CERTLABZ CERTIFICATION BOARD

Certified Distributed Ledger Professional (CDLP)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working distributed ledger professional would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1 — Domain 1: Distributed Ledger Foundations — DATA

Emma Wilson matches each need to whether a distributed ledger is a good fit.

Need	Best fit
Shared records across distrusting parties	(to choose)
A private single-company data store	(to choose)
Tamper-evident audit trail	(to choose)
High-speed local caching	(to choose)

Table 1.1 Needs and the best fit.

Which set of choices is correct?

A. Ordinary database, ledger, ordinary database and ledger, so a ledger is used for the single-company store and the local cache only.

B. Distributed ledger, ordinary database, distributed ledger and ordinary database, matching a ledger to shared and tamper-evident needs.

C. Distributed ledger for all four, since a shared ledger is always the better option than a database for any storage need at all.

D. Ordinary database for all four, since a database is always faster and cheaper than a ledger for every possible use case here.

Correct answer: B
A distributed ledger suits records shared across parties that do not fully trust each other and needs that demand a tamper-evident trail. A private single-company store and high-speed local caching are served better and more cheaply by an ordinary database. Using a ledger for everything, or nothing, ignores the tradeoffs.

Question 2 — Domain 1: Distributed Ledger Foundations — CONFIG

James Carter lists the parts of a block, shown out of order.

```
A. Transactions included in the block
B. Hash of the previous block
C. This block hash, over its contents
D. Timestamp and metadata
```

Listing 2.1 Block parts.

Why does each block include the hash of the previous block?

- (A) To make the chain faster to read, because linking by hash lets a node skip directly to the block it wants to load.
- (B) To chain blocks so that altering an earlier block changes every later hash, making tampering evident across the chain.
- (C) To compress the transactions, because the previous hash stores the earlier block data in a smaller encoded form.
- (D) To hide the transactions, because the previous hash encrypts the contents of the earlier block from other nodes.

Correct answer: B
Each block commits to the previous block by including its hash, so changing any earlier block changes its hash and breaks every block after it. That is what makes tampering evident. Hash linking does not speed reads, compress data or encrypt contents.

Question 3 — Domain 2: Consensus and Networks — DATA

Olivia Davis matches each consensus term to its meaning.

Term	Meaning
Proof of work	(to choose)
Proof of stake	(to choose)
Finality	(to choose)
Fork	(to choose)

Table 3.1 Consensus terms.

Which set of meanings is correct?

A. Validators chosen by stake, spending compute to win blocks, a network split, and the point a block cannot be reversed, in that order.

B. Spending compute to win blocks, validators chosen by stake, the point a block cannot be reversed, and a divergence in the chain.

C. A network split, the point a block cannot be reversed, spending compute to win blocks, and validators chosen by stake, in that order.

D. The point a block cannot be reversed, a network split, validators chosen by stake, and spending compute to win blocks, in order.

Correct answer: B
Proof of work spends compute to win the right to add blocks, proof of stake selects validators by the stake they lock up, finality is the point a block can no longer be reversed, and a fork is a divergence in the chain. Only option B pairs each term with its meaning.

Question 4

Domain 3: Cryptography and Keys

CONFIG

Liam Anderson reviews how a user signs a transaction.

```
private key  : kept secret by the user
public key   : derived, shared as the address
signature    : made with the private key
verification : done with the public key
```

Listing 4.1 Signing model.

What does this model guarantee?

A. That only the holder of the private key could have produced the signature, and anyone can verify it with the public key.

B. That the private key can be recovered from the public key, so a lost private key is always easy to restore for the user.

C. That the signature encrypts the transaction, so its contents stay hidden from every other node on the whole network.

D. That the public key must stay secret, because sharing the address would let anyone else sign on behalf of the user.

Correct answer: A
Only the private key can create a valid signature, and anyone can verify it using the derived public key, which proves authenticity without revealing the private key. The private key cannot be recovered from the public key, signing is not encryption, and the public key (address) is meant to be shared.

Question 5

Domain 4: Smart Contracts and Transactions

DATA

Ava Thompson matches each term to its role.

Term	Role
Gas	(to choose)
Smart contract	(to choose)
Account	(to choose)
Token standard	(to choose)

Table 5.1 Transaction terms.

Which set of roles is correct?

A. Code that runs on chain, a shared rule set for tokens, the fee for computation, and an entity that holds a balance, in that order.

B. The fee for computation, code that runs on chain, an entity that holds a balance, and a shared rule set for tokens.

C. An entity that holds a balance, the fee for computation, a shared rule set for tokens, and code that runs on chain, in order.

