than centralized databases. 8. Slow transaction speed: Transactions may require considerable time to achieve sufficient confirmations. 9. Irreversible transactions: Incorrectly sent transactions are generally difficult to reverse. 10. Smart contract bugs: Programming errors can cause unintended transfers or permanently locked funds.	
C.	**Structure and diagram (2 Mark)**  **Working and generation of Merkle root (2 Marks)** Step 1: Generate individual transaction hashes.H1,H2,H3,H4 Step 2: Combine pairs of hashes to generate parent hashes.H12,H34 Step 3: Combine the parent hashes to obtain the Merkle root.H1234 **Significance (1 Mark)** If a transaction is modified, its hash changes, which propagates through the parent hashes and changes the Merkle root. Thus, the root helps detect data tampering and supports efficient verification of transaction integrity.	05
D.	**1. Initial Coin Offering (ICO) — 1 Mark** An Initial Coin Offering (ICO) is a fundraising method in which a project issues and sells cryptocurrency tokens to investors to raise funds for developing a blockchain-based project. **2. Security Token Offering (STO) — 1 Mark**	05

A Security Token Offering (STO) is a fundraising method in which security tokens representing investment interests, such as ownership, profit-sharing or other financial rights, are issued to investors, subject to applicable securities regulations.

Differences between ICO and STO — 3 Marks

Basis	ICO	STO
Token type	Often utility or project tokens	Security tokens representing
Purpose	Primarily to raise funds for a blockchain project	To raise funds through regu offerings
Regulation	Regulatory treatment varies depending on the token and jurisdiction	Generally subject to applica regulations
Investor rights	Depend on the token's design and terms	May include financial or ow rights
Compliance	Requirements vary by jurisdiction and token classification	Typically involves more exte compliance

E. **Any 5 points:1 Marks each** — 05

Basis	Externally Owned Account (EOA)	Contract Account (CA)
1. Control	Controlled by a private key.	Controlled by the logic of its smart contract code.
2. Code	Does not have associated smart contract code.	Has smart contract code stored on the blockchain.
3. Transaction initiation	Can initiate transactions.	Cannot initiate transactions independently.
4. Execution	Can send transactions to transfer Ether or invoke contracts.	Executes its code when triggered by a transaction or a message from another contract.
5. State and balance	Has an Ether balance and account state.	Has an Ether balance, account state and contract storage.
6. Address creation	Derived from the public key.	Generated when the contract is deployed.

F. **Role of Endorsement Policies:2 marks** — 05

An endorsement policy in Hyperledger Fabric specifies which organizations' peers must approve a transaction for it to satisfy the required endorsement conditions

1. They specify which organizations must endorse a transaction.
2. They ensure that the required organizations approve the transaction before it can be committed.

How Endorsement Policies Determine Transaction Validity:3 Marks

1. Proposal: The client sends a transaction proposal to the required endorsing peers.
2. Simulation: The endorsing peers execute the chaincode and generate a transaction response without updating the ledger.
3. Endorsement: The endorsing peers sign their transaction responses and return them to the client.

4. Validation: The committing peers check whether the required endorsements and signatures are valid. They also check whether the transaction's read-set is consistent with the current ledger state.
5. Commit: If all required checks pass, the transaction is marked valid and updates the world state. Otherwise, it is marked invalid and does not update the world state.

Q.2A

Comparison of consensus mechanisms (6Marks) 10

Parameter	PoW	PoS	DPoS	PoET
Validator selection	Miners compete to solve a cryptographic puzzle. The first to solve it earns the right to propose a block.	A validator is selected using factors such as stake and, age of staked coins randomness etc.	Token holders vote to elect a limited number of delegates who produce blocks in a scheduled sequence.	Each node receives a random waiting time from a Trusted Execution Environment (TEE). The node whose timer expires first proposes a block.
Security assumptions	Security depends on the difficulty of the puzzle and the cost of controlling sufficient mining power.	Validators have cryptocurrency at stake, which can be penalized through slashing for dishonest behaviour.	Security depends on honest elected delegates, fair elections and the ability of token holders to replace misbehaving delegates.	Security depends on the trustworthiness of the TEE and the validity of its attestation.
Energy consumption	High, due to continuous computational mining.	Low, because competitive mining is not required.	Low, because a limited group of delegates produces blocks without competitive mining.	Low, because nodes wait instead of performing energy-intensive mining.
Scalability	Generally lower transaction throughput due to mining competition and block	Generally better than PoW, although performance depends on the protocol and validator participation.	High, because a small, predictable group of delegates enables faster consensus and higher	Efficient block production because nodes wait for timers rather than solve computational puzzles.

	confirmation delays.		transaction throughput.	
Decentralization	Can be highly decentralized, but mining pools and expensive hardware may concentrate power	Can support many validators, but large stakes may concentrate influence.	Lower decentralization because block production is controlled by a limited number of elected delegates.	Depends on the number and distribution of participating nodes and trust in the TEE technology.

Suitable applications for each mechanism (4 Marks)

1. Proof of Work (PoW) — 1 Mark
Suitable application: Bitcoin cryptocurrency. PoW is suitable for public, permissionless blockchain networks where miners compete to solve computational puzzles to earn the right to add blocks.
2. Proof of Stake (PoS) — 1 Mark
Suitable application: Ethereum. PoS is suitable for public blockchain networks requiring energy-efficient validation, where validators stake cryptocurrency to participate in consensus.
3. Delegated Proof of Stake (DPoS) — 1 Mark
Suitable application: EOS or BitShares. DPoS is suitable for blockchain networks requiring fast transaction processing, where token holders elect a limited number of delegates to produce blocks.
4. Proof of Elapsed Time (PoET) — 1 Mark
Suitable application: Permissioned blockchain networks using Hyperledger Sawtooth. PoET is suitable for networks with known participants where energy-efficient block selection can be achieved using randomized waiting times generated by a Trusted Execution Environment (TEE).

OR

Q.2B. (i) — 05

Parameter	RAFT	PAXOS
Leader election	A follower becomes a candidate when it does not receive heartbeats. It requests votes from other servers and becomes leader after receiving majority votes.	In Basic Paxos, multiple proposers can compete; there is no permanent leader. In Multi-Paxos, a leader is used to reduce conflicts.
Agreement among nodes	The leader replicates log entries to followers. An entry is committed when it is replicated on a majority of servers, subject to Raft's commitment rules.	Proposers send numbered proposals to acceptors. A proposal is accepted through a majority quorum, and learners are informed of the agreed value.
Handling node failures	If the leader fails, followers detect missing heartbeats and start a new election. The system can continue if a majority of servers are available.	Paxos can continue despite node or message failures if a majority of acceptors remains available. Proposals may be retried, and competing proposers can cause repeated retries.

	Leader Election:2 Marks **Agreement among Nodes:2 Marks** **Handling Node Failures:1 Marks**	
Q.2B(ii)	**1. Need for consensus mechanisms (1 Mark)** A consensus mechanism enables distributed blockchain nodes to agree on a common, valid version of the ledger without relying on a central authority. It maintains data integrity, prevents fraud and handles faulty or malicious nodes. **2. Byzantine Generals Problem (2 Marks)** The Byzantine Generals Problem describes the difficulty of reaching a common decision when some participants may send false or conflicting messages.In a blockchain, some nodes may behave maliciously or provide incorrect information. Consensus mechanisms help honest nodes agree on the same valid state despite such behaviour. Byzantine Fault Tolerant mechanisms, such as PBFT, are specifically designed to handle Byzantine faults. **3. Double-spending problem (2 Marks)** Double spending occurs when the same digital currency is spent more than once. Blockchain nodes verify transaction history and check whether the sender has sufficient unspent funds. The consensus mechanism ensures that only one of two conflicting transactions is accepted into the agreed ledger. Waiting for sufficient block confirmations reduces the risk of accepting a transaction that may later be reversed.	05
Q.3A	**Non-Fungible Tokens (NFTs) — 1 Marks** A Non-Fungible Token (NFT) is a unique digital token used to identify a particular digital or physical asset on a blockchain. Each NFT is individually identifiable and cannot be exchanged on a one-to-one basis as an identical unit. Examples include digital artwork, collectibles, gaming items, concert tickets and numbered seats. **ERC-721 Standard — 1 Mark** ERC-721 is an Ethereum token standard used to create and manage non-fungible tokens. Each NFT has a unique tokenId, and the combination of the smart contract address and tokenId uniquely identifies the NFT. **ERC-721 Functions and Their Purposes — 6 Marks** **ERC-721 Events and Their Purposes — 2 Marks**	10