D. A shared rule set for tokens, an entity that holds a balance, code that runs on chain, and the fee for computation, in order.

Correct answer: B
Gas is the fee paid for computation, a smart contract is code that runs on the ledger, an account is an entity that holds a balance and sends transactions, and a token standard is a shared rule set that lets tokens interoperate. Only option B maps each term to its role.

Question 6

Domain 5: Enterprise and Permissioned Ledgers

DATA

A consortium compares a public and a permissioned ledger for a supply-chain record.

Property	Permissioned ledger
Who can join	Approved members only
Privacy between members	Channels and controls
Throughput	Generally higher
Open public access	No

Table 6.1 Permissioned ledger properties.

When is a permissioned ledger the better choice for the consortium?

A. Never, because a public ledger is always the correct choice for any organisation regardless of privacy or membership needs at all.

B. When known members need shared, tamper-evident records with privacy and higher throughput, rather than open public participation.

C. Only when the consortium wants anyone in the world to join and read every record without any approval or membership control.

D Only when the data must be fully public, since permissioned ledgers are designed to publish all records openly to everyone.

Correct answer: B
A permissioned ledger fits a known group of members who need shared, tamper-evident records with privacy controls and higher throughput, without opening participation to the public. It is not always wrong or right, it is not for open public joining, and it does not publish everything openly.

Question 7 Domain 2: Consensus and Networks

A team wants very high transaction throughput, strong decentralization and strong security all at once, at no tradeoff. What is the realistic guidance?

A All three can always be maximized together at once, so the team should simply demand the highest possible level of each one.

B These three goals trade off against each other, so the team must decide which to prioritize for its use case and accept the balance.

C Throughput is the only goal that matters, so the team should ignore decentralization and security entirely to reach the highest speed.

D Security can be dropped safely on a ledger, because the chain structure alone removes any need to think about security tradeoffs.

Correct answer: B
The scalability trilemma means throughput, decentralization and security trade off against one another, so a design prioritizes among them for its use case rather than maximizing all three. Ignoring decentralization or security to chase speed produces a system that fails the reasons a ledger was chosen.

Question 8 Domain 6: Governance, Security and Adoption

A user loses the private key to a self-custody wallet with no backup. What is the reality, and the lesson?

A The provider can reset the key on request, so losing a private key is a minor inconvenience that support can quickly resolve.

B With no key and no backup the funds are generally unrecoverable, so secure key management and backups are essential from the start.

C The funds move automatically to a new key, because the ledger detects the loss and reassigns the balance to the rightful owner.

D The transactions can simply be reversed, because a distributed ledger lets any user undo their past transactions when needed.

Correct answer: B
In self-custody there is no provider who can reset a lost private key, and without a backup the funds are generally unrecoverable, which is why key management and backups matter so much. Ledgers do not reassign balances on loss, and confirmed transactions are not reversible on demand.

Question 9 Domain 1: Distributed Ledger Foundations

A manager says immutability means a blockchain can never contain wrong or fraudulent data. How should you correct this?

A. The manager is right, because nothing incorrect can ever be written to a blockchain once its consensus rules are switched on.

B. Immutability means recorded data is hard to change, not that inputs are true; wrong or fraudulent data can be recorded and persist.

C. Immutability means the ledger deletes any incorrect entry automatically, so only accurate data can remain on the chain over time.

D. Immutability applies only to timestamps, so the transaction contents themselves can be freely edited by participants at any time.

Correct answer: B
Immutability means it is very hard to alter data once recorded, not that the data is correct. Garbage or fraudulent input can be written and will then persist, so validation belongs before data reaches the ledger. The chain does not auto-delete bad entries, and contents are not freely editable.

Question 10 Domain 5: Enterprise and Permissioned Ledgers

An enterprise design needs a smart contract to act on the current market price of a commodity. What is required, and what is the caution?

A. Nothing extra, because a smart contract can read any external website directly to fetch the current market price on its own.

B. An oracle must supply the external price on chain, and its trustworthiness and manipulation risk must be managed carefully.

C. The price must be hard-coded once at deployment, because a contract can never be given any new external data after it is live.

D. The contract should guess the price from its balance, since on-chain balances always reflect current external market prices.

Correct answer: B
Smart contracts cannot reach off-chain data on their own, so an oracle is needed to bring the external price on chain, and the oracle becomes a trust and manipulation risk that must be managed. Contracts cannot browse the web, a hard-coded price goes stale, and balances do not reflect market prices.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CDLP-001. https://certlabz.com/certifications/certified-distributed-ledger-professional.html