Function or event	Purpose
`balanceOf()`	Returns the number of NFTs owned by an address.
`ownerOf()`	Returns the owner of a specific NFT using its `tokenId`.
`transferFrom()`	Transfers a specific NFT from one address to another.
`safeTransferFrom()`	Transfers an NFT after checking whether the receiving smart contract can receive NFTs; otherwise, the transaction is reverted.
`approve()`	Approves another address to transfer a specific NFT.
`setApprovalForAll()`	Authorizes another address to manage all NFTs owned by the user.
`getApproved()`	Returns the address approved to transfer a particular NFT.
`isApprovedForAll()`	Checks whether an operator is authorized to manage all NFTs of an owner.
`tokenURI()`	Returns the location of the NFT's metadata.

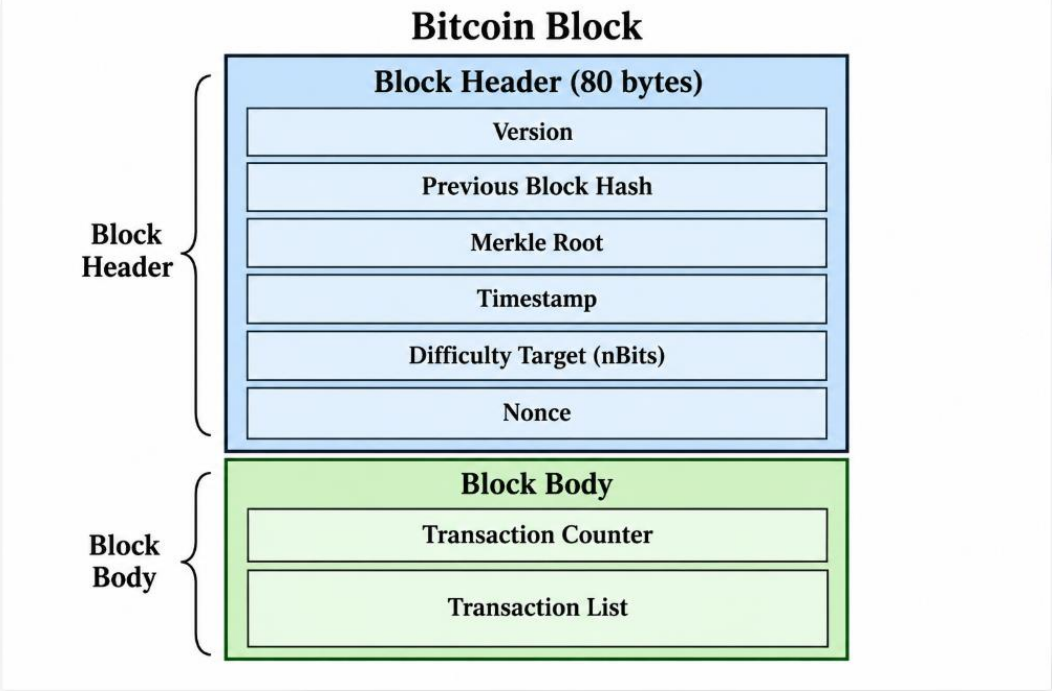
Important events:

- `Transfer`: Generated whenever NFT ownership changes.
- `Approval`: Generated when an owner approves an address to transfer a specific NFT.
- `ApprovalForAll`: Generated when an owner authorizes an operator to manage all their NFTs.

Metadata extension: name(), symbol() and tokenURI() provide the NFT collection's name, symbol and metadata location.
Enumeration extension: totalSupply(), tokenByIndex() and tokenOfOwnerByIndex() help list NFTs in a collection or those owned by a particular address.

OR

Q.3B **Diagram :2 Marks** 10



Explanation:8Marks

Block Header:6 Marks

Bytes	Name	Data Type	Description
4	version	int32_t	The block version number indicates which set of block validation rules to follow. See the list of block versions below.
32	previous block header hash	char[32]	A SHA256(SHA256()) hash in internal byte order of the previous block's header. This ensures no previous block can be changed without also changing this block's header.
32	merkle root hash	char[32]	A SHA256(SHA256()) hash in internal byte order. The merkle root is derived from the hashes of all transactions included in this block, ensuring that none of those transactions can be modified without modifying the header. See the merkle trees section below.
4	time	uint32_t	The block time is a Unix epoch time when the miner started hashing the header (according to the miner). Must be strictly greater than the median time of the previous 11 blocks. Full nodes will not accept blocks with headers more than two hours in the future according to their clock.
4	nBits	uint32_t	An encoded version of the target threshold this block's header hash must be less than or equal to. See the nBits format described below.
4	nonce	uint32_t	An arbitrary number miners change to modify the header hash in order to produce a hash less than or equal to the target threshold. If all 32-bit values are tested, the time can be updated or the coinbase transaction can be changed and the merkle root updated.

The hashes are in internal byte order; the other values are all in little-endian order.

Explain each field in detail.

Block body:2 Marks

The block body contains the actual transactions included in the block.

1. Transaction counter: Indicates how many transactions are included in the block.
2. Transaction list: Contains the serialized transaction data. It includes the coinbase transaction, which creates the block subsidy and can collect transaction fees according to Bitcoin's rules. Other transactions transfer bitcoins by spending previously created transaction outputs.The transactions are represented in the header indirectly through the Merkle root, rather than by storing the entire transaction list in the header.

Q.4A

A decentralized application (DApp) uses blockchain and related technologies to provide services without depending entirely on a central server. A DApp may require decentralized storage, peer-to-peer communication, and external data to function effectively. Swarm and IPFS support decentralized file storage and sharing, Whisper supports peer-to-peer messaging, and oracles provide external information to smart contracts.

A. Swarm (2 marks)

Role: Swarm is a decentralized storage and content-distribution system associated with the Ethereum ecosystem. It allows DApps to store and retrieve files, such as images, documents and website content, across a distributed network rather than relying on a single centralized server.

- Supports distributed storage and content delivery.
- Helps DApps serve website files and other digital content.
- Reduces dependence on a single hosting server.

Example: An NFT marketplace can store artwork images using Swarm and link the artwork's location or content reference to the NFT metadata.

10

	B. Whisper (2 marks) Role: Whisper was designed as a peer-to-peer messaging protocol for communication between Ethereum application users and nodes. It supports message exchange without requiring all messages to be recorded as blockchain transactions. - Supports communication between peers. - Can provide message confidentiality and filtering based on protocol features. - Is useful for application messages and notifications rather than permanent blockchain state. Example: A decentralized trading application could use Whisper to send a user a notification that an order has been matched. Whisper is an older Ethereum messaging protocol **C. IPFS — InterPlanetary File System (3marks)** Role: IPFS is a distributed system for storing and sharing files. As explained in your uploaded PPT, it uses content addressing rather than relying only on a file's location. It creates a Content Identifier (CID) for content, enabling users to identify and retrieve it through a peer-to-peer network - IPFS nodes: Connect to other nodes, store content, and request or provide files. - Content Identifier (CID): Identifies content rather than its network location. - Content discovery: Mechanisms such as Distributed Hash Tables (DHT) and delegated routing help find peers that provide the content. - Data transfer and pinning: Bitswap can exchange data blocks between peers, while pinning instructs a node to retain particular content locally. Example: A college can upload lecture notes as a PDF to IPFS and share the CID with students. Students can use the CID to retrieve the PDF from available IPFS providers. Pinning the file helps ensure that the pinning node retains it. IPFS does not automatically guarantee permanent availability; the content must remain available from one or more providers. **D. Oracles (3 marks)** Role: Oracles connect smart contracts with data or events originating outside the blockchain. Smart contracts cannot independently access arbitrary external information, so oracles supply external data for contract execution. - External data: Supply information such as market prices, weather conditions and sports results. - Smart-contract execution: Allow a contract to use external information when evaluating its conditions. - Real-world integration: Help blockchain applications interact with external systems and events. - Trust and security: Oracle data must be reliable because incorrect or manipulated inputs can cause incorrect contract outcomes. Example: A decentralised insurance application uses weather data supplied by an oracle. If the verified rainfall data meets a predefined threshold, the smart contract automatically releases the insurance payout to the eligible farmer.	
	OR	
Q.4B	**Definition and objectives of Ethash (2 marks)** Ethash was the Proof-of-Work (PoW) mining algorithm used by Ethereum before its transition to Proof of Stake. It is a modified version of the Dagger-Hashimoto algorithm. Ethash was designed to be memory-hard, meaning that mining requires substantial memory access in addition to computation. Its main objectives were:	10

- ASIC resistance: Reduce the advantage of specialized Application-Specific Integrated Circuit (ASIC) hardware and make GPU mining more practical.
- Light-client verification: Enable devices with limited memory to verify a mining result without storing the entire mining dataset.
- However, ASIC resistance was not permanent, as specialized Ethash ASICs were eventually developed.

Mining process of Ethash (4 marks)

The steps are explained below,

Step 1: Take the blockchain information. The algorithm starts with information from the blockchain, especially the previous block's information. The seed is used to generate the mining dataset.

Step 2 — Generate a large DAG

Using the seed, Dagger generates a large DAG/dataset

This large dataset is what makes mining memory-intensive.

Step 3 — Miner chooses a nonce

The miner chooses a nonce.

Step 4 —Combine block information + nonce

The miner combines:block header and nonce

This initial hash is used to determine which locations in the DAG to access.

Step 5 — Access the DAG

The resulting value tells the miner which DAG location to read.

The miner reads that value.Then the result is updated and another location is selected: This is repeated many times.

Step 6 — Mix the retrieved values

The miner combines the values obtained from the DAG.

Step 7 — Calculate the final hash

Step 8 — Check the difficulty

The miner checks:Is the final hash below the required difficulty target? f not, the miner tries another nonce. If valid, the block is broadcast for verification by other nodes.

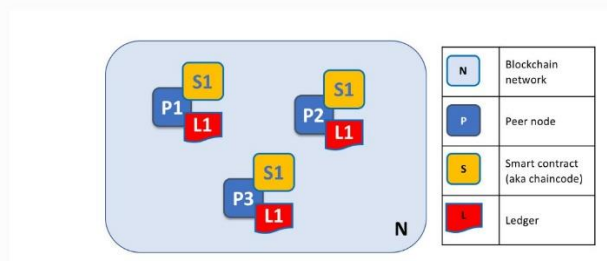
3. Role of Ethash in Proof of Work (2 marks)

- Block creation: Miners competed to find a valid Proof of Work before proposing a block.
- Network security: The computational and memory resources required for mining made it costly to repeatedly produce competing blocks.
- Consensus: The PoW mechanism helped nodes agree on the accepted blockchain history.
- Independent verification: Other nodes could check the submitted proof and validate the block before accepting it.

4. Historical significance (2 marks)

- Ethash was used to secure Ethereum Mainnet during its original PoW era.
- Its memory-hard design aimed to reduce specialized-hardware advantages, although ASIC miners eventually emerged.
- On 15 September 2022, Ethereum completed The Merge, transitioning from PoW to PoS.
- Ethereum Mainnet no longer uses Ethash mining. It is now secured by validators participating in PoS consensus.

Q.5A	Hyperledger Fabric uses an execute–order–validate architecture. Unlike systems that order transactions before executing them, Fabric first simulates and endorses a transaction, then orders it into a block, and finally validates it before updating the ledger. **Stage 1: Transaction proposal submission(1.5 Marks)** 1. A client application initiates a transaction by invoking a chaincode function, such as transferAsset(), with the required arguments. 2. The application uses its Fabric identity and cryptographic credentials to create and sign the transaction proposal. 3. The proposal is sent to the Fabric Gateway or a peer, depending on the client API and network configuration. 4. The Gateway directs the proposal to the peers required by the chaincode's endorsement policy. Example: A company requests a transfer of ownership of an asset from one participant to another. **Stage 2: Endorsement and transaction simulation(2 Marks)** 1. The endorsing peers verify the proposal's format, client signature, identity and authorization. 2. Each peer executes the requested chaincode function against its current ledger state. This is a simulation of the proposed transaction. 3. The peer produces a response containing the result, a read set (the state values and versions read) and a write set (the proposed state changes). 4. The peer signs the proposal response, creating an endorsement, and returns it to the Gateway or client. Important: The ledger is not updated at this stage. Peers simulate the transaction and produce proposed changes; they do not yet commit those changes. **Stage 3: Checking endorsements and assembling the transaction(1.5 Marks)** 1. The Gateway collects endorsement responses from the required peers. 2. The application or Gateway checks that the responses are consistent and that the required endorsement policy is satisfied. 3. The endorsed proposal and its read/write sets, together with the required signatures and channel information, are assembled into a signed transaction envelope. 4. The client submits the signed transaction to the ordering service. For example, an endorsement policy may require approval from peers belonging to both the buyer's and seller's organizations. The exact requirement depends on the network's policy. **Stage 4: Transaction ordering and block creation(2 Marks)** 1. The ordering service receives endorsed transactions from applications. 2. It establishes a consistent order for transactions on the channel. 3. Transactions are grouped into blocks according to the ordering service's configuration. 4. The blocks are distributed to the peers on that channel. The ordering service establishes transaction order; it does not execute the chaincode or make the final decision about whether each transaction is valid for updating world state. **Stage 5: Transaction validation(1.5 marks)** Each peer independently validates transactions in the received block. The checks include: - Endorsement-policy validation: Are the required organizations or peers represented by valid endorsements?	10

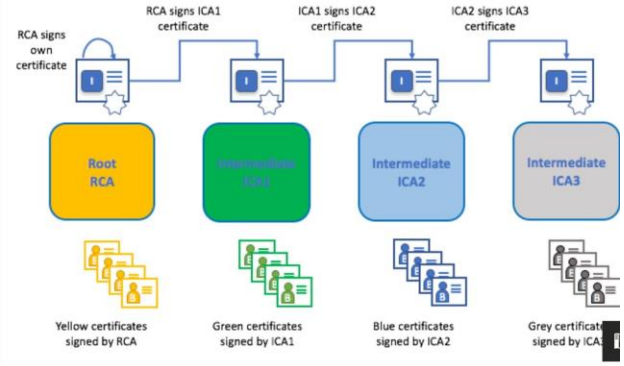
	- Endorsement consistency: Do the endorsed simulation results and read/write sets match as required? - Version validation (MVCC): Have the versions of the state entries in the read set remained unchanged since simulation? If another transaction modified a relevant value first, the transaction may be invalidated. Each transaction is marked valid or invalid. A transaction can therefore be included in an ordered block but still fail validation. **Stage 6: Commitment and client notification(1.5 marks)** 1. Each peer appends the received block to its channel's blockchain ledger, recording the transactions and their validation status. 2. For valid transactions, the peer applies the write set to the world state database. 3. Invalid transactions remain recorded in the block with an invalid status, but their proposed state changes are not applied. 4. The peer or Gateway provides a commit-status event so that the client application can determine whether the transaction was valid or invalid.	
	OR	
Q.5B	**Peers(2 Marks)** A peer is a computer (node) in a Hyperledger Fabric network that maintains the ledger and runs smart contracts.Imagine two companies, a manufacturer and a retailer, sharing a blockchain to track products.Each company may operate its own peer. Peers are fundamental because they manage two important things: Ledger — the information The ledger records the transaction history. Example: Product created by the manufacturer,Product transferred to the retailer.,Product received by the retailer. The ledger helps participants track what happened. Smart contracts — the processes Smart contracts contain the business rules. For example : A product can be transferred only by its current owner. In Fabric, smart contracts are contained in chaincode.  Peers are a flexible and redundant element that can be created, started, stopped, reconfigured and deleted. Peers expose a set of APIs that enable client applications to interact with the services that peers provide — the Fabric Gateway service in particular. **Membership Service Provider(1 Mark)** For an identity to be verifiable, it must come from a trusted authority. A membership service provider (MSP) is that trusted authority in Fabric. More specifically, an MSP is a component that defines the rules that govern the valid identities for this organization. The default MSP implementation in Fabric uses X.509 certificates as identities, adopting a traditional Public Key Infrastructure (PKI) hierarchical model.	10

Certificate Authority(2 Marks)



A Certificate Authority dispenses certificates to different actors. These certificates are digitally signed by the CA and bind together the actor with the actor's public key. As a result, if one trusts the CA (and knows its public key), it can trust that the specific actor is bound to the public key included in the certificate, and owns the included attributes, by validating the CA's signature on the actor's certificate.

Root CAs, Intermediate CAs and Chains of Trust-Explain this diagram



Diagrams are not madndatory.But Concepts shown in picture should be explained properly.

Client Application(1 Mark)

Acts on behalf of the user or business. It creates and signs transaction proposals, requests endorsement, submits endorsed transactions for ordering and receives transaction commit-status notifications. A client application could be:

- A web application
- A mobile application
- An enterprise application
- A banking application

Orderer (1.5 Marks)

An orderer is a node participating in Fabric's ordering service.Its major responsibility is to establish an agreed order of transactions and create blocks from those transactions. It Receive endorsed transactions, establish their order and package them into blocks. They distribute blocks to channel peers. They do not execute chaincode or independently determine the final validity of each transaction.

Endorsers (1.5Marks)

Endorsers are peers that simulate a proposed transaction by executing chaincode, generate the read/write set and digitally sign the proposal response. The required endorsements must satisfy the chaincode's endorsement policy.

Committers (1 Mark)

The committing function is performed by peers. Each peer validates transactions in the received block, marks them valid or invalid, appends the block to its ledger and applies write sets of valid transactions to world state. "Committer" is a functional role, not necessarily a separate node type.

Q.6A	**Case Study: 5 Marks** Walmart explored blockchain technology to improve food traceability, particularly for products such as mangoes and pork. Traditional food supply chains involve multiple stakeholders, including farmers, processors, distributors, transporters and retailers. Their records may be stored in separate systems, making it difficult to trace a product's origin quickly during contamination or food-safety incidents. Walmart collaborated with IBM on a blockchain-based food-tracking initiative. The objective was to create a shared, tamper-evident record of food's journey from its source to the retail stage. Walmart's solution used IBM's blockchain platform, built on Hyperledger Fabric, to share traceability records among authorized supply-chain participants. The process works as follows: 1. Data recording: Each participating organization records relevant information about the product or batch as it moves through the supply chain. 2. Shared, tamper-evident history: Blockchain links transactions into a ledger that makes unauthorized alteration of recorded history detectable and helps participating organizations rely on a common record. 3. Rapid tracing: When a food-safety problem occurs, Walmart can use the recorded information to identify the product's origin and movement, helping narrow the scope of an investigation or recall. Case-study result: Walmart reported reducing the time required to trace the origin of mangoes from approximately seven days to 2.2 seconds. In a separate pilot in China, pork products could be traced to their farm of origin using a QR code. **Challenges and critical analysis (5 marks)** A. Data accuracy and reliability (2 marks) Blockchain protects recorded history, but it cannot automatically verify that the information entered is true. - Incorrect input: A supplier could enter a false farm location, harvest date or inspection result. - Manual errors: Incorrect batch numbers or incomplete records can create gaps in the traceability chain. - Physical–digital link: A QR code or product identifier must remain correctly associated with the actual food batch. Otherwise, accurate blockchain records may refer to the wrong product. - Possible solutions: Validate entries against inspection certificates, use authenticated sensors and scanning systems, perform independent audits and cross-check records between supply-chain participants. B. Scalability and integration (1.5 marks) Walmart handles large volumes of food products across multiple locations and organizations. - High transaction volume: Recording every batch movement and inspection can generate large numbers of transactions. - System integration: Suppliers may use different inventory, logistics and tracking systems. Integrating these with a shared blockchain requires time, money and technical expertise. - Data storage and performance: Detailed records, sensor data and supporting documents can increase system and infrastructure demands.	10

	Possible solutions: Use batch-level tracking, standardized data formats, efficient network configurations and off-chain storage for large files, while recording verifiable references on the blockchain. C. Stakeholder participation (1.5 marks) Blockchain traceability depends on cooperation among farmers, processors, transporters, distributors and retailers. - Adoption cost: Smaller suppliers may lack the money, infrastructure or expertise to implement the system. - Reluctance to share information: Businesses may be unwilling to disclose commercially sensitive information about suppliers, costs or operations. - Inconsistent participation: If a participant fails to record a transfer or inspection, the shared history becomes incomplete. - Possible solutions: Provide training, affordable onboarding, simple data-entry tools, clear data-sharing policies and incentives for suppliers to participate.	
	OR	
Q.6B	**Crossborder Payments using Conventional Payments (1 Mark)** Cross-border payments involve transferring money between parties in different countries through banks, correspondent banks and payment networks. Conventional systems may involve multiple intermediaries, reconciliation processes and delays. Blockchain offers a shared, tamper-evident ledger that can help financial institutions coordinate transactions and potentially reduce processing time and reconciliation costs. **Transaction throughput (1.5 marks)** Benefits: - Blockchain allows participating financial institutions to share transaction records and reduce duplicate processing. - A suitably designed blockchain can process transactions efficiently, depending on its consensus mechanism and network configuration. Limitations: - Transaction throughput depends on the blockchain architecture, consensus protocol, transaction complexity and number of participating nodes. - Some blockchain networks may struggle with very high transaction volumes compared with mature, centralized payment infrastructures. **Settlement time (1.5 marks)** Benefits: - Blockchain can support continuous, 24/7 transaction processing, potentially reducing dependence on banking hours and cut-off times. - Shared records can reduce delays caused by reconciliation between intermediaries. - Suitable settlement arrangements may enable near-real-time settlement. Limitations: - Transaction processing does not always mean final settlement. Finality depends on the consensus protocol and the legal and operational arrangements. - Currency conversion, liquidity availability, compliance checks and integration with existing banking systems may still cause delays. **Privacy (1.5 marks)** Benefits: - Permissioned blockchain networks allow access to be restricted to authorized financial institutions.	10

	- Access controls and privacy-enhancing mechanisms can help protect sensitive transaction information. Limitations: - Sharing transaction records across institutions may expose commercially sensitive information if access controls are inadequate. - Even on permissioned networks, transaction metadata may reveal information about business relationships or payment patterns. **Security (1.5 marks)** Benefits: - Cryptographic signatures help verify transaction authorization and integrity. - Shared, tamper-evident records make unauthorized changes easier to detect. - Distributed recordkeeping can reduce reliance on a single institution's transaction database. Limitations: - Stolen private keys, software vulnerabilities, compromised endpoints and smart contract errors can cause financial losses. - A blockchain does not eliminate fraud or guarantee that information entered into the system is accurate. - Governance failures and inadequate recovery procedures may affect the entire consortium. **Regulatory requirements (1.5 marks)** Benefits: - A shared audit trail can support transaction monitoring, reporting and regulatory audits. - Permissioned networks can incorporate identity verification and role-based authorization. Limitations: - Cross-border payments must comply with applicable anti-money laundering and counter-terrorist financing (AML/CFT) rules, sanctions requirements and customer verification obligations. - Different jurisdictions may have different rules concerning data protection, financial licensing and settlement finality. - The consortium must determine responsibility for compliance, transaction disputes and operational failures. A permissioned blockchain may be appropriate when multiple financial institutions need a shared transaction record, improved reconciliation and continuous processing. However, the consortium should conduct a pilot, test throughput and settlement performance, establish privacy and governance controls, and verify regulatory compliance before full deployment. A hybrid approach that connects blockchain with existing banking and payment infrastructure may be more practical than replacing conventional systems completely. **(1.5 Marks)**	

*********** All the Best***